



# Conga Composer



# Table of Contents

- Composer Release Notes.....21
  - Summer21.10.13 Release Notes ..... 21
  - Packages..... 22
  - System Requirements and Supported Platforms ..... 22
  - New Features..... 23
  - Enhancements..... 23
  - Fixed Issues..... 23
  - Known Issues ..... 24
- Summer21.09.08 Release Notes .....24
  - Packages..... 25
  - System Requirements and Supported Platforms ..... 25
  - New Features..... 26
  - Enhancements..... 26
  - Resolved Issues ..... 27
  - Known Issues ..... 27
- Summer21.09.02 Release Notes ..... 27
  - Packages..... 28
  - System Requirements and Supported Platforms ..... 28
  - New Features..... 29
  - Enhancements..... 30
  - Resolved Issues ..... 30
  - Known Issues ..... 30
- Summer21.09.01 Release Notes ..... 30
  - Packages..... 31
  - System Requirements and Supported Platforms ..... 31
  - New Features..... 33
  - Enhancements..... 33
  - Resolved Issues ..... 33
  - Known Issues ..... 33

Summer21.07.21 Release Notes .....	34
Packages.....	34
System Requirements and Supported Platforms .....	35
New Features.....	36
Enhancements.....	36
Resolved Issues .....	36
Known Issues .....	36
Summer21.07.14 Release Notes.....	37
Packages.....	37
System Requirements and Supported Platforms .....	38
New Features.....	39
Enhancements.....	39
Resolved Issues .....	39
Known Issues .....	40
Summer '21 Release Notes .....	40
Packages.....	41
System Requirements and Supported Platforms .....	41
New Features.....	42
Enhancements.....	42
Resolved Issues .....	43
Known Issues .....	44
About Conga Composer.....	46
What's New in Composer Documentation.....	48
Composer for Administrators .....	52
Getting Started with Composer.....	53
Composer Basics .....	53
Installing Composer .....	78
Configuring Composer .....	86
Building Composer Solutions.....	130
Using Composer .....	130
Using the Document Automation Wizard.....	143
Using Solution Manager .....	151
Composer for Lightning.....	202

Composer for Salesforce1.....	214
Composer Advanced Features .....	224
Composer Lightning Component.....	236
Conga Keys .....	243
Create a Solution Collection .....	248
Gathering Data for Composer.....	249
Data Gathering Basics.....	249
Using Salesforce Reports with Conga Composer .....	257
Using Conga Queries with Conga Composer .....	270
Creating Composer Templates.....	287
Composer Template Basics.....	287
Word Templates.....	306
PowerPoint Templates .....	401
Excel Templates .....	427
HTML Email Templates .....	451
PDF Templates.....	463
Conga Composer Parameter Guide.....	484
Parameter Basics.....	484
Parameter Categories.....	484
A Few Rules About Parameters.....	485
Adding Parameters to a Solution .....	486
How Parameters are Formatted.....	487
Activity Logging Parameters.....	489
Adobe PDF Template Parameters .....	493
Adobe Sign Integration Parameters.....	494
Advanced Control Parameter.....	495
AmazonS3Visible.....	496
APDF .....	497
APF.....	498
Appearance Parameters .....	498
AssembleChatter .....	500
AttachmentID.....	500
AttachmentParentId.....	501
AttachmentSourceId.....	502

AWD.....	502
Background Mode Parameters .....	503
Bigtincan Integration Parameters.....	504
bigtincanVisible.....	505
BML.....	506
Box Integration Parameter.....	506
BoxVisible.....	506
BrandingCode.....	507
btcuser .....	507
CETFilter.....	508
ChatterCommunityId .....	509
Conga Contracts Parameters .....	509
CongaEmailTemplateGroup .....	510
CongaEmailTemplateld.....	510
Conga Email Template Parameters .....	511
Conga Sign Integration Parameters.....	512
ContactId .....	513
ContentCustomFieldName .....	514
ContentCustomFieldValue .....	514
ContentDocumentId.....	515
ContentNetworkId.....	515
contentShareType.....	516
contentVisibility .....	517
ContentVisible.....	518
ContentWorkspaceld.....	519
CSAuth[1-10].....	519
CSBusinessUnit.....	520
CScurl[1-10].....	521
CSEmail[1-10] .....	522
CSEmailMessage.....	523
CSEmailSubject .....	524
CSExpiration.....	524
CSExpirationReminder .....	525
CSFirst[1-10] .....	525

CSLang[1-10].....	526
CSLast[1-10] .....	527
CSMobile[1-10].....	528
CSMultiDocs.....	529
CSName[1-10].....	530
CSReassignAllowed.....	531
CSRecipient[1-10] .....	531
CSRequestReminder .....	532
CSRole[1-10].....	532
CSRoutingType .....	533
CSSenderReturnUrl.....	534
CSVisible.....	535
Culture.....	535
CurrencyCulture .....	536
Data Gathering Parameters.....	537
DataTagEnable.....	537
DC.....	538
DCL.....	539
DefaultLocal .....	539
DefaultPane .....	540
DefaultPDF .....	540
DisableRowNums .....	541
DocFolderId .....	542
DocuSign7 .....	542
DocuSignAuthoritativeCopy.....	543
DocuSignBrandName .....	543
DocuSignCustomField .....	544
DocuSignCustomFieldValue.....	545
DocuSignEmailBody .....	545
DocuSignEmailSubject.....	546
DocuSignEnableWetSign .....	546
DocuSignEndpoint.....	547
DocuSignExpireIn .....	547
DocuSignExpireVoidEnv.....	548

DocuSignIgnoreOCR .....	548
DocuSign Integration Parameters .....	549
DocuSignR1AccessCode .....	551
DocuSignR1Email .....	551
DocuSignR1EmailBlurb .....	552
DocuSignR1EmailSubject .....	552
DocuSignR1IID .....	553
DocuSignR1IIDCheck .....	554
DocuSignR1IIDCheckSMSPhone .....	554
DocuSignR1IIDCheckType .....	555
DocuSignR1Language .....	557
DocuSignR1Name .....	559
DocuSignR1Role .....	560
DocuSignR1RoutingOrder .....	560
DocuSignR1SigningGroup .....	561
DocuSignR1SignNow .....	562
DocuSignR1SIPName .....	563
DocuSignR1Type .....	563
DocuSignRelatedAccountId .....	564
DocuSignRemindAgainIn .....	565
DocuSignRemindIn .....	565
DocuSignSendAsUserId .....	566
DocuSignSendReminders .....	566
DocuSignTag .....	567
DocuSignVisible .....	568
DocuSignWarnIn .....	568
DS0 .....	569
DS1 .....	569
DS2 .....	570
DS3 .....	570
DS4 .....	571
DS5 .....	572
DS6 .....	572
DS7 .....	573

DS7Preview .....	577
DS8 .....	578
DS9 .....	579
DS10 .....	579
DV0.....	580
EC.....	581
ECL .....	581
EmailAdditionalTo .....	581
EmailBCC.....	582
EmailCC.....	582
EmailFromId.....	583
EmailLetterheadId .....	584
Email Parameters.....	584
EmailRelatedToId .....	585
EmailReplyToId .....	586
EmailSubject.....	586
EmailTemplateAttachments.....	587
EmailTemplateId.....	587
EmailToId.....	588
EmailUseSignature.....	589
ESAccountId .....	589
ESAdditionalTo.....	590
ESAgreementName.....	590
ESAgreementTemplateId.....	591
ESContactId.....	591
ESContactId1 .....	592
ESContactId2 .....	593
ESContactId3 .....	593
ESContractId.....	594
ESCopyTo .....	594
ESCustomField .....	595
ESCustomFieldValue .....	595
ESDataMappingId.....	596
ESExpireInXDays.....	596



eSignLive Integration Parameters .....	597
eSignLivePackageName.....	597
eSignLiveRIEmail.....	598
eSignLiveRIName.....	598
eSignLiveVisible .....	598
ESLanguage.....	599
ESLeadId.....	599
ESLeadId1.....	600
ESLeadId2.....	600
ESLeadId3.....	601
esSignerIID.....	601
ESMessage.....	602
ESOpportunityId.....	602
ESPreview.....	603
ESRecipient1.....	603
ESRecipient1Role .....	603
ESRecipient2.....	604
ESRecipient2Role .....	604
ESRecipient3.....	605
ESRecipient3Role .....	605
ESRecipient4 .....	606
ESRecipient4Role .....	606
ESRecipient5 .....	607
ESRecipient5Role .....	607
ESRecipient6 .....	608
ESRecipient6Role.....	608
ESRecipient7.....	609
ESRecipient7Role .....	609
ESRecipient8 .....	610
ESRecipient8Role .....	610
ESRecipient9 .....	611
ESRecipient9Role .....	611
ESRecipient10.....	612
ESRecipient10Role .....	612

ESRecipient11.....	613
ESRecipient11Role.....	613
ESRecipient12Role.....	614
ESRemindRecipient.....	614
ESRequirePWSign.....	615
ESRequirePWView.....	615
ESRequireSelfSign.....	616
ESSignatureGroup1.....	616
ESSignatureOrder.....	617
ESSignatureType.....	617
ESSVBA.....	618
ESVisible.....	618
ExcelAutoFitColumns.....	619
ExcelAutoFitRows.....	620
Excel Parameters.....	620
ExcelPW.....	621
ExcelStripFormulas.....	621
FlattenPDFForms.....	622
FP0.....	622
Google Drive Parameters.....	623
GoogleVisible.....	623
HideContent.....	624
Id.....	624
LG0.....	625
LG1.....	625
LG2.....	626
LG3.....	626
LG4.....	627
LG5.....	627
LG6.....	628
LG7.....	628
LGAttachOption.....	629
LGPParentID.....	630
LGWhoID.....	630

LinkService.....	631
LiveEditEnable .....	631
LiveEditVisible.....	632
Locale and Culture Parameters .....	633
LRTF .....	633
MFTS0.....	634
MFTS1.....	634
MFTS2.....	635
MFTSId0 .....	635
MFTSId1.....	636
MFTSId2.....	636
MFTSValue0.....	637
MFTSValue1 .....	638
MFTSValue2.....	638
MTOFN.....	639
NETId.....	639
OCNR.....	640
OFN.....	641
OFP.....	642
OneDrive Parameters.....	643
OneDriveVisible .....	643
Output Parameters .....	644
PDFA.....	645
PDFEF.....	646
PDFIQ.....	646
PDFQR .....	647
PDFVersion.....	647
PowerPoint Parameters .....	648
PPTPreserveRows.....	648
PPTXDUC.....	649
PS0.....	649
PS1.....	650
PS2.....	650
PS3.....	651

PS4 .....	651
PSPW .....	652
QMode .....	652
QueryID .....	658
QueryStringField.....	659
QueryStringField1.....	660
QueryStringField2.....	660
QVar0Format .....	661
QVar0Id .....	662
QVar1Format .....	663
QVar1Id.....	663
QVar2Format .....	664
QVar2Id.....	665
ReportId.....	665
ReportVar .....	667
ReturnPath.....	667
RM .....	668
RTLFont .....	669
Salesforce1 Parameters.....	669
Salesforce Chatter Parameters .....	670
Salesforce Content Parameters.....	670
SC0 .....	671
SC1 .....	671
SCHP .....	673
SCMEndpoint.....	674
SCMPreview.....	674
SCMVisible .....	675
SelectAttachments.....	675
SelectTemplates.....	676
SertifiCCSignerEmail .....	677
SertifiCustomSettings .....	677
SertifiEmailTo .....	678
SertifiExpirationDate.....	678
SertifiFileName .....	679

Sertifi Integration Parameters .....	679
Sertifi Pass Through Method.....	680
SertifiPaymentAmount .....	680
SertifiPaymentName .....	680
SertifiPaymentType.....	681
SertifiSigner1Email.....	681
SertifiSigner1Id .....	682
SertifiSigner2Email.....	683
SertifiSigner2Id .....	683
SertifiSigner3Email.....	683
SertifiSigner4Email.....	684
SertifiSigner5Email.....	685
SertifiVisible .....	685
SETI.....	686
SF1.....	686
SMD.....	687
SpringCM Integration Parameters.....	688
Template Fields Used in Parameters.....	688
Template Group.....	689
Templateld.....	690
Template Selection Parameters.....	690
TNC.....	691
TPV .....	691
UF0 .....	692
WordCheckboxSize.....	693
Word Parameters.....	693
WordRNP .....	694
WordStriplfs .....	694
WPD .....	695
WPDPW .....	695
xid.....	696
XMD .....	697
ZipFiles .....	698
ZipPDFs .....	698

Advanced Composer Solutions.....	699
Composer Tips and Tricks.....	699
Composer Integrations.....	755
Global Reporting with Conga Composer .....	775
Conga Composer and DocuSign for Salesforce Integration.....	782
Transition from SpringCM to the Conga Product Suite .....	796
Composer Enterprise Edition.....	797
Advanced Features.....	798
CipherCloud for Salesforce.....	798
Symantec Cloud Data Protection (fka PerspecSys).....	800
External Data.....	801
Custom Branding.....	806
Negotiate, Redline, and True-Up in Composer Enterprise .....	807
Increased Conga Template Limit .....	807
Composer Troubleshooting Help.....	807
Caching Error on the Conga Composer Setup Menu.....	809
Chrome Redirect Issues when using Composer.....	809
Composer Not Redirecting to Adobe Esign in Community.....	812
Composer Troubleshooting Tips .....	812
Conditionally Show or Hide Data in Word Templates .....	816
Conga Composer Debug URL .....	816
Conga Word templates - When there is no data in a field, spacing and font size are altered.....	820
Connection to Salesforce Failed .....	821
CreateEnvelope in DocuSignAPI UserID does not have a valid membership in Account.....	822
Create a Conga Composer List View Button (Lightning Experience).....	822
Critical Update Actions for Composer Versions.....	826
Data appears on the QueryDataX dataset for some users, but not others. Why? .....	835
Data appears on the ReportDataX dataset for some users, but not others. Why? .....	835
Differences between Conga Composer versions.....	835
Disable Composer 7 Triggers.....	836
Document not Merging to SpringCM.....	837
Removing Duplicate Conga Composer Setup Tab.....	838
Duplicate solutions record created where solution pack installed.....	838
Edit My Own Posts Permission Required to Post to Chatter.....	839

Error Using Google Drive Integration with Composer Templates .....	840
Excel to PDF - Different Values .....	840
Fields not Copying to the Clipboard in Conga Composer .....	841
Fields Won't Display Even After Permission Is Granted in Salesforce to Access that Field (Release 8) .....	842
Field Not Merging in PowerPoint Template .....	842
How do I know what version of Composer I'm using?.....	843
How do I move my Conga Composer subscription from one Salesforce instance to another?.....	843
How to use Extended Ping and Trace Route Instructions for Conga.....	844
Images in Word.....	845
Insufficient Privileges Error when a System Admin launches an OAuth button as a user .....	845
Log Into One Salesforce Instance or as One User per Browser .....	846
Merge Fields Displaying Unexpected Results.....	846
One or more documents failed to convert to PDF .....	847
OpplinelItems Dataset Missing in Template Builder.....	848
Remote Site Error in Conga Solution Manager .....	849
Removing duplicate tabs.....	850
Shipping and Billing Merge Fields.....	851
Templates not displaying or showing fewer than expected in Composer (Release 8) .....	851
Template is not populating in template list on the Conga dialog box or is not visible in Conga Template Manager.....	853
Troubleshooting Font Styles .....	854
Using Send through External Email Services with Conga Composer.....	855
What is Aspose related to my PDF? .....	857
Where Did My Conga Mail Merge Tab Go? (Release 8).....	858
Why didn't my dynamic formula update correctly in my Conga Composer Excel output file?.....	858
Templates appear for administrators but not end-users.....	859
Composer Error Messages .....	859
Adobe eSign Installation Error.....	861
A problem occurred in the merge process. Please try once more before contacting Support. (Error: 0122) (Release 7).....	861
Cannot remove because there is no parent (Release 8).....	862
Composer Authorization Error   404 Page Cannot Be Found .....	862
Conga Composer Cannot Communicate with Salesforce (Release 7).....	863
Conga Composer cannot connect with Salesforce because master ID is invalid. (Release 7) .....	864

Conga Composer Expiration Warning .....	864
Conga Composer Settings Error .....	865
Conga Email Staging Record Error .....	866
Error "id parameter is missing" with Composer 8 buttons in Lightning.....	866
Error 0754 The document appears to be corrupted & cannot be loaded (Release 7).....	867
Error 500 - "Internal Server Error" (Release 7) .....	868
Error appears when clicking security token in Sandbox Orgs.....	868
ERROR at Row:1:Column:40 value of filter criterion for field 'CloseDate' must be of type date and should not be enclosed in quotes (Release 7) .....	870
Error: CreateEnvelope in DocuSignAPI. One or both of Username and Password are invalid. You may need to set &DocuSignEndpoint="AlternateEndpoint, e.g., Demo).....	871
A problem occurred in the rendering process. Please verify your template is valid (Release 7) .....	872
Found end of mail merge region 'ReportData1 0' without matching start (Release 7).....	872
Input string was not in the correct format .....	873
Insufficient Access: Current IP Address is not allowed or invalid_grant: IP restricted .....	874
Insufficient Access to Custom Settings Error .....	875
Invalid Grant: authentication failure.....	875
Invalid master object ID(must be a Salesforce id) (Release 7).....	876
Invalid PPTX TableGroupStart Field.....	877
Invalid Query Filter Operator .....	877
Invalid Session ID found in SessionHeader (Release 7) .....	877
Merge and Box Errors .....	878
No DocuSign Recipients Defined (Release 7).....	879
Object reference is not set to an instance of an object (Release 7) .....	880
Object reference is not set to an instance of an object (Release 8).....	880
One or More Templates Failed to Merge or Operation Was Cancelled .....	882
Ouch... something's definitely gone wrong. Please try again. (Error #1890: ORNF) (Release 7).....	883
Our Apologies . . . Your Conga URL contained illegal characters (Release 8 and 7) .....	884
Page Cannot Be Found .....	884
Read pdf error: Trailer is not found (Release 7).....	885
Reduce complexity of SOQL statement to avoid errors.....	885
Report exceeds maximum supported rows from PointMerge (Release 7).....	886
Report size cannot be determined, either because the report is excessively large, the report is unavailable or the report type is matrix. Please verify the Report Id. ....	887
REQUEST LIMIT EXCEEDED:TotalRequests Limit exceeded.....	888



Server was unable to process request. Value cannot be null. Parameter name: password.....	888
Sign in with Google temporarily disabled for this app. This app has not been verified yet by Google in order to use Google Sign In.....	889
Something's gone wrong! An internal server error has occurred.....	891
Something's not quite right about your data - it can't be displayed in Excel. Column 'id' does not allow nulls. (Release 7).....	891
Something's wrong with your template (bad mojo). The merge template has serious problems.....	892
Stack Empty (Release 7).....	894
String not recognized as a valid DateTime .....	895
Template Id(s) do not exist.....	895
Template Id is Invalid.....	896
There hasn't been a file or email template ID defined, background mode cannot continue when trying to use (Release 8).....	897
There is no row at position 0 .....	898
There was an issue getting the Conga Authorization Response .....	898
There was an issue processing the Composer request .....	899
The ',' character, hexadecimal value 0x2C, cannot be included in a name.....	899
The report is invalid. Are you using My Domains or matrix style reports? .....	901
Parameter p3lkid value error.....	901
Unable to parse the query id entries, please check the URL syntax (Release 8) .....	902
Unexpected end of children nodes reached (Release 7).....	903
URLENCODE function for illegal characters in button URL.....	903
User Management Access Error.....	905
Composer Output Contains Blank Fields.....	906
Your Conga Session has Expired .....	907
Salesforce session expired or the Salesforce API Servers are offline - INVALID SESSION ID or SERVER URL (Release 7 and 8).....	907
"Conga can't communicate with Salesforce" error.....	908
Collection size exceeds maximum size of 1,000 .....	909
Conga Trigger .....	910
Conga Trigger Installation and Setup.....	911
Conga Trigger Product Guide .....	920
The About Conga Trigger Tab.....	921
The Conga Trigger Setup Tab.....	922
The Conga Trigger Formula Builder Tab.....	923

The Conga Trigger Dashboard Tab.....	924
Upgrade Conga Trigger to Release 8.....	925
Automate the Conga Trigger Solutions.....	929
Troubleshooting Tips for Conga Trigger Solutions.....	949
Conga Batch.....	954
Conga Batch Installation and Setup.....	954
Conga Batch Product Guide.....	971
The Conga Batches Tab.....	972
The Conga Batch Dashboard Tab.....	973
The Conga Batch Setup Tab.....	974
The About Conga Batch Tab.....	976
Building Conga Batch Solutions.....	977
Troubleshooting Conga Batch.....	1013
Conga Mail Merge.....	1031
Mail Merge Product Guide.....	1031
What is Conga Mail Merge?.....	1032
Using Mail Merge.....	1033
Create and Store a Mail Merge Template.....	1041
Troubleshooting Mail Merge.....	1043
QuickMerge Product Guide.....	1047
Learn QuickMerge Basics.....	1048
Install QuickMerge.....	1048
Use QuickMerge.....	1048
About Conga QuickMerge.....	1048
Installing Conga QuickMerge.....	1049
Launching Conga Mail Merge from Conga QuickMerge.....	1050
Use Conga QuickMerge to Automate Conga Composer (Release 7).....	1050
Use Conga QuickMerge to Automate Conga Mail Merge.....	1051
Composer for Users.....	1053
Generating a Document.....	1054
Using Composer Button.....	1054
Using Composer Lightning Component.....	1057
Using a Local Template.....	1061

Using Live Edit.....	1062
To use Live Edit .....	1062
Considerations.....	1063
Using Composer Advanced Features .....	1064
Send a Document as a Link.....	1065
Email Notification.....	1065
View Document History.....	1066
PDF Viewer .....	1068
Launching Conga Batch from a List View Button .....	1069
To launch Conga Batch from a List View Button.....	1070
Localization of Classic Composer UI .....	1071
To enable dynamic translation in the Composer user interface.....	1071
Localization of Lightning Component UI .....	1072
Composer Features by Release .....	1074
Features by Release.....	1074
Community & Learning Center Resources.....	1075
Ready to get started? .....	1075

## Composer

Conga Composer enables Salesforce customers to customize, streamline and scale document generation and reporting for unlimited use cases across all organizations and industries.

# Composer Release Notes

Discover what's new in the latest release of Conga Composer.


- [Summer21.10.13 Release Notes](#)
- [Summer21.09.08 Release Notes](#)
- [Summer21.09.02 Release Notes](#)
- [Summer21.09.01 Release Notes](#)
- [Summer21.07.21 Release Notes](#)
- [Summer21.07.14 Release Notes](#)
- [Summer '21 Release Notes](#)

## Summer21.10.13 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.10.13 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- **Packages:** Lists packages that are required to upgrade to this release of the product
- **System Requirements and Supported Platforms:** Lists requirements and recommendations for installing this release
- **New Features:** Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- **Enhancements:** Provides high-level descriptions of enhancements to existing features
- **Fixed Issues:** Lists customer-reported issues that are fixed in this release or known issues fixed from previous releases
- **Known Issues:** Lists known issues that are applicable in this release

 This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.

Product	Latest Certified Version (Version Number   Version Name)
Conga Composer <b>(New)</b>	8.20.338


### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>

System Requirement	Minimum Supported Version
Browser	<ul style="list-style-type: none"> <li>• Google Chrome® <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox® <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.

## New Features

There are no new features in this release. Keep checking the Conga Documentation Portal for new updates.

## Enhancements

There are no new enhancements in this release.

## Fixed Issues

The following table lists the issue fixed in this release. If any actions are required, they will be listed in this table.

Case Number	Conga Internal ID	Description
00444332	CM-3063	The line break in the watermark now works as expected for the PDF template.

## Known Issues

The following table provides the cumulative list of known issues up to this release.

Conga Internal ID	Description
CM-3679	All Salesforce Clouds, except Salesforce Community Cloud, support Box integration with Composer in background mode.
CM-3680	All Salesforce Clouds, except Salesforce Community Cloud, support Adobe Sign integration with Composer in background mode.

## Summer21.09.08 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.09.08 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- [Packages](#): Lists packages that are required to upgrade to this release of the product
- [System Requirements and Supported Platforms](#): Lists requirements and recommendations for installing this release
- [New Features](#): Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- [Enhancements](#): Provides high-level descriptions of enhancements to existing features
- [Resolved Issues](#): Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- [Known Issues](#): Lists known issues that are applicable in this release



**i** This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.

Product	Latest Certified Version (Version Number   Version Name)
Conga Compose	8.20.330


### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>

System Requirement	Minimum Supported Version
Browser	<ul style="list-style-type: none"> <li>• Google Chrome®                             <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox®                             <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser’s configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	<p>Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.</p>

## New Features

There are no new features in this release. Keep checking the Conga Documentation Portal for new updates.

## Enhancements

The following section describes the existing feature that is changed (or is no longer supported) in this release.

## Salesforce API Version 52 Support

In this release, Composer now supports *Salesforce API Version 52* for **North America (NA)** region. To access the new Salesforce objects, users must use the **Template Builder**. For more details, see [Support of Salesforce API Version with Composer](#).

### Important

The users can access the new Salesforce objects within the **Document Automation Wizard** only with the upcoming [Composer December '21](#) release package.

## Resolved Issues

There are no resolved issues in this release.

## Known Issues

The following table provides the cumulative list of known issues up to this release.

Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

## Summer21.09.02 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.09.02 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- [Packages](#): Lists packages that are required to upgrade to this release of the product

- [System Requirements and Supported Platforms](#): Lists requirements and recommendations for installing this release
- [New Features](#): Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- [Enhancements](#): Provides high-level descriptions of enhancements to existing features
- [Resolved Issues](#): Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- [Known Issues](#): Lists known issues that are applicable in this release

**i** This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.


Product	Latest Certified Version (Version Number   Version Name)
Conga Compose	8.20.330

### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>
Browser	<ul style="list-style-type: none"> <li>• Google Chrome® <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox® <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.

## New Features

There are no new features in this release. Keep checking the Documentation Portal for new updates.

## Enhancements

The following section describes the existing feature that is changed (or is no longer supported) in this release.

### Salesforce API Version 52 Support

In this release, Composer now supports *Salesforce API Version 52* for **Europe, Middle East, and Africa (EMEA)** region. To access the new Salesforce objects, users must use the **Template Builder**. For more details, see [Support of Salesforce API Version with Composer](#).

 **Important**

The users can access the new Salesforce objects within the **Document Automation Wizard** only with the upcoming [Composer December '21](#) release package.

## Resolved Issues

There were no resolved issues in this release.

## Known Issues

The following table provides the cumulative list of known issues up to this release.


Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

## Summer21.09.01 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.09.01 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- **Packages:** Lists packages that are required to upgrade to this release of the product
- **System Requirements and Supported Platforms:** Lists requirements and recommendations for installing this release
- **New Features:** Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- **Enhancements:** Provides high-level descriptions of enhancements to existing features
- **Resolved Issues:** Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- **Known Issues:** Lists known issues that are applicable in this release

 This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.


Product	Latest Certified Version (Version Number   Version Name)
Conga Composer <b>(New)</b>	8.20.330

### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>
Browser	<ul style="list-style-type: none"> <li>• Google Chrome® <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox® <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.



## New Features

There are no new features in this release. Keep checking the Documentation Portal for new updates.

## Enhancements

The following section describes the existing feature that is changed (or is no longer supported) in this release.

### Salesforce API Version 52 Support

In this release, Composer now supports *Salesforce API Version 52* for the **Asia-Pacific (APAC)** region. To access the new Salesforce objects, users must use the **Template Builder**. For more details, see [Support of Salesforce API Version with Composer](#).

#### Important

The users can access the new Salesforce objects within the **Document Automation Wizard** only with the upcoming [Composer December '21](#) release package.

## Resolved Issues

There were no resolved issues in this release.

## Known Issues

The following table provides the cumulative list of known issues up to this release.


Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

# Summer21.07.21 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.07.21 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- **Packages:** Lists packages that are required to upgrade to this release of the product
- **System Requirements and Supported Platforms:** Lists requirements and recommendations for installing this release
- **New Features:** Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- **Enhancements:** Provides high-level descriptions of enhancements to existing features
- **Resolved Issues:** Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- **Known Issues:** Lists known issues that are applicable in this release

 This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.


Product	Latest Certified Version (Version Number   Version Name)
Conga Composer <b>(New)</b>	8.20.322

### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>
Browser	<ul style="list-style-type: none"> <li>• Google Chrome® <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox® <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>

System Requirement	Minimum Supported Version
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.

## New Features

There are no new features in this release. Keep checking the Documentation Portal for new updates.

## Enhancements

The following section describes the existing feature that is changed (or is no longer supported) in this release.

### Track APIMode Usage

The *Composer Activity Report* now allows *Composer Administrators* to track their **APIMode** usage. For details on how to generate an Activity Report, see [Composer Activity Report Overview](#).

## Resolved Issues

There were no resolved issues in this release.

## Known Issues

The following table provides the cumulative list of known issues up to this release.

Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.


Conga Internal ID	Description
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

## Summer21.07.14 Release Notes

These Release Notes contain the following information about Conga Composer Summer21.07.14 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- **Packages:** Lists packages that are required to upgrade to this release of the product
- **System Requirements and Supported Platforms:** Lists requirements and recommendations for installing this release
- **New Features:** Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- **Enhancements:** Provides high-level descriptions of enhancements to existing features
- **Resolved Issues:** Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- **Known Issues:** Lists known issues that are applicable in this release

 This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. The packages marked as **(New)** are new in this release.

Product	Latest Certified Version (Version Number   Version Name)
Conga Composer (New)	8.20.320


### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>

System Requirement	Minimum Supported Version
Browser	<ul style="list-style-type: none"> <li>• Google Chrome® <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox® <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.

## New Features

There are no new features in this release. Keep checking the Documentation Portal for new updates.

## Enhancements

There are no enhancements in this release.

## Resolved Issues

The following table lists the issue resolved in this release.

Conga Internal ID	Description
CM-3776	Conga Composer now allows you to run the <i>Composer Activity Report</i> for all regions (NA, APAC, and EMEA) from the <i>Conga Composer Setup</i> menu. Additionally, if you select "Today" as the end date, all "Today" activity is now reflected.

## Known Issues

The following table provides the cumulative list of known issues up to this release.

Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

## Summer '21 Release Notes

These Release Notes contain the following information about Conga Composer Summer '21 Release.

For more information on new features, enhancements, and document improvements refer to [What's New in Composer Documentation](#).

- [Packages](#): Lists packages that are required to upgrade to this release of the product
- [System Requirements and Supported Platforms](#): Lists requirements and recommendations for installing this release
- [New Features](#): Provides high-level descriptions of new features introduced in this release, with links to more detailed information
- [Enhancements](#): Provides high-level descriptions of enhancements to existing features
- [Resolved Issues](#): Lists customer-reported issues that are resolved in this release or known issues resolved from previous releases
- [Known Issues](#): Lists known issues that are applicable in this release



**i** This documentation may contain descriptions of software features that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document. Please contact your CSM or AE for information on your specific features and licensing.

## Packages

The following packages and dependent packages are required to upgrade to this release to utilize all the new features of this release. These are the minimum required versions; later versions are also supported. Separate prerequisites for each feature can be found in the respective documentation. Packages marked as **(New)** are new in this release.

Product	Latest Certified Version (Version Number   Version Name)
Conga Composer <b>(New)</b>	8.200 - Wilson Peak


### Upgrade Notes

To upgrade Conga Composer, see [Upgrading to the Latest Version of Composer](#).

## System Requirements and Supported Platforms

The following table lists the minimum requirements for installing and using Conga Composer.

System Requirement	Minimum Supported Version
Salesforce Edition	<ul style="list-style-type: none"> <li>• Performance / Unlimited</li> <li>• Enterprise</li> <li>• Professional (limitations may exist)</li> <li>• Developer</li> <li>• Sales Cloud</li> <li>• Service Cloud</li> <li>• Force.com / App Cloud</li> <li>• Health Cloud</li> <li>• Financial Services Cloud</li> <li>• Quote-to-Cash</li> </ul>

System Requirement	Minimum Supported Version
Browser	<ul style="list-style-type: none"> <li>• Google Chrome®               <ul style="list-style-type: none"> <li>• You may have to <a href="#">enable redirects</a> for Composer in Chrome version 64 or higher.</li> </ul> </li> <li>• Mozilla Firefox®               <ul style="list-style-type: none"> <li>• See <a href="#">Mozilla getting started</a> to ensure Salesforce works optimally with Firefox, update the browser's configuration settings.</li> </ul> </li> <li>• See <a href="#">Supported browsers for Salesforce Classic</a>. Salesforce Classic is supported with Microsoft® Internet Explorer® version 11, and Apple® Safari® version 11 on macOS. The most recent stable versions of Microsoft Edge, Mozilla® Firefox®, and Google Chrome™ are also supported. There are some limitations.</li> <li>• <a href="#">Supported browsers for the Lightning Experience</a>.</li> <li>• <a href="#">Recommendations and Requirements for all browsers</a>.</li> </ul>
Template Software	<ul style="list-style-type: none"> <li>• Microsoft Office® 2016 (Mac or PC)</li> <li>• Adobe Acrobat® XI, Acrobat Reader</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Adobe LiveCycle PDFs are not supported.</p> </div>
Mobile	Compatible with Salesforce Mobile and all mobile browsers when configured following the documentation.

## New Features

There are no new features in this release. Keep checking the Documentation Portal for new updates.

## Enhancements

The following section describes the existing features that are changed (or are no longer supported) in this release.

## Enhancement to the Composer Lightning UI

The Composer Lightning User Interface supports a mix of Qvars, Custom field references, and Ids/Keys for `templated` and other [supported parameters](#).

## Integration of Composer with DocuSign App Launcher Package

*Conga Composer* now integrates with the new *DocuSign App Launcher Package*. For more information, see [Integrating Composer with DocuSign App Launcher Package](#).

## New Parameter Functionality for the Composer Lightning Component

The [DSI](#) and [EC](#) parameters are now supported with the Composer Lightning Component. For more information, see [Supported Parameters for the Composer Lightning Component](#).

## Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
CM-2775	Conga Composer Lightning Component Solution now displays attachments when there is a Qvar for <code>attachmentId</code> and Qvar contains a query with dynamic filters.
CM-3316	The user with the <i>Composer Community User</i> permission set can now access the page with a Lightning Component without encountering a permission set-related error.
CM-3092	When you use <i>Templated</i> in a Conga Composer lightning solution that contains a formula field and the formula results in multiple templates, the templates in the merge UI and in the merged document are now ordered as mentioned in the formula field result.

Conga Internal ID	Description
CM-3447	When the DocuSignRoutingOrder parameter is set to 2, the user can now select or deselect any template from the list. Previously, the template section in the Lightning Component UI was disabled, and the user couldn't select or deselect any templates.

The following table lists the known issues resolved from the previous release.

Conga Internal ID	Description
CM-2566	Conga Composer Solutions with the AC0 (Activity Logging) parameter launched from the Composer Lightning Component in the Salesforce Community now successfully downloads a merged document.
CM-3051	When the merging process is canceled, Conga Solutions launched from Salesforce using the Lightning Component does not display a component error.
CM-3670	All the templates are displayed properly on the merge UI when you use TemplateId in a Conga Composer Lightning solution that contains a mix of Qvar with formula field and static value.
CM-3678	The user can successfully launch the Conga Composer Solutions from an Android device using the Composer Lightning Component.

## Known Issues

The following table provides the cumulative list of known issues up to this release.

Conga Internal ID	Description
CM-3679	Box integration with Composer is not working properly with the background mode in the Salesforce Community.

Conga Internal ID	Description
CM-3680	Adobe Sign integration with Composer is not working properly with the background mode in the Salesforce Community.

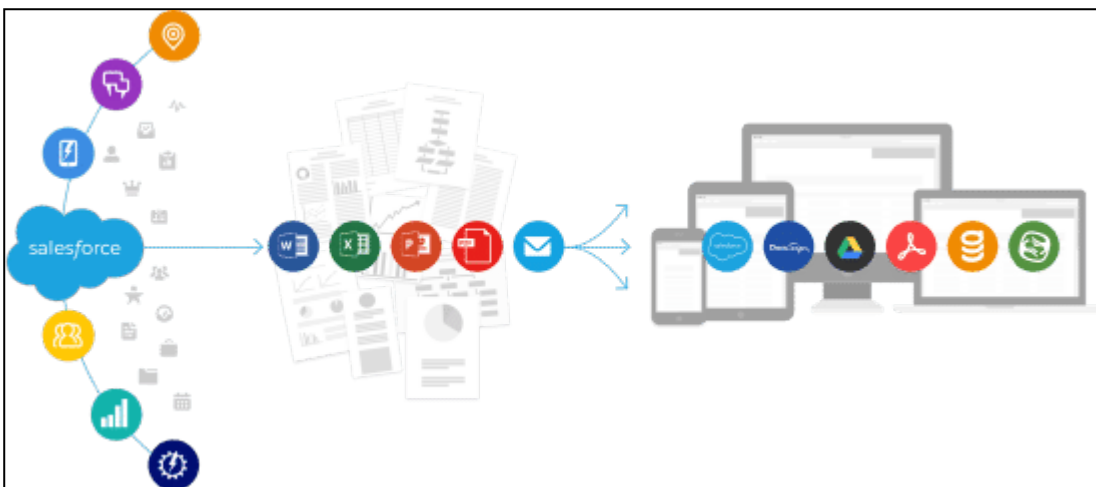
# About Conga Composer

Conga Composer enables Salesforce customers to customize, streamline and scale document generation and reporting for Salesforce. Composer can automatically populate richly formatted templates with data from any standard or custom object in Salesforce and send it via a variety of delivery options.

The administrator configures Composer Solutions for end-users to run Composer from a button or link on an object in Salesforce. The administrator decides what Salesforce object data to leverage, create the document templates, dictates formatting, and determines the customization options available to end-users. End-users can then click the button to run the solution.

The administrator can allow end-users to make selections, such as which template to use for the final merged files, or pre-select templates and other options and run the solution entirely in the background, and also automate solutions using Salesforce workflows or on a schedule using Conga Batch and Conga Trigger.

Once documents are created, Composer can also allow you to email documents, send documents for eSignature, or store the documents in services like Box, Google Drive, or OneDrive for safekeeping.



With the ability to predefine everything from templates and email recipients to activity logging and output file storage, Composer transforms complex routines into single button clicks.

Conga Composer allows an administrator to perform the following administrative tasks:





- Install and Upgrade Conga Composer from the Salesforce AppExchange.
- Set up and configure Conga Composer with Conga Composer Setup.
- Create a Composer Solution.

- Use the Document Automation Wizard to create and modify a Composer Solution.
- Use the Composer Solution Manager to create and modify a Composer Solution.
- Create or modify a Composer Solution manually in Salesforce Setup.
- Create a Conga Template.
  - Create a Conga Template and add merge fields to the template using the Conga Template Builder or Document Automation Wizard.
  - Conga supports the following file types as Conga Templates:
    - Word
    - Excel
    - PowerPoint
    - PDF
    - HTML Email
- Gather additional data for a Composer Solution.
  - Use SOQL Queries and Salesforce Reports to gather additional data to merge for a Composer Solution.
- Customize your Conga Solution with additional behavior.
  - Use Conga Composer parameters to add additional functionality to your Conga Solution.
- Deliver merged documents.
  - Conga Composer allows you to deliver merged documents through the following methods:
    - Download in your browser
    - Email
    - Save to Salesforce
    - Save to a third party repository
    - Send for eSignature
    - Post to Chatter
- Automate file generation and delivery with Conga Batch and Conga Trigger.

# What's New in Composer Documentation

This section lists changes in the documentation to support each release.

## Summer '21

Document	Publication Date	Topic	Description
Summer '21	 08 Sep 2021	<a href="#">Support of Salesforce API Version with Composer</a>	Updated topic as part of Summer21.09.08 Patch.
	 01 Sep 2021	<a href="#">Support of Salesforce API Version with Composer</a>	Updated topic as part of Summer21.09.01 Patch.
	 06 Aug 2021	<a href="#">Add Conga Composer IP Addresses</a>	New topic
		<a href="#">Support of Salesforce API Version with Composer</a>	New topic
	 07 Jul 2021	<a href="#">Supported Parameters for the Composer Lightning Component</a>	Modified Topic with more supported parameters.
		<a href="#">Integrating Composer with DocuSign App Launcher Package</a>	New topic
		<a href="#">Arranging Images Horizontally in Word</a>	New topic
		<a href="#">Using Composer Advanced Features</a>	New topic

## Spring '21

Document	Topic	Description
Spring '21 Rev D	<a href="#">Conga Composer Solution Migration Utility Tool</a>	Updated topic as part of Spring21.04.14 Patch.



Document	Topic	Description
Spring '21 Rev C	<a href="#">PDFVersion</a>	New Parameter
Spring '21 Rev B	<a href="#">CSMultiDocs</a>	New Parameter
Spring '21 Rev A	<a href="#">Transitioning from Courier to a Composer and Batch Solution</a>	New topic
Spring '21	<a href="#">Supported Parameters for the Composer Lightning Component</a>	Updated topic with more supported parameters.
	<a href="#">Supported Languages for Composer Lightning Component UI</a>	New topic
	<a href="#">Supported Languages for Document Automation Wizard</a>	New topic
	<a href="#">Supported Languages for Composer Setup Menu</a>	New topic
	<a href="#">Composer Authorization and Permission Matrix</a>	New topic

**Winter '20**

Document	Topic	Description
Winter 2020 Rev D	<a href="#">How to Integrate Conga Composer in Salesforce Communities</a>	Updated topic
	<a href="#">Composer Activity Report Overview</a>	
	<a href="#">Setting Up Amazon SES Integration for Conga Composer</a>	
	<a href="#">TPV Parameter</a>	
	<a href="#">DSI Parameter</a>	

Document	Topic	Description
Winter 2020 Rev C	<a href="#">CSSenderReturnUrl</a>	New topic
Winter 2020 Rev B	<a href="#">Product Data Limits</a>	Updated topic as part of Winter20.01.13 Patch.
	<a href="#">About Conga Batch</a>	
	<a href="#">Accommodating Large Batch Sizes for a Consolidated output with Conga Batch</a>	
	<a href="#">Conga Batch file size limits</a>	
Winter 2020 Rev A	<a href="#">Entering a SendGrid API Key</a>	New topic introduced as part of Winter20.12.02 Patch.
	<a href="#">SendGrid Integration Parameters</a>	Updated topic as part of Winter20.12.02 Patch.
Winter 2020	<a href="#">Conga Composer Lightning Component FAQs</a>	Updated topic
	<a href="#">Supported Parameters for the Composer Lightning Component</a>	New topic
	<a href="#">Overview of External Data</a>	Updated topic
	<a href="#">Setup and Configure External Data Sources</a>	Updated topic
	<a href="#">Build Your Conga Query to Retrieve External Data</a>	Updated topic
	<a href="#">Build Your External Data Composer Solution</a>	Updated topic
	<a href="#">xid (parameter)</a>	Updated topic
	<a href="#">Amazon SES Integration</a>	New topic

Document	Topic	Description
	<a href="#">Setting up Amazon SES Integration for Conga Composer</a>	New topic

# Composer for Administrators

This section describes how *Conga Composer* explains works and how to install and configure Conga Composer to create solutions.

Topic	Description
What's Covered	This documentation walks the administrators through the process of Composer administration. It provides conceptual information, set-by-step instructions for Composer administrative tasks.
Primary Audience	Administrator
IT Environment	Refer to the latest <i>Conga Composer Release Notes</i> for information on System Requirements and Supported Platforms.
Updates	For a comprehensive list of updates to this documentation for each release, see <a href="#">What's New in Composer Documentation</a> .
Other Resources	Refer to <i>Conga Composer Release Notes</i> for information on new features and enhancements, fixed issues, and known issues for a specific release.

This section describes the following tasks and topics:

- Getting Started with Composer
- Conga Composer Installation and Setup
- Building Composer Solutions
- Gathering Data for Composer
- Creating Composer Templates
- Conga Composer Parameter Basics
- Advanced Composer Solutions
- Composer Enterprise Edition
- Composer Troubleshooting Help
- Composer Error Messages
- Conga Trigger
- Conga Batch
- Conga QuickMerge

Before using Conga Composer, you must be familiar with the following:

- [Basic Salesforce administration](#)
- [Salesforce and Conga terms and definitions](#)

If you are new to Conga Composer, begin here: [Get started](#) and [About Conga Composer](#).

Select one of the following topics for more information:

- [Getting Started with Composer](#)
- [Building Composer Solutions](#)
- [Gathering Data for Composer](#)
- [Creating Composer Templates](#)
- [Conga Composer Parameter Guide](#)
- [Advanced Composer Solutions](#)
- [Composer Enterprise Edition](#)
- [Composer Troubleshooting Help](#)
- [Composer Error Messages](#)
- [Conga Trigger](#)
- [Conga Batch](#)
- [Conga Mail Merge](#)
- [QuickMerge Product Guide](#)

## Getting Started with Composer

Select one of the following topics to get up and running with Conga Composer®.

- [Composer Basics](#)
- [Installing Composer](#)
- [Configuring Composer](#)

## Composer Basics

Basic information about Conga Composer®.

- [Composer Product Guide](#)
- [What Does Conga Composer Do?](#)
- [Feature Comparison Between Conga Composer Release 7 and 8](#)
- [About Conga Composer Solution Manager](#)
- [Anatomy of a Conga Composer Button](#)
- [Composer Quick Starts](#)
- [License Provisioning Tips](#)
- [Product Overview Video](#)

- [Product Data Limits](#)
- [Supported Culture Codes](#)

## Composer Product Guide

### Learn Conga Composer Basics

- [About Conga Composer](#)
- [What Does Conga Composer do?](#)
- [Anatomy of a Conga Composer Button](#)

### Install Conga Composer

- [Technical Requirements](#)
- [Install Conga Composer from the Salesforce AppExchange](#)
- [What's in the Managed Package?](#)

### Configure Conga Composer

- [Basic Settings](#)
- [Configure the Composer Connected App](#)
- [Enable Remote Sites](#)
- [Prepare Composer Settings for Use in a Sandbox](#)
- [Assign User Licenses](#)
- [Profile Permissions Needed to Use Conga Composer](#)

### Use Conga Composer

1. Create your first solution with the [Document Automation Wizard \(DAW\)](#).
2. Create and Manage Solutions Using [Solution Manager](#).
  - [Using Solution Manager with Salesforce Professional Edition](#)
  - [Create and Manage Solutions Manually](#)
  - [Global Merge with Solution Manager](#)
3. Gather Data for Composer.
  - [Gather Salesforce Data](#)
  - [Use Salesforce Reports with Conga Composer](#)
  - [Use SOQL Queries with Conga Composer](#)
4. Create Templates for Composer.

- [Word Templates](#)
  - [PowerPoint Templates](#)
  - [Excel Templates](#)
  - [PDF Templates](#)
  - [HTML Email Templates](#)
5. [Customize Composer with Parameters.](#)
- [Composer Parameter Guide](#)

## Troubleshoot Conga Composer

- [Conga Composer Debug URL](#)
- [Product Data Limits](#)
- [Composer Troubleshooting Help](#)

## Automate Conga Composer


- [Batch Automate with Conga Batch](#)
- [Event-based Automation with Conga Trigger](#)

## Composer Add-On Products

- [Conga Mail Merge](#)
- [Conga QuickMerge](#)

## What Does Conga Composer Do?

Conga Composer makes it easy to create and deliver sophisticated documents, presentations, and reports from Salesforce by automatically populating richly formatted templates with data from any standard or custom object.

 Solution Manager is included automatically in Conga Composer release 8.7 and above. Solution Manager is an intuitive interface that makes building solutions much easier. Release 8.7 also introduced Solution Packs, ready-made solutions that enable admins to produce and release a solution in minutes. For more information, see [Creating and Managing Solutions using Solution Manager](#) and [Build a Solution for Me: Solution Packs](#).

- Most Composer solutions use a custom button or link on a Salesforce object. This button or link will retrieve all of the standard and custom fields on that object and make them available to merge into your document.

- The object where you place your button or link is called the Master Object.
- Start building the button or link in Salesforce Setup on a Master Object's Buttons, Links, and Actions menu.
- Pick and choose which fields you need and how you want them formatted so your document is fully customized to your requirements.
  - Conga's Template Builder provides an easy drag and drop or copy/paste mechanism to add Salesforce fields to your document or email template.
  - The Template Builder is located in the Tools & Settings menu of the Conga Composer user interface.
- Build the Composer custom button or link to launch the Composer user interface.
- Use Salesforce reports or SOQL queries to add more data to your solution - fields from objects related to the object where your button or link is placed.
  - Build a Salesforce report in Salesforce's report builder and filter the report by the Master Object ID so it will retrieve the related data (e.g. child object records) specific to the Master Object record from which you launch Composer.
  - Add the report to your Composer solution by pasting the Report ID into the Composer button or link URL.
  - SOQL queries can be used in addition to reports to retrieve related data.
- Add parameters to customize the behavior of your solution so you can automatically log activities in Salesforce, save a copy of your merged output file, integrate with Conga's technology partners, along with dozens of other actions.
  - Explore what you can do with [parameters](#).
- Automate your Composer solution with our add-on products, Conga Batch® for batch processing and Conga Trigger® for Salesforce Conga Trigger-initiated solutions.
  - Explore our add-on products [Conga Batch](#) and [Conga Trigger](#).

## Create

Conga Composer offers document generation capabilities to create virtually any important document, report, or presentation using Salesforce data. Placing a custom button or link on any object's page layout gives you immediate access to all its fields. Salesforce reports or SOQL queries can retrieve any additional required data from as many as 50 other objects or lists.

Armed with the data you need, Composer gives you the ability to build richly formatted templates that are completely customized to your business. Composer provides easy access to available data, fields are mapped directly to your template, and then the built-in template manager stores them in Salesforce for user selection on pre-defined routines.



Enjoy the flexibility of personalized templates while producing consistent output results every time.

## Distribute

Conga Composer offers a variety of options for downloading, storing, and delivering your output files:

- Immediately downloaded
- Saved directly in Salesforce, Google Drive®, or SpringCM®
- Automatically emailed
- Sent for eSignature

Users can create documents, presentations, and reports from their web browser, mobile device, or both. Flexible activity logging allows you to create completed tasks, instantly update fields, assign follow-up tasks to specific users and save copies of your output in Salesforce.

Whether you're automatically emailing invoices to customers, posting quarterly reports to Chatter, or sending a contract for eSignature, Conga Composer makes file delivery part of a seamless overall process. Simplify document creation and delivery with Composer to eliminate errors, increase productivity and user adoption.

## Automate

Composer doesn't just produce and deliver documents – it automates entire business processes. Use parameters to control the behavior of your Composer button and streamline the user experience. You have a wide variety of options – preselect up to 10 templates to generate one or more output files, use parameters to restrict options in the user interface, or enable background mode to bypass it entirely.

With the ability to predefine everything from templates and email recipients to activity logging and output file storage, Conga Composer transforms complex routines into single button clicks. Tackle unlimited use cases with a single application, increasing user adoption across your entire organization. Generate thousands of documents per day without increasing staff resources.


Composer also offers powerful add-ons for running solutions in batches, scheduling the creation and delivery of files, or triggering a merge with Salesforce workflow.

## Feature Comparison Between Conga Composer Release 7 and 8

### Conga Composer® Feature Comparison.

Feature	Release 8	Release 7
User Interface	Refined UI provides a more intuitive user experience	Original UI - orange and gray color scheme
Solution Packs	Pre-built solutions that can be deployed in minutes.	Unsupported
Solution Manager	<p>An intuitive way to build solutions from one screen, including:</p> <ul style="list-style-type: none"> <li>• Add reports and queries</li> <li>• Add document and email templates</li> <li>• Add parameters and view parameter definitions</li> <li>• Launch Template Builder</li> <li>• Select a sample record and test the solution before adding it to a page layout</li> <li>• Create Global Merge solutions that are not tied to a single master object</li> </ul>	Unsupported

Feature	Release 8	Release 7
Template Builder	Enhanced template builder makes creating and maintaining templates quicker and easier <ul style="list-style-type: none"> <li>• Automatically format fields according to template type selected</li> <li>• Ability to drag and drop fields into a template</li> <li>• Create a file template directly from the Template Builder</li> </ul>	Copies field name
Microsoft Office Versions Supported	Office 2016®/ Office 365® and earlier	Microsoft Office 2010 and previous*
Word® Template Merge Fields	Text-Based and Traditional	Traditional
HTML Email Editor	Advanced HTML email editor provides more formatting control	Original
Number of Rows Allowed per Salesforce® Report or SOQL Query	5,000	1,500
Google Drive® Integration	Supported	Unsupported
Updates and New Features Released Regularly	Yes - updated continuously	No - maintained as-is

 \* Initial template creation must be done in Microsoft Office 2010 or previous; however, template editing is supported in Microsoft Office 2013 and previous.

## About Conga Composer Solution Manager

Conga Composer Solution Manager dramatically accelerates the creation of document automation solutions. It is a one-stop shop for creating new Conga Composer solutions which allow Administrators to easily add reports, queries, parameters, and templates all within one tab.


With an intuitive interface to build Conga Composer solutions, the Conga Composer Solution Manager dramatically accelerates the creation of document automation solutions. Administrators can simply gather the required information for the solution and enter it into a familiar Salesforce® page layout, with menu-driven options for defining the desired output, distribution method, and other customized behaviors. It is a one-stop shop for creating new Conga Composer solutions which allow Administrators to easily add reports, queries, parameters, and templates all within one tab.

Solution Manager for Conga Composer is intended to:

- Provide an easier way to create new Conga Composer solutions.
- Jump start new, advanced Conga Composer solutions.
- Provide an organizational tool for the components of a Conga Composer button.

It is not intended as:

- A replacement for the standard way of creating complex Conga Composer solutions.
- A way to manage existing Conga Composer solutions.

 You cannot reference merge fields that are not on the Master Object in a parameter or User object. In addition, you can add formula syntax, but you must know the correct syntax; the Solution Manager cannot access the Salesforce Function editor.

## Anatomy of a Conga Composer Button

All Conga Composer URLs start the same way. The first three lines are identical in every Composer URL, and the fourth line is customized to identify the master object from which you'll be gathering data and running the solution.

There are two types of Composer buttons: **non-OAuth** and **OAuth** (Open Authorization).

### Non-OAuth Buttons

All non-OAuth Conga Composer® URLs start the same way. The first three lines are identical in every Composer URL, and the fourth line is customized to identify the Master Object from which you'll be gathering data and running the solution. These first lines are referred to as the Base URL, which looks like this:

```
https://composer.congamerge.com
?SessionId={!API.Session_ID}
&ServerUrl={!API.Partner_Server_URL_210}
&Id=
```

## OAuth Buttons

OAuth (Open Authorization) is a different security protocol that is required for all Conga Composer buttons created in the Salesforce Lightning user interface. The Base URL of an OAuth button looks like this:

```
/apex/APXTConga4__Conga_Composer
?serverUrl={!API.Partner_Server_URL_370}
&id=
```

In both button types, the `&id=` parameter (or `&id=` in the OAuth button) defines your master object - the Salesforce® object from which you will launch your Composer solution. The value of this parameter will always be the ID of your master object. For example:

To launch the solution from this type of object:	The fourth line would contain:
Account	<code>&amp;id={!Account.Id}</code>
Opportunity	<code>&amp;id={!Opportunity.Id}</code>
Case	<code>&amp;id={!Case.Id}</code>
A custom object	<code>&amp;id={!CustomObjectName_c.Id}</code>

 Conga Composer includes sample buttons on all standard objects.

## Composer Parameters

After the Base URL, you will see the Composer parameters that have been added to the Composer solution (assuming they have been added by the Conga Admin). Parameters are settings that control the behavior of your solution. They always use the following convention: `&ParameterName=Value`

In the example below, you can see seven parameters have been added to this Composer button URL:

```
https://composer.congamerge.com?SolMgr=1
&sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&Id={!Opportunity.Id}
&TemplateId=a0436000001AbCp
&CongaEmailTemplateId=a0236000001gl7C
&TemplateGroup=Invoice
```

```
&DefaultPDF=1  
&OFN=Conga+Invoice+++{!Opportunity.Name}+++{!Today}  
&SCO=1  
&SCI=Attachments
```


The parameters in this example control template selection, output file type, output file name, and where to save a copy of the output document. For more information about parameters, see [Customizing Composer with Parameters](#) and the [Parameter Guide](#).

## How to Populate Parameter Values

There are three ways to populate parameter values:

1. Type the value into the parameter (using Solution Manager or manually).
2. Using a Salesforce merge field (such as `{!Opportunity.Id}`), which will dynamically update with a value when the solution is run.
3. Using a Qvar, which allows you to retrieve data with a Conga Query and then pass that data to another parameter to use as the value.

## Composer Quick Starts


 Quick Starts are available in Composer managed package 8.15 and newer (see the [Product Install Links](#) to install or upgrade to Composer 8.15).

You can use the Quick Start wizard to get started with Composer solutions quickly. The wizard allows you to select and deploy pre-built solutions complete with queries, templates, and behavior settings that are already configured.

To access Composer Quick Start:


<b>Salesforce Classic</b>	Select <b>All Tabs (+)</b> → <b>Conga Quick Start</b> .
<b>Salesforce Lightning Experience</b>	Click the <b>Conga Quick Start</b> tab.

## Available Quick Start Solutions:


Solution Name	What does it do?	Additional information
Account Brief	This solution creates account briefs that use the Account Object. This template includes pre-built queries and Word templates.	<ul style="list-style-type: none"> <li>• Pre-selects template</li> <li>• Output as PDF</li> <li>• Saves a copy to Notes &amp; Attachments</li> <li>• Logs an Activity</li> <li>• Includes Account Name and current date in the output file name</li> </ul>
Client Letter	This solution creates client letters that use the Contact Object. This template includes a pre-built Word template.	<ul style="list-style-type: none"> <li>• Pre-selects template</li> <li>• Output as PDF</li> <li>• Save a copy to Chatter</li> <li>• Includes Contact Name and current date in the output file name</li> </ul>
Sales Proposal	This solution creates sales proposals that use the Opportunity and Opportunity line items. This template includes pre-built Word and email templates.	<ul style="list-style-type: none"> <li>• Pre-selects Proposal template and Email template</li> <li>• Output as PDF</li> <li>• Saves a copy to Chatter</li> <li>• Delivers via Email</li> <li>• Includes Opportunity Name and current date in the output file name</li> </ul>
Sales Quotes	<p>This solution creates sales quotes that use the Quote Object. The template includes a pre-built Excel template.</p> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p> The Quote object must be enabled for this solution.</p> </div>	<ul style="list-style-type: none"> <li>• Pre-selects template</li> <li>• Saves a copy to Notes &amp; Attachments</li> <li>• Logs an Activity</li> <li>• Includes Quote Name and current date in the output file name</li> </ul>

For more information on templates, see the topic: [Use a Template from the Template Library](#).

After you select a template, the template is created on your Conga Solutions tab. A button is created for you on the related Object. To use it, add the button to the page layout. To learn more about adding or removing buttons, see [Add or Remove Buttons on Page Layouts](#).

 You can use the wizard as many times as you want to deploy one or all of the Quick Start solutions.

## License Provisioning Tips

 Conga cannot provision your licenses unless you install Composer. You must install it for your subscription to be provisioned. If you are holding off on installation for any reason, the best practice would be for you to submit a case when you install to ensure timely provisioning.

## What Do I Have Access to When I First Install Composer?

- All new Composer License installs in Production Orgs are provisioned as a 30-day trial with access for 10 Users.
- All new Composer License installs in Sandboxes are provisioned as a 30-day trial with a Site License and access for all Users with access to the Sandbox.
- Beginning in June 2019, when you refresh a sandbox or install Composer in a Sandbox, the subscription does not expire.
- If you installed Composer in a sandbox prior to June 2019 and the license expires, submit a case to prevent a sandbox Composer license from expiration moving forward. If you are a Conga Communities User, submit the case through [Conga Communities](#).
- If you are an Administrator in your org, submit a case to request a Conga Communities license.
- It is a best practice to submit Composer license extension cases if you see an expiration warning less than one year away.

## Why does my Sandbox have an expiration countdown, but my Production Org doesn't?

- All sandbox orgs are provisioned as trial licenses and display expiration warnings (often years in the future). Production orgs are provisioned per your specific



subscription term and only display the license expiration date 14 days prior to expiration.

## I just installed Composer and only see 10 licenses but I purchased more than that, help!

- When you install, you are allotted 10 licenses initially for a 30-day trial period. If you are waiting on your additional licenses, then the fastest way to ensure that all of your licenses are provisioned is to submit a case.

## I don't see "Manage Licenses" in my Sandbox, like I do in Production. Why?

- We do not enable License Management Application (LMA) in Sandboxes unless you ask specifically. You have an unlimited Site License in all sandboxes. This means that you do not have to assign a Salesforce user a license for Composer.
- If you want this enabled, remember to add all of the users who need access. You can submit a case on the support page and request LMA be turned on.

## How do I enable License Management Application in my Org?

- License Management Application is a Salesforce application that allows you to manage licenses in your Installed Packages.
- LMA is disabled by default in Sandboxes for all Conga Products.
- If you are unable to Manage Licenses, it is likely that LMA is not enabled.
  - Ensure that more users are not using Composer than you are licensed for to avoid potential overages.
- If you want to turn on License Management and enable the feature to Assign Licenses, no Salesforce users can use Composer until you assign licenses.
- Please submit a Case to enable LMA ([Assign User Licenses](#)).

## How Do I Assign Users Licenses?

- Please click here for instructions: [Assign User Licenses](#).

## I Just Renewed But I Still see an Expiration Warning.

- If you have signed your renewal order form, and your expiration warning is less than 2 weeks away, please submit a case.

I just refreshed my Sandbox but I am not seeing Conga Batch and Conga Trigger enabled, even though I purchased these products.

- Please submit a case and the Provisioning team will enable these products.
- As a best practice, you can follow these instructions about [Preparing Conga Composer Settings for use in a Sandbox](#) to prevent needing to submit a case.

## Does Conga Use License Keys?

- Conga does not use license keys. All new installs for Conga Sign, Contracts for Salesforce Conga Grid will be assigned with 5 Trial licenses for 15 days. All new Composer installs will come with 10 licenses for 30 days.
- All products can be downloaded via the Salesforce AppExchange. Exceptions to this are Conga Orchestrate, and Conga Contracts (off platform), and Contracts for Salesforce.
- Here is [a link](#) to the Conga Installation Links.
- Conga Orchestrate, Conga Contracts, and Contracts for Salesforce will be installed by your Conga Professional Services team member as part of your Professional Services engagement. (If this does not apply to you, then please contact your Conga account representative.)

## Product Overview Video

Click [here](#) to view the Conga Composer Overview video.

## Product Data Limits

The below table summarizes the limits for each product in the Composer family.

### Conga Composer 8

Conga Composer® 8	Business Edition	Enterprise Edition
Reports per Composer solution (tabular and summary reports <i>only</i> )	50	100
Rows per Report	5,000	5,000

<b>Conga Composer® 8</b>	<b>Business Edition</b>	<b>Enterprise Edition</b>
Rows per Composer solution (all report(s) and query(ies) combined)	25,000	50,000
Queries per Composer solution	50	100
Rows per Query	5,000	25,000
Template size	10 MB	10 MB
Number of template(s) per operation	20	40
Number of attachment(s) per operation	150	150
Aggregate template size (when specifying multiple templates to use in the merge process)	18 MB	18 MB
Aggregate output size	40 MB	40 MB
Columns per worksheet in View Data workbook	256	256
Querystring variable	6	6

## Conga Composer Release 7

<b>Conga Composer® 7</b>	<b>Maximum</b>
Queries per Composer solution	50
Reports per Composer solution	50
Lines of Data in a report	1,500 Rows or 5 MB/merge
Columns per tab in View Data workbook	256
Dataset	25,000 total rows

Conga Composer® 7	Maximum
Qvar	50 rows of data
Number of Templates per operation	20
Template size	5 MB

## Conga Mail Merge

Conga Mail Merge® Limits	Maximum
Reports per Mail Merge solution	1
Rows per Report	5,000
Template size	5 MB
Local Template size	2 MB
Output file size	40 MB
Documents per batch (master records)	1,500

**Note:**

Mail Merge is a legacy product and intended for Simple documents. The inclusion of Images and Graphics in your template decreases the amount of Master Records that are processed in a Batch.

## Conga Query Manager

Conga Query Manager	Maximum
Rows from an aggregated (SUM, COUNT, etc.) query	1,000
Rows of data from a query variable (QVar)	50

## Conga Batch

Conga Batch Limits	Maximum
Records in a batch	200 Records for consolidated output. 1000 Records for distributed output.

**Note:**

Batch output to Google Drive, OneDrive, Box, and BigTinCan is limited to 200 records for consolidated output and 1000 records for distributed output.

## Conga Trigger

Conga Trigger®	Maximum
Transactions	500 per day

**Note:**

The 500 per day maximum can be increased upon request.

## Browser URL Limits

Browser URL Limits	Maximum
Internet Explorer	2000 bytes (closer to 1,700 bytes in practice)
Chrome, Firefox, Safari	64 KB

## Supported Culture Codes

English Name	Abbreviation
For all supported locales as well as examples of display formats for each, visit <a href="https://help.salesforce.com/apex/HTViewHelpDoc?id=admin_supported_locales.htm">https://help.salesforce.com/apex/HTViewHelpDoc?id=admin_supported_locales.htm</a>	
Afrikaans (South Africa)	af-ZA

English Name	Abbreviation
Albanian (Albania)	sq-AL
Amharic (Ethiopia)	am-ET
Arabic (Algeria)	ar-DZ
Arabic (Bahrain)	ar-BH
Arabic (Egypt)	ar-EG
Arabic (Iraq)	ar-IQ
Arabic (Jordan)	ar-JO
Arabic (Kuwait)	ar-KW
Arabic (Lebanon)	ar-LB
Arabic (Libya)	ar-LY
Arabic (Morocco)	ar-MA
Arabic (Oman)	ar-OM
Arabic (Qatar)	ar-QA
Arabic (Saudi Arabia)	ar-SA
Arabic (Syria)	ar-SY
Arabic (Tunisia)	ar-TN
Arabic (U.A.E.)	ar-AE
Arabic (Yemen)	ar-YE
Armenian (Armenia)	hy-AM

English Name	Abbreviation
Assamese (India)	as-IN
Bashkir (Russia)	ba-RU
Basque (Basque)	eu-ES
Belarusian (Belarus)	be-BY
Bengali (Bangladesh)	bn-BD
Bengali (India)	bn-IN
Breton (France)	br-FR
Bulgarian (Bulgaria)	bg-BG
Catalan (Catalan)	ca-ES
Chinese (Hong Kong S.A.R.)	zh-HK
Chinese (Macao S.A.R.)	zh-MO
Chinese (People's Republic of China)	zh-CN
Chinese (Singapore)	zh-SG
Chinese (Taiwan)	zh-TW
Corsican (France)	co-FR
Croatian (Croatia)	hr-HR
Croatian (Latin) (Bosnia and Herzegovina)	hr-BA
Czech (Czech Republic)	cs-CZ
Danish (Denmark)	da-DK
Dari (Afghanistan)	prs-AF

English Name	Abbreviation
Divehi (Maldives)	dv-MV
Dutch (Belgium)	nl-BE
Dutch (Netherlands)	nl-NL
English (Australia)	en-AU
English (Belize)	en-BZ
English (Canada)	en-CA
English (Caribbean)	en-029
English (India)	en-IN
English (Ireland)	en-IE
English (Jamaica)	en-JM
English (Malaysia)	en-MY
English (New Zealand)	en-NZ
English (Republic of the Philippines)	en-PH
English (Singapore)	en-SG
English (South Africa)	en-ZA
English (Trinidad and Tobago)	en-TT
English (United Kingdom)	en-GB
English (United States)	en-US
English (Zimbabwe)	en-ZW



English Name	Abbreviation
Estonian (Estonia)	et-EE
Faroese (Faroe Islands)	fo-FO
Finnish (Finland)	fi-FI
French (Belgium)	fr-BE
French (Canada)	fr-CA
French (France)	fr-FR
French (Luxembourg)	fr-LU
French (Principality of Monaco)	fr-MC
French (Switzerland)	fr-CH
Frisian (Netherlands)	fy-NL
Galician (Galician)	gl-ES
Georgian (Georgia)	ka-GE
German (Austria)	de-AT
German (Germany)	de-DE
German (Liechtenstein)	de-LI
German (Luxembourg)	de-LU
German (Switzerland)	de-CH
Greek (Greece)	el-GR
Greenlandic (Greenland)	kl-GL
Gujarati (India)	gu-IN

English Name	Abbreviation
Hebrew (Israel)	he-IL
Hindi (India)	hi-IN
Hungarian (Hungary)	hu-HU
Icelandic (Iceland)	is-IS
Igbo (Nigeria)	ig-NG
Indonesian (Indonesia)	id-ID
Irish (Ireland)	ga-IE
isiXhosa (South Africa)	xh-ZA
isiZulu (South Africa)	zu-ZA
Italian (Italy)	it-IT
Italian (Switzerland)	it-CH
Japanese (Japan)	ja-JP
Kannada (India)	kn-IN
Kazakh (Kazakhstan)	kk-KZ
Khmer (Cambodia)	km-KH
Kinyarwanda (Rwanda)	rw-RW
Kiswahili (Kenya)	sw-KE
Korean (Korea)	ko-KR
Kyrgyz (Kyrgyzstan)	ky-KG

English Name	Abbreviation
Lao (Lao P.D.R.)	lo-LA
Latvian (Latvia)	lv-LV
Lithuanian (Lithuania)	lt-LT
Luxembourgish (Luxembourg)	lb-LU
Macedonian (Former Yugoslav Republic of Macedonia)	mk-MK
Malay (Brunei Darussalam)	ms-BN
Malay (Malaysia)	ms-MY
Malayalam (India)	ml-IN
Maltese (Malta)	mt-MT
Maori (New Zealand)	mi-NZ
Mapudungun (Chile)	arn-CL
Marathi (India)	mr-IN
Nepali (Nepal)	ne-NP
Norwegian, Bokmål (Norway)	nb-NO
Norwegian, Nynorsk (Norway)	nn-NO
Occitan (France)	oc-FR
Oriya (India)	or-IN
Pashto (Afghanistan)	ps-AF
Persian (Iran)	fa-IR
Polish (Poland)	pl-PL

English Name	Abbreviation
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-PT
Punjabi (India)	pa-IN
Romanian (Romania)	ro-RO
Romansh (Switzerland)	rm-CH
Russian (Russia)	ru-RU
Sami (Northern) (Finland)	se-FI
Sami (Northern) (Norway)	se-NO
Sami (Northern) (Sweden)	se-SE
Sanskrit (India)	sa-IN
Setswana (South Africa)	tn-ZA
Sinhala (Sri Lanka)	si-LK
Slovak (Slovakia)	sk-SK
Slovenian (Slovenia)	sl-SI
Spanish (Argentina)	es-AR
Spanish (Bolivia)	es-BO
Spanish (Chile)	es-CL
Spanish (Colombia)	es-CO
Spanish (Costa Rica)	es-CR

English Name	Abbreviation
Spanish (Dominican Republic)	es-DO
Spanish (Ecuador)	es-EC
Spanish (El Salvador)	es-SV
Spanish (Guatemala)	es-GT
Spanish (Honduras)	es-HN
Spanish (Mexico)	es-MX
Spanish (Nicaragua)	es-NI
Spanish (Panama)	es-PA
Spanish (Paraguay)	es-PY
Spanish (Peru)	es-PE
Spanish (Puerto Rico)	es-PR
Spanish (Spain)	es-ES
Spanish (United States)	es-US
Spanish (Uruguay)	es-UY
Spanish (Venezuela)	es-VE
Swedish (Finland)	sv-FI
Swedish (Sweden)	sv-SE
Tamil (India)	ta-IN
Tatar (Russia)	tt-RU
Telugu (India)	te-IN

English Name	Abbreviation
Thai (Thailand)	th-TH
Tibetan (People's Republic of China)	bo-CN
Turkish (Turkey)	tr-TR
Turkmen (Turkmenistan)	tk-TM
Uighur (People's Republic of China)	ug-CN
Ukrainian (Ukraine)	uk-UA
Urdu (Islamic Republic of Pakistan)	ur-PK
Vietnamese (Vietnam)	vi-VN
Welsh (United Kingdom)	cy-GB
Wolof (Senegal)	wo-SN
Yi (People's Republic of China)	ii-CN
Yoruba (Nigeria)	yo-NG

## Installing Composer

Information about installing Composer.

- [Installing Composer from the Salesforce AppExchange](#)
- [What's in the Composer Package?](#)
- [Conga Composer Technical Requirements](#)
- [Upgrade Conga Composer to Release 8](#)
- [Update the Conga Composer Button URL from Release 7 to Release 8](#)
- [Conga Composer Test Guide](#)
- [Find Your Conga Buttons](#)

# Installing Composer from the Salesforce AppExchange

Conga Composer is available to install from the Salesforce AppExchange.

## Prerequisites

- You must provide System Administrator credentials to install an application from the AppExchange.
- If upgrading from an earlier version of Conga Composer, [read these notes](#). You might also have to update the button URL for all Conga Composer Release 7 solutions.

## To install Conga Composer

1. Install Conga Composer into your Salesforce instance by [visiting our listing on the AppExchange](#).
2. Click **Get It Now** and follow the on-screen prompts to install Conga Composer.
3. Log in to the AppExchange.
4. Specify where you want to install Conga Composer.
  - Click Install in production to install in your production instance.
  - Click Install in sandbox to install in a sandbox.
5. Read and accept the terms and conditions.
  - Select the I have read and agree to the terms and conditions box.
  - Click Confirm and Install.
6. As a best practice, and to avoid access issues for end-users Conga recommends selecting Install for All Users. Then click Install.

**⚠ FOR PAID SUBSCRIBERS ONLY** - Email [onboarding@conga.com](mailto:onboarding@conga.com) when you have installed Composer to ensure the fastest provisioning of any additional licenses, and our Composer automation products. This step is not necessary if you are trialing Conga products.


## Post-Installation

- After installing, you must [configure the Composer application](#), including integration settings for third-party applications according to your business use case.
- (Optional) [Install Conga Trigger](#)
- (Optional) [Install Conga Batch](#)

## What's in the Composer Package?

In the managed package you download, you'll find the Conga Composer application, including:

- Conga Solution Manager
  - a set of custom objects used by the Solution Manager. These custom objects do not count against your Salesforce® limit on custom objects.
- Conga Composer SF1
  - a connected app to allow Conga Composer to operate on the Salesforce1® mobile platform.
- Conga Global Merge
  - a custom object into which Global Merge solutions are stored. This custom object does not count against your Salesforce limit on custom objects.
- Conga Template Manager
  - a custom object into which Conga Templates are stored. This custom object does not count against your Salesforce limit on custom objects.
- Conga Email Templates
  - a custom object to build and store HTML email templates. This custom object does not count against your Salesforce limit on custom objects.
- Conga Query Manager
  - (a.k.a Conga Queries) – a custom object to build and store SOQL queries. This custom object does not count against your Salesforce limit on custom objects.
- Conga Composer Reports
- Custom Tabs
  - About Conga Composer, Conga Composer Setup, Conga Solutions, Conga Templates, Conga Email Templates, Conga Queries, Conga Resources, Conga Global Merges, Reports, and Custom Tabs.
- Conga Composer Custom Settings for Production Org ID, which is useful for customers who create Salesforce Sandbox instances and Server Endpoint URL for third-party data security services.

 In Release 8, Conga Mail Merge was removed from the Conga Composer managed package and is now available as a separate add-on product at no extra charge. No existing Conga Mail Merge solutions are removed or changed, but customers who use Conga Mail Merge must [install](#) a separate add-on to continue running solutions. In Release 8, Conga Query Manager is included in the Conga Composer managed



package. Customers who used Conga Query Manager previous to Release 8 see an additional Conga Queries tab after upgrading the managed package and can decide to either consolidate existing queries in the new tab or keep both tabs.

## Conga Composer Technical Requirements

To install and successfully use Conga Composer, you must have one of the following from each category:

### Salesforce® Edition

- Performance / Unlimited
- Enterprise
- Professional (limitations may exist)
- Developer

### Salesforce® Products

- Sales Cloud
- Service Cloud
- Force.com / App Cloud
- Health Cloud
- Financial Services Cloud
- Quote-to-Cash

### Internet Browsers (latest stable version of)

- Firefox®
- Chrome® (you may have to [enable redirects](#) from Composer in Chrome version 64 or higher)
- Internet Explorer® (version 11 or the latest version supported by Salesforce)
- Safari® (Mac only)
- Microsoft Edge

### Template Software

- Microsoft Office® 2016 and earlier (Mac or PC)
- Adobe Acrobat® XI and earlier, Acrobat Reader  
Adobe LiveCycle PDFs are not supported.

## Mobile

Compatible with Salesforce1 and all mobile browsers when set up per documentation.

## Upgrade Conga Composer to Release 8

Conga Composer Release 8 offers a wealth of enhancements and is backward compatible with previous releases of Composer. We recommend most customers upgrade to this latest release immediately. For a comparison of the features in Release 7 and the current release, see [Feature Comparison between Conga Composer Release 7 and 8](#).

Conga Composer is a composite application, which means that some parts of the system are installed into your Salesforce® instance in a managed package, however, most of the system runs on our servers. As a result, the portion of the system that runs on our servers does not require upgrading on your part. Updates to the servers are automatically made available at various times throughout the year.

**⚠ DO NOT UNINSTALL** the previous version of Conga Composer before upgrading to Release 8. If you uninstall before upgrading, you will lose all of your existing Conga Composer solutions and will have to rebuild each of them manually.

The portions of the system that are installed into your instance of Salesforce can be upgraded by visiting [our listing on the AppExchange](#), clicking Get it Now, and following the prompts to update the installed components. After completing the install prompts with the new package, be sure to [update your Conga Composer button URLs](#). For instructions on upgrading, [watch this video](#).

## Upgrading to the latest version of Composer

You can upgrade to the latest version of Composer whenever you are ready.

Before you begin:

- Do not uninstall any versions of Composer.
- If you are using Conga Batch, you receive an email reminding you to update your security token.

See the [Differences between Conga Composer versions](#) to learn how to check what version of Composer you are on.

You can see a list of the latest versions of Composer in the Release notes.

## To upgrade to the latest version

1. Open the AppExchange in Salesforce.
2. In the search bar, type Composer. Select Conga Composer under All Apps.
3. Click **Get it Now**.

## Update the Conga Composer Button URL from Release 7 to Release 8

When you upgrade from previous versions of Conga Composer® to Release 8, you must also update the Conga Composer button URLs to use the new user interface, template building options, and support tools. You can do this in a few minutes by copying and pasting.

 You must do this for each button you want to use in Release 8.

### Prerequisite

Upgrade the Conga Composer application to Release 8.

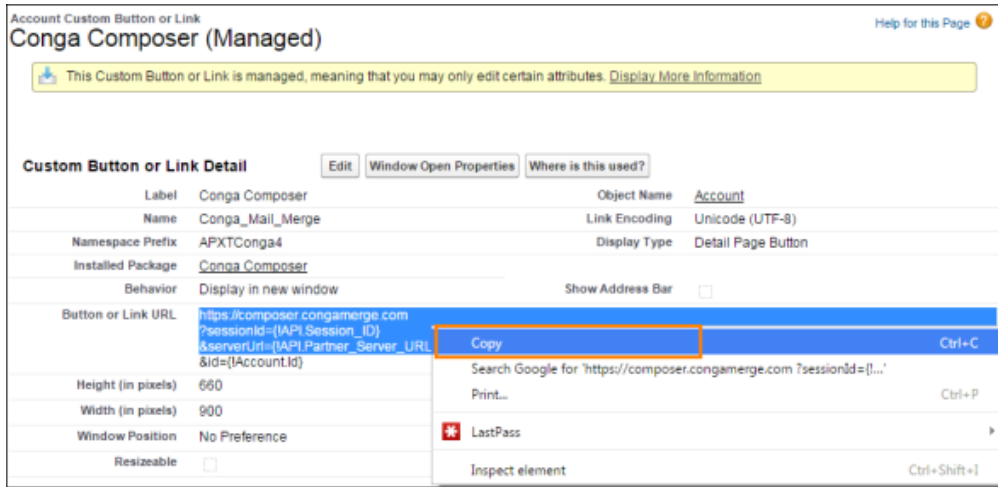
## Updating the Conga Composer Button URL from Release 7 to Release 8

To update your Conga Composer Button URL from Release 7 to Release 8:

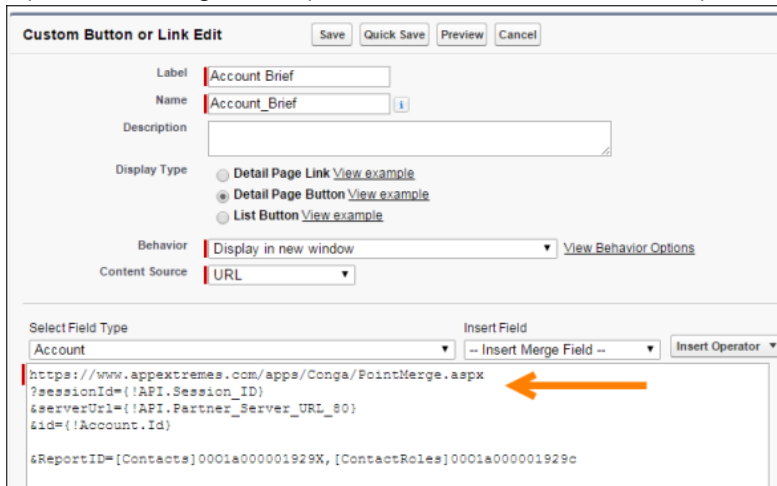
1. Navigate to a Conga Composer managed button (available on standard Salesforce® objects) and click **Edit**.

Edit	Accounts Tab	Tab	Standard Salesforce.com Page	<input type="checkbox"/>
Edit	 Adobe Sign Manager	EchoSign_Account_Publisher_Action	Custom Visualforce Account Assessment Action (Visualforce)	<input type="checkbox"/>
Edit   Del	Billing	Billing	Detail Page Link URL	<input type="checkbox"/>
Edit	Clean	ListClean	Standard Salesforce.com Page	<input type="checkbox"/>
Edit	 Conga Composer	Conga_Mail_Merge	Detail Page Button URL	<input type="checkbox"/>
<p>*** DO NOT USE THIS BUTTON ON A PAGE LAYOUT ***</p> <p>COPY THE URL FROM THIS BUTTON and paste it into your own custom button or link. Then add optional parameters to finely tune the button's behavior. See the Conga Configuration Guide for details.</p>				
<p>*** DO NOT USE THIS BUTTON ON A PAGE LAYOUT ***</p> <p>OAuth buttons require additional setup. Please see the Composer Installation Guide for more details.</p>				
Edit	 Conga Composer (OAuth)	Conga_Composer_OAuth	Detail Page Button URL	<input type="checkbox"/>
<p>COPY THE URL FROM THIS BUTTON and paste it into your own custom button or link. Then add optional parameters to finely tune the button's behavior. See the Composer Configuration Guide for details.</p>				

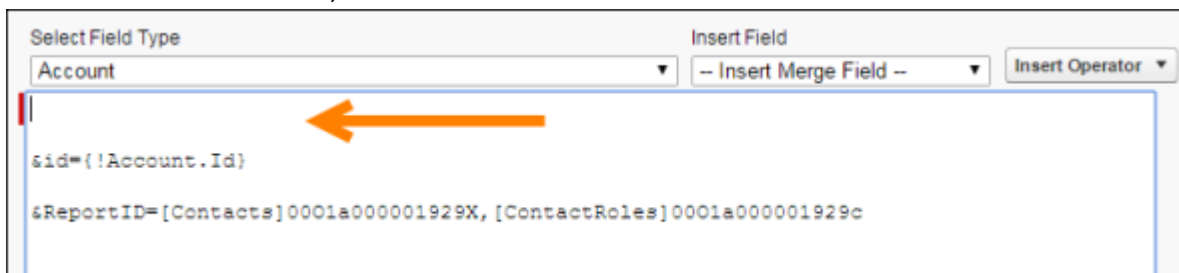
2. Highlight and copy the first three lines of the button URL.



3. Open the Conga Composer custom button to be updated and click **Edit**.

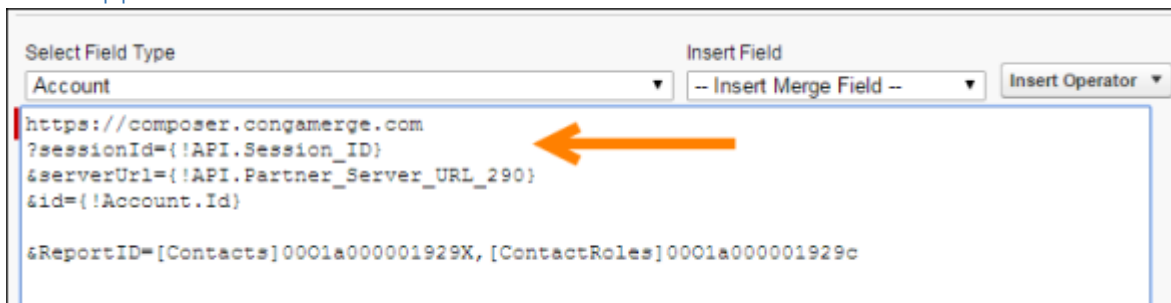


4. Highlight and delete the first three lines of the button URL (from "https" through the Partner Server URL field).

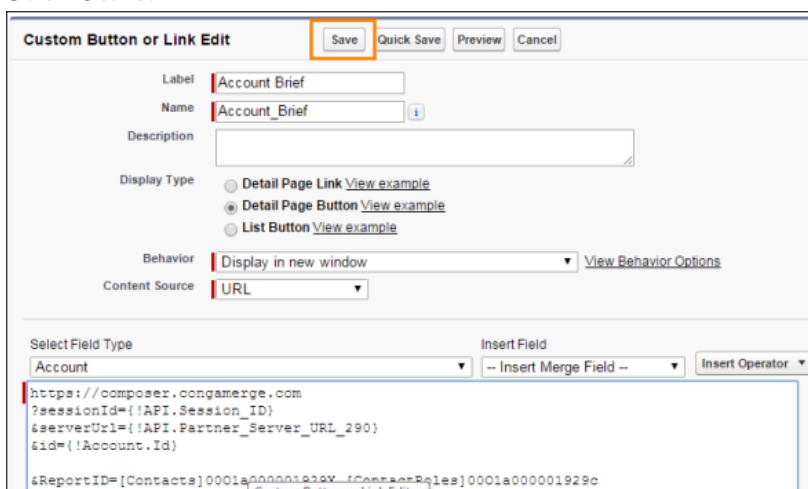


5. Paste the three lines of the button URL you copied above in step 2 at the beginning of this URL.

It should now begin with <https://composer.congamerge.com> instead of <https://www.appxtremes.com/>.



6. Click **Save**.



## Additional Details

Ensure there are no spaces in your button URL. They sometimes appear at the ends of the lines after copying and pasting. Spaces in the button URL will prevent Composer from working.

After saving, the updated button will now run on Composer Release 8, including the new user interface, template building options, and support tools.

## Conga Composer Test Guide

Use this Test Guide to test your Composer Release 7 solution to ensure it will work properly in Composer Release 8.

Conga Composer Test Guide

## Find Your Conga Buttons

By implementing Conga Solutions and following these steps, you will be able to find the Conga buttons and be up to speed with the latest and great Composer release.

1. Install the latest Composer package.
2. Once installed, go to navigate to Setup in Salesforce and search for **Custom Settings** (under Develop).
3. Click **Manage** next to Conga Solutions Settings.
4. Click **New**.
5. Name it "PROD", check on the check box and leave the endpoint URL alone.
6. Click **Save**, then go back to the Conga Solutions Setup tab.
7. When you find the Conga buttons, just click **Create Solution** and **Update** the button. Now the button is in the Conga Solutions user interface.
8. There is a new page below the remote sites that allows you to pick buttons and create solutions (it has to be a valid solution with good IDs, etc.) to work correctly.

## Configuring Composer


Information for configuring Conga Composer.

- [Configure the Conga Composer Connected App](#)
- [Add Conga Composer IP Addresses](#)
- [Configuring Solutions Setup \(Remote Sites\)](#)
- [Preparing Conga Composer Settings for use in a Sandbox](#)
- [Assign User Licenses](#)
- [Permissions Needed to Use Composer](#)
- [Composer Authorization and Permissions Matrix](#)
- [Basic Settings and Configuring Composer](#)
- [Conga Composer Setup Menu](#)
- [Enable Conga Composer Security Settings](#)
- [Create a Conga Composer OAuth Button](#)
- [OAuth for Conga Composer](#)
- [Composer for Salesforce Mobile Configuration](#)
- [Integration Partners Settings](#)
- [Composer Activity Report Overview](#)
- [Do I Need a Conga Composer User License During My Conga Trial?](#)
- [How Composer Sends Emails from Salesforce](#)
- [How to migrate Composer servers from NA to EMEA or APAC](#)

- [Migrating Conga Composer from a Salesforce Sandbox Instance to a Salesforce Production Instance](#)
- [Required Profile Permissions for Salesforce Winter '20 Critical Update](#)
- [Supported Composer User Interface Languages](#)
- [User Management Settings \(Legacy Document\)](#)
- [Where can I find Salesforce Org ID?](#)
- [Support of Salesforce API Version with Composer](#)


## Configure the Conga Composer Connected App

Administrators can pre-approve users to connect via OAuth when running solutions. This method masks the session ID and URL when using Visualforce® buttons, and prevents pop-up authorization requests on mobile devices when running Composer solutions on Salesforce Mobile.

 In addition to configuring Conga Composer as a Connected App, you will want to [Create Conga Composer OAuth buttons](#) (or update existing non-OAuth buttons) to launch your Composer solutions. This is not required in Salesforce Classic, but it is required in Salesforce Lightning.

## To configure Conga Composer as a connected app to use OAuth

1. Navigate to **Setup > Manage Apps > Connected Apps > Conga Composer**.
2. Click **Edit** next to the Conga Composer link.
3. Change the Permitted Users picklist to **Admin approved users are pre-authorized** and click **OK**.

 Admins must be pre-approved in the Connected App. If not, you may see an error message when trying to launch a Conga Composer OAuth button.

4. Click **Save**.
5. Click the Conga Composer link.
6. Scroll down to the Profiles section in the Conga Composer Connected App.
7. Designate which users connect to Conga Composer via OAuth by adding Profiles (click Manage Profiles) or Permission Sets (click Manage Permission Sets).

- ⚠ After setting the Permitted Users to be pre-authorized, administrators can add Profiles or Permission Sets in order to specify which users are approved (also include the Sys Admin profile when setting profiles).
- Pre-approved users will not see Allow or Deny prompts when accessing Composer via OAuth, and Conga Composer automatically creates a refresh token.
  - Users who are not part of the defined Profiles or Permission Sets will be denied access.

## Add Conga Composer IP Addresses

Conga recommends using the Relax IP setting for Conga Composer Connected App for the simplest installation and maintenance of the application. If you don't use the Relax IP setting, you must [whitelist the Composer IP addresses below to connect with Conga Composer](#) and complete the document merge.

If you restrict Profiles to specific IP Ranges, you must also include the Composer IP Addresses in each [Profile](#) that need to use Conga Composer.

### ⚠ Important

Composer IP addresses are subject to change, and the most recent list can be found [here](#). It is your responsibility to check back regularly for updates and to add or update the most recent IP addresses in your Salesforce org.

**Composer IP addresses to whitelist for the respective infrastructure:**

North America	Asia-Pacific	European Union
13.52.14.141	13.210.199.83	18.196.126.153
13.56.118.252	13.210.224.17	18.198.162.65
13.56.137.41	13.211.23.66	3.120.91.121
13.56.95.87	13.238.235.5	3.121.204.210
13.56.96.202	3.105.163.177	3.125.141.220




North America	Asia-Pacific	European Union
13.57.103.80	3.24.252.21	3.65.137.32
13.57.137.13	52.62.124.206	3.65.245.254
13.57.14.46	52.62.158.4	3.66.71.39
13.57.96.8	52.62.219.123	35.157.188.0
184.169.220.168	52.64.238.135	35.158.67.161
3.101.4.15	52.64.246.142	52.28.161.56
50.18.212.136	52.65.134.123	52.29.110.43
52.52.123.9	54.153.155.254	52.29.3.21
52.52.153.41	54.153.225.68	52.57.139.40
52.52.17.187	54.206.92.149	52.57.46.6
52.52.189.95		
52.52.241.229		
52.52.43.74		
52.52.58.21		
52.52.66.178		
52.52.9.16		
52.53.102.168		
52.53.111.230		
52.53.42.149		
52.53.49.72		

North America	Asia-Pacific	European Union
52.53.53.150		
52.53.54.93		
52.8.140.248		
52.8.220.110		
52.8.240.50		
52.8.58.54		
52.8.70.214		
52.8.70.253		
52.8.75.242		
52.9.109.249		
52.9.169.128		
52.9.169.217		
52.9.186.128		
52.9.246.74		
52.9.42.252		
52.9.83.156		
52.9.93.97		
54.151.58.224		
54.153.122.2		


North America	Asia-Pacific	European Union
54.176.239.143		
54.176.82.164		
54.177.123.97		
54.177.76.112		
54.183.137.163		
54.183.33.218		
54.183.38.181		
54.183.43.114		
54.183.48.15		
54.183.51.204		
54.193.185.79		
54.193.220.155		
54.193.245.248		
54.193.33.213		
54.193.50.119		
54.193.7.76		
54.193.75.202		
54.193.85.221		
54.215.176.149		
54.215.227.206		

North America	Asia-Pacific	European Union
54.219.139.180		
54.219.145.149		
54.219.153.223		
54.219.158.58		
54.241.218.240		
54.67.0.56		
54.67.48.248		

## To whitelist a Composer IP Address on your Salesforce Org

1. On the Salesforce homepage, click the gear icon () in the right corner and select **Setup**.
2. Enter **Network Access** in the **Quick Find** box, then select **Network Access**.
3. Click **New**.
4. Enter the IP address in both the Start IP Address and End IP Address fields.
5. You may enter a description (for example, "Conga Composer") in the description field.
6. Click **Save**.
7. Repeat for all Composer IP Addresses.

## To whitelist a Composer IP Address in user Profile

1. On the Salesforce homepage, click the gear icon () in the right corner and select **Setup**.
2. Enter **Profiles** in the **Quick Find** box, then select **Profiles**.
3. Click the respective profile name link.
4. Go to the **Login IP Ranges** section and click **New**.
5. Enter the IP address in both the Start IP Address and End IP Address fields.
6. You may enter a description (for example, "Conga Composer") in the description field.
7. Click **Save**.
8. Repeat for all Composer IP Addresses.

### Important

1. If you get an error accessing Conga Composer Setup after whitelisting the Composer IP Addresses, you must check the **Login History** of the respective user and add the failed IP address as shown below:
  - a. On the Salesforce homepage, click the gear icon (⚙️) in the right corner and select **Setup**.
  - b. Enter **Users** in the **Quick Find** box, then select **Users**.
  - c. Click the respective **User Name** link.
  - d. Go to the **Login History** section and check for **Failure Status** and the **Source IP**.

**Check for failure status and the Source IP**

Login Time	Source IP	Login Type	Status	Application
12/2/2021, 8:46:35 AM MST	34.210.106.249	Remote Access 2.0	Success	Conga Platform
12/2/2021, 8:46:32 AM MST	52.38.231.188	Remote Access 2.0	Success	Conga Platform
12/2/2021, 8:33:06 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:33:02 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:32:45 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:57 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:46 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:29 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:23:40 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:23:07 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning

[Download login history for last six months, including logins from outside the website, such as API logins. \(Excel .csv file\) »](#)

- e. Add the **Source IP** address into the respective user **Profiles** and **Salesforce Org**.

## Configuring Solutions Setup (Remote Sites)

You can change various user management, integration, and basic settings in Conga by visiting the Conga Setup page, which is accessible from Salesforce Setup, then Installed Packages and Configure for Conga Composer.

To build Conga Composer solutions with the Solution Manager you first need to enable remote sites.

1. Navigate to the Conga Solutions Setup tab.

⚠️ Use the All Tabs (+) option to locate Conga Solutions Setup if you do not see it.


2. Click **Add** for all four of the Remote Site URLs.
3. Click **Save** in the Remote Site Edit window.
4. Close the Remote Site Edit window.

5. Repeat for each Remote Site until all four sites have a green checkmark.

 You also need to perform these steps after you migrate your Salesforce Org to a new Pod.

## Preparing Conga Composer Settings for use in a Sandbox


You can configure your instance of Salesforce so that if you later generate a Sandbox from the production instance, Conga automatically transfers your license information to the Sandbox. This is recommended for any organization using Sandboxes. Conga views each Sandbox as a new 30-day free trial of Conga Composer.

 Configuring Conga Composer Settings in a production instance will only affect new sandboxes and does not have any influence on existing, refreshed sandboxes.

### To configure Conga Composer for use in a sandbox

1. In your Production instance, copy your Salesforce Organization ID to the clipboard. To view your ID, visit **Setup > Company Profile > Company Information**.
2. In Salesforce Setup, click **Develop > Custom Settings**.
3. Click **Manage** for Conga Composer.
4. If necessary, click **New** to create a new record, otherwise click **Edit**.
5. In the Name field, type in **Conga Settings**.
6. In the Production Org ID field, paste your Organization ID.
7. Click **Save**.

With this Custom Setting established, Conga carries the ID of your Production instance to any Sandboxes generated. The first time you click a Conga button from within the Sandbox, Conga recognizes that the Sandbox is linked to an existing production instance and automatically transfer account subscription dates and template manager settings. User permissions are not transferred. Conga user license restrictions are not enforced in Sandbox instances.

 Remember to e-mail [onboarding@conga.com](mailto:onboarding@conga.com) after installation (if you have not done so already) to ensure the fastest provisioning of any additional licenses or Composer automation products.

## Assign User Licenses

When installing Conga Composer from the AppExchange, the default license count upon installation is 10, one of which is assigned to the administrator who installed Composer. Conga Provisioning then updates your license count to match the amount that you have purchased within 1-3 business days.

 It is recommended to use Salesforce Classic to assign licenses.

## To assign the remaining licenses for Conga Composer


1. Navigate to the Installed Packages in Salesforce Setup.
2. Click **Manage Licenses** next to Conga Composer.
3. Click **Add Users**.
4. Select the box to the left of names of the users who need access to Conga Composer.
5. Click **Add** to add the selected users.
6. Click **Add Users** to save changes.

Remember to email [onboarding@conga.com](mailto:onboarding@conga.com) when you have installed (if you have not done so already) to ensure the fastest provisioning of any additional licenses and our Composer automation products.

## Permissions Needed to Use Composer

User permissions and access settings are specified in profiles and permission sets. Every user is assigned only one profile, but can also have multiple permission sets. When determining access for your users, use **profiles to assign the minimum permissions and access settings** for specific groups of users. Then use **permission sets to grant more permissions** as needed.

- [Configuring Profile Permissions](#)
- [Configuring Permissions Sets](#)

 For more information about Profiles and Permission Sets, refer to the following Salesforce Documentation:

- [Give Users Access to Data](#)
- [Share Objects and Fields](#)
- [Control Access to Objects](#)

- [Control Access to Fields](#)

## Configuring Profile Permissions

A profile in Salesforce is a group/collection of settings and permissions that define what a user can do in Salesforce. A profile controls "Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges". You can define profiles by user's job function. For example, System Administrator, Developer, Sales Representative. A profile can be assigned to many users, but a user can be assigned a single profile at a time.

This topic describes the **minimum levels of permission and access settings** required for a Conga Composer user to successfully run solutions.

You must ensure the following permissions and access settings are set correctly:

1. When installing Conga Composer from the AppExchange, select **Install for All Users**. If you did not initially install Composer for all users, you will need to modify the field level security for each field on the custom Conga Objects.
2. To edit Profile permissions, navigate to **Setup → Manage Users → Profiles**.


⚠ It is highly suggested to clone the **Standard User Profile** and use that as the Profile you customize and assign to your Composer users. According to Salesforce, a user assigned to a standard profile cannot view any custom object, as opposed to a custom profile which allows users to view custom objects.

3. For every Profile that runs Composer:
  - Minimum of **Read Access** to all custom Conga Objects (Conga Email Templates, Conga Global Merges, Conga Queries, Conga Solutions, Conga Solution Email Templates).
  - Minimum of **Read Access** to ALL Objects and Fields included in the merge templates (keep in mind the most restrictive settings apply, so if a user has visibility to a field but not the Object, they will not be able to merge data from that field). Minimum of Create permission to the Documents Object is needed if the user is going to utilize email functionality in Composer.

✔ You can also use the [Permission Sets](#) to grant additional access or temporary access to a specific set of fields and objects as per your business use cases.



- Check **API Enabled** under Administrative Permissions.
- Check **Run Reports** and **Export Reports** (if you are using reports with your Composer solutions) and also ensure the report folders that contain your Composer reports are shared with end-users.
- [Configure Composer as a Connected App](#) for any Profiles that will use the OAuth button (this is required to use Conga Composer in the Salesforce Lightning interface).
- If you save output data to chatter, update System Settings to check Moderate Chatter.


 To use Composer OAuth buttons, users must have the APXTConga4.Conga\_Composer Visualforce page added in their profile's *Enabled Visualforce Page Access* section. If this is not included, users receive an Insufficient Privileges error.

- Enable Custom Setting Definitions Access
    - **Composer**
      - Conga Composer.APXTConga4.Conga Composer Settings
      - Conga Composer.APXTConga4.Conga Solutions Settings
  - Enable Custom Metadata Type Access
    - **Composer**
      - APXTConga4.Composer Setting
      - APXTConga4.Conga Solution Export Environment
      - APXTConga4.Conga Solution Export Setting
      - APXTConga4.Connected App Setting
    - **Batch** (if Conga Batch is installed)
      - APXT\_BPM.Conductor Region
      - APXT\_BPM.Conductor Setting
    - **Trigger** (if Conga Trigger is installed)
      - CongaWorkflow.Workflow Region
      - CongaWorkflow.Workflow Setting
4. Navigate to **Setup** → **Security Controls** → **Session Settings** and ensure that *Lock sessions to the IP address from which they originated* is not enabled.

For more detailed information about setting up Profile permissions for the Standard and Enhanced View, see the following:

- [Using Standard View](#)
- [Using Enhanced View](#)

When you are ready to start building solutions, see [Composer Basics](#).

 You can use the [PermComparator tool](#) to troubleshoot differences between Salesforce profiles.

## Profile Permissions Using Enhanced View

This topic describes the Composer Profile Permissions using the Salesforce Enhanced View.

-  To enable the enhanced view:
1. From **Setup**, in the *Quick Find* box, enter *User*, and then select **User Management Settings**.
  2. Enable **Enhanced Profile User Interface**.

To enable permission to use Conga Permission in the Profile Permissions Enhanced View:

1. From **Setup**, in the *Quick Find* box, enter *User*, and then select **Profiles**.
2. Select the Profile you want to change.

### Open System Permissions

Click **Edit** and enable the following:

- API Enabled - Required for any profile using Conga Composer.
- Export Reports - Required for any profile using reports as part of their Conga Composer solution.
- Run Reports - Required for any profile using reports as part of their Conga Composer solution.
- Send Outbound Messages - Required for the user that initiates the workflow process, otherwise, the email is not sent.
- Visual Force - If you use Visual Force/Apex, click Visual Force Page Access and add access.


### Apps Section

To open Assigned Apps:

Click **Edit** and then make the following fields visible by selecting the respective checkboxes:

- Conga Composer
- Conga Batch (if Conga Batch is installed)
- Conga Mail Merge (if Conga Mail Merge is installed)
- Conga Trigger (if Conga Trigger is installed)


If the profile uses **Conga Composer OAuth buttons**, open *Assigned Connected Apps*. Click **Edit** and move Conga Composer from the Installed Connected Apps menu to the Enabled Connected Apps menu.

 Making the above fields visible is not required, but enabling them allows the profile to see the Conga apps in the *Salesforce Custom App* menu.

#### Open Object Settings

Select the following Object Names and click **Edit**. Each object must have at least Read permission (Create, Edit, and Delete may be enabled at your discretion) except for the Conga Email Staging object which requires Read, Create, Edit, and Delete permissions, and the Documents object which requires Create permission for email solutions (Delete permission is also highly recommended for Documents).

- Conga Batches (if Conga Batch is installed) - Conga Batch is an optional, paid add-on to Conga Composer that launches on-demand or scheduled Conga Composer solutions in batches.
- Conga Email Staging - Conga Email Staging is a component of Conga Email; Read, Create, Edit, and Delete permissions are required for any profile using Conga Email.
- Conga Email Templates - Conga Email Templates is a custom object that allows you to build and store HTML email templates for use in your merge operation, whether as the primary destination for merged data or used in conjunction with a document template.
- Conga Queries - Conga Queries is a custom object that allows you to build and store SOQL queries to retrieve data for use in your merge operation.
- Conga Templates - Conga Templates is a custom object that allows you to store your document (Word, Excel, PowerPoint, or PDF) templates.
- Documents - The Create permission is required on the Documents object in order for a user to utilize email functionality with Conga Composer. Delete permission is also recommended for the Documents object.

 The Documents object is used in Conga Composer as a temporary holding area for the merged output file while the email is assembled. The Create permission allows a user to utilize this functionality while the Delete permission ensures that each document that is created by a user is replaced the next time that the user initiates a Conga Composer email process.

- QuickMerge Links (if Conga QuickMerge is installed) - Conga QuickMerge is an optional, free add-on to Conga Composer that helps automate Composer and Conga Mail Merge solutions by expanding the scope of how they can be launched.

#### Open Custom Metadata Types

Add the following Custom Metadata Types to the Enabled Custom Metadata Types section for each Conga product that the profile (that is assigned this specific profile) will use. After adding the Custom Metadata Types, click **Save**.

- **Composer**
  - APXTConga4.Composer Setting
  - APXTConga4.Conga Solution Export Environment
  - APXTConga4.Conga Solution Export Setting
  - APXTConga4.Connected App Setting
- **Batch** (if Conga Batch is installed)
  - APXT\_BPM.Conductor Region
  - APXT\_BPM.Conductor Setting
- **Trigger** (if Conga Trigger is installed)
  - CongaWorkflow.Workflow Region
  - CongaWorkflow.Workflow Region

Open Custom Setting Definitions

Add the following Custom Setting Definitions. After adding the Custom Setting Definitions, click **Save**.

- **Composer**
  - Conga Composer.APXTConga4.Conga Composer Settings
  - Conga Composer.APXTConga4.Conga Solutions Settings

## Profile Permissions Using Standard View

This topic describes the Composer Profile Permissions configuration using the Salesforce Standard View.

Navigate to **Salesforce Setup** → **Administration** → **Users** → **Profiles** > Select the Profile Name you want to change (e.g. Business Analyst) → **Edit**.

You must review and change the following permissions and access settings:

Custom App Settings

Select the following checkboxes to make them visible. This is not required but enabling it will allow the profile to see the Conga® apps in the Salesforce Custom App menu.

- Conga Composer
- Conga Batch® (if Conga Batch is installed)
- Conga Mail Merge® (if Conga Mail Merge is installed)
- Conga Trigger® (if Conga Trigger is installed)

### Connected App Access

If the profile is using the Conga Composer OAuth buttons, enable Conga Composer as a Connected App by selecting the check box.

### Tab Settings

Custom Tab Settings - "Default On" is the recommended setting, however, is not require permission for end-users; enabling it will simply allow a user to see the Conga tabs. We do require this permission for System Administrators, as it is needed for maintenance and troubleshooting of solutions.

- About Conga Composer
- Conga Composer Setup
- Conga Templates
- Conga Email Templates
- Conga Queries
- Conga Resources
- Conga Mail Merge (if Conga Mail Merge is installed)
- QuickMerge Links (if Conga QuickMerge is installed)

If Conga Batch is installed, we recommend enabling its tabs as "Default On." These tabs are required as "Default On" for System Administrators.

- About Conga Batch
- Conga Batches
- Conga Batch Setup
- Conga Batch Dashboard

If Conga Trigger is installed, we recommend enabling its tabs as "Default On." These tabs are required as "Default On" for System Administrators.

- About Conga Trigger
- Conga Trigger Setup
- Conga Trigger Formula Builder
- Conga Trigger Dashboard

### Administrative Permissions

Select the API Enabled check box. This is required permission for anyone using Conga Composer.

- API Enabled

### General User Permissions

Select the following checkboxes; both are required for any users who are using reports as part of their Conga Composer solutions.

- Export Reports
- Run Reports

#### Standard Object Permissions

Create permission is required on the Documents object in order for a user to utilize email functionality with Conga Composer. Delete permission is also highly recommended for the Documents object.


The Documents object is used in Conga Composer as a temporary holding area for the merged output file while the email is assembled. The Create permission allows a user to utilize this functionality while the Delete permission ensures that each document that is created by a user has subsequently replaced the next time that user initiates a Conga Composer email process.

#### Custom Object Permissions

Provide at least Read access (Create, Edit, Delete, View All, and Modify All are not required for end-users and can be enabled at your discretion) for the following objects. Read, Create, Edit, and Delete permissions are required for Conga Email Staging.

- Conga Templates<sup>SM</sup> - Conga Templates is a custom object that allows you to store your document (Word®, Excel®, PowerPoint®, or PDF) templates.
- Conga Email Templates<sup>SM</sup> - Conga Email Templates is a custom object that allows you to build and store HTML email templates for use in your merge operation, whether as the primary destination for merged data or used in conjunction with a document template.
- Conga Email Staging<sup>SM</sup> - requires Read, Create, Edit, and Delete permissions.  
Note: Conga Email Staging is a component of Conga Email; Read, Create, Edit, and Delete permissions are required for any profile using Conga Email
- Conga Queries<sup>SM</sup> - Conga Queries is a custom object that allows you to build and store SOQL queries to retrieve data for use in your merge operation.
- Conga Batch (if Conga Batch is installed) - Conga Batch is an optional, paid add-on to Conga Composer that launches on-demand or scheduled Conga Composer solutions in batches.
- Conga Trigger (if Conga Trigger is installed) - Conga Trigger is an optional, paid add-on to Conga Composer that launches Conga Composer operations as the result of a Salesforce Conga Trigger rule.

- QuickMerge Links<sup>SM</sup> (if Conga QuickMerge is installed) - Conga Quick Merge Links is an optional, free add-on to Conga Composer that helps automate Composer and Conga Mail Merge solutions by expanding the scope of how they can be launched.
- Scheduled Conga Batch History<sup>SM</sup> (if Conga Batch is installed) - Scheduled Conga Batch History is a child object of Conga Batch and displays details of past scheduled Conga Batch routines.]

 In addition to providing access to Objects, ensure Profiles have access to all necessary fields on the custom Conga Objects as well as fields on other Objects from which you are merging data.

#### Enabled Custom Setting Definitions

Click **Edit** under the Enabled Custom Setting Definitions Access section and add the following Available Custom Setting Definitions to Enabled Custom Setting Definitions.

- **Composer**
  - Conga Composer.APXTConga4.Conga Composer Settings
  - Conga Composer.APXTConga4.Conga Solutions Settings

#### Enabled Custom Metadata Type Access

Click **Edit** under the Enabled Custom Metadata Type Access section and add the following Available Custom Metadata Types to the Enabled Custom Metadata Types. After adding the Custom Metadata Types, click **Save**.

- **Composer**
  - APXTConga4.Composer Setting
  - APXTConga4.Conga Solution Export Environment
  - APXTConga4.Conga Solution Export Setting
  - APXTConga4.Connected App Setting
- **Batch** (if Conga Batch is installed)
  - APXT\_BPM.Conductor Region
  - APXT\_BPM.Conductor Setting
- **Trigger** (if Conga Trigger is installed)
  - CongaWorkflow.Workflow Region
  - CongaWorkflow.Workflow Setting

## Configuring Permissions Sets

The permission set is also very similar to the profile. Whatever you can manage at profiles (Like Object permissions, Field Permissions, User permissions, Tab settings, App settings,

Apex class permission, visualforce permission) the same you can manage using permission sets also. But the main difference between these two is that users can have only one profile and can have multiple permission sets at a time.

In order to grant a very specific ability to a user, you obviously don't want to create a whole new profile just for that one difference between their abilities and the rest of their team's! So, you can define [profiles to grant minimum permissions and settings](#) that every type of user needs, then you can use the **permission set to grant additional access**.

#### For Example:

1. You have many users in your organization with some fundamental job functions. You can assign all of them with one profile that grants them all access to do their job. But some sets of people are working on special apps or some special functionality, for this type of special user you can create permission sets and can be assigned to them.

**Let's take an example:** There is a sales team, who have the profile 'Sales User'.

Only James should be able to change the team's email templates, so you as an Admin create a Permission Set called 'Modify Email Templates' which you add to James's user record.

2. Some users need some temporary access to a specific set of fields and objects you can create a permission set with those objects & field access and you can assign those specific users.

For more information, see [Permission Sets](#) and [Permission Set Groups](#).

## Composer Out-of-Box Permission Sets

Conga Composer is installed with the following Permission Sets that you can use for user access and permissions.


- **Composer\_Administrator:** Setup permissions for Conga Composer Administrator.
- **Composer\_User:** Setup permissions for Conga Composer User.
- **Composer\_Salesforce\_Community\_User:** Setup permissions for Conga Composer Communities User.

**If you want to use the out-of-box permission sets instead of profile level settings**, the following are the **minimum levels of permission and access settings** required for a Conga Composer user to successfully run solutions. You must ensure the following permissions and access settings are set correctly:


1. Assign the **Composer\_User** permission set to the Composer user who runs the solutions.



2. Create and assign a new Permission Set or Profile to the Composer user, and ensure the following permissions and access settings are set correctly:
  - a. Check **API Enabled** under System Permissions.
  - b. Check **Run Reports** and **Export Reports** under System Permissions (if you are using reports with your Composer solutions) and also ensure the report folders that contain your Composer reports are shared with users.
  - c. Check **Send Outbound Messages** under System Permissions. (Required for the user that initiates the workflow process, otherwise, the email is not sent)
  - d. Check **APXTConga4.Conga Composer(APXTConga4\_CongaMerge)** under the Assigned Apps. Click **Edit** and move *APXTConga4.Conga Composer(APXTConga4\_CongaMerge)* from the Available Apps to the Enabled Apps.
  - e. [Configure Composer as a Connected App](#) for any Permission Set that will use the OAuth button (this is required to use Conga Composer in the Salesforce Lightning interface).


 To use Composer OAuth buttons, users must have the APXTConga4.Conga\_Compiler Visualforce page added in their profile's *Enabled Visualforce Page Access* section. If this is not included, users receive an Insufficient Privileges error.

3. Minimum of **Read Access** to ALL Objects and Fields included in the merge templates (keep in mind the most restrictive settings apply, so if a user has visibility to a field but not the Object, they will not be able to merge data from that field). Minimum of Create permission to the Documents Object is needed if the user is going to utilize email functionality in Composer.

 You can use Permission Set to grant additional access or temporary access to a specific set of fields and objects as per your business use cases.

4. Navigate to **Setup** → **Security Controls** → **Session Settings** and ensure that *Lock sessions to the IP address from which they originated* is not enabled.

#### Tab Setting under Profile Permissions:

 Navigate to **Salesforce Setup** → **Administration** → **Users** → **Profiles** > Select the Profile Name you want to change (e.g. Business Analyst) → **Edit**.

Custom Tab Settings - "Default On" is the recommended setting, however, is not require permission for end-users; enabling it will simply allow a user to see the Conga tabs. We do require this permission for System Administrators, as it is needed for maintenance and troubleshooting of solutions.

- About Conga Composer
- Conga Composer Setup
- Conga Templates
- Conga Email Templates
- Conga Queries
- Conga Resources
- Conga Mail Merge (if Conga Mail Merge is installed)
- QuickMerge Links (if Conga QuickMerge is installed)

If Conga Batch is installed, we recommend enabling its tabs as "Default On." These tabs are required as "Default On" for System Administrators.

- About Conga Batch
- Conga Batches
- Conga Batch Setup
- Conga Batch Dashboard


If Conga Trigger is installed, we recommend enabling its tabs as "Default On." These tabs are required as "Default On" for System Administrators.

- About Conga Trigger
- Conga Trigger Setup
- Conga Trigger Formula Builder
- Conga Trigger Dashboard

When you are ready to start building solutions, see [Composer Basics](#).

## Composer Authorization and Permissions Matrix

You must refer to the following Authorization and Permission Matrix to run your Conga Composer Solution successfully.

 The method of authorization and the permissions needed to execute a merge vary on the basis of the execution point and the method.

	Type of Authentication			Connected Apps			Pre- Requisit es and Permissi ons
	Session ID	OAu th	JWT (JSON Web Token)	Comp oser NA,E MEA, AP	Compos er SF1, NA, EMEA, AP	Sche duler	
Classic SF Button	✓	✓		✓			3, 4
SF Custom Action		✓			✓		3, 4
Lightning Component		✓		✓			1, 2, 3, 4
Orchestrate			✓	✓			3, 4
ServiceUser			✓	✓			3, 4
Trigger	From Outbound Message			✓			3, 4
Batch	From Refresh Token			✓		✓	3, 4, 6
Community Lightning Component		✓		✓			1, 2, 3, 4
Mobile Custom Action		✓			✓		3, 4
Mobile Lightning Component		✓			✓		1, 2, 3, 4, 7
React Setup Menu			✓	✓			5

## Pre-Requisites and Permissions Legend


1. You must have a My Domain org to use the Lightning Component.

2. To use the Lightning Component, the users in the org must have read permission to custom metadata types for Composer (and CQG) to read the refresh token stored there (this is in the Composer permission set).
3. All Composer packages older than version 8.179 "Grays" must provide explicit Read permissions for all Composer users to custom settings and metadata.
4. If an Admin installs the Conga Composer and selects "For Admins Only" during the installation, then they will need to create a permission set to give their users access to the connected apps and all Composer package components.
5. Composer\_Administrator permission set.
6. Conga Batch Region Metadata permission is required to run Batch from Grid view.
7. To use the Lightning Component for SF1, you must have "Modify Metadata Through Metadata API Functions" AND "Customize Application".

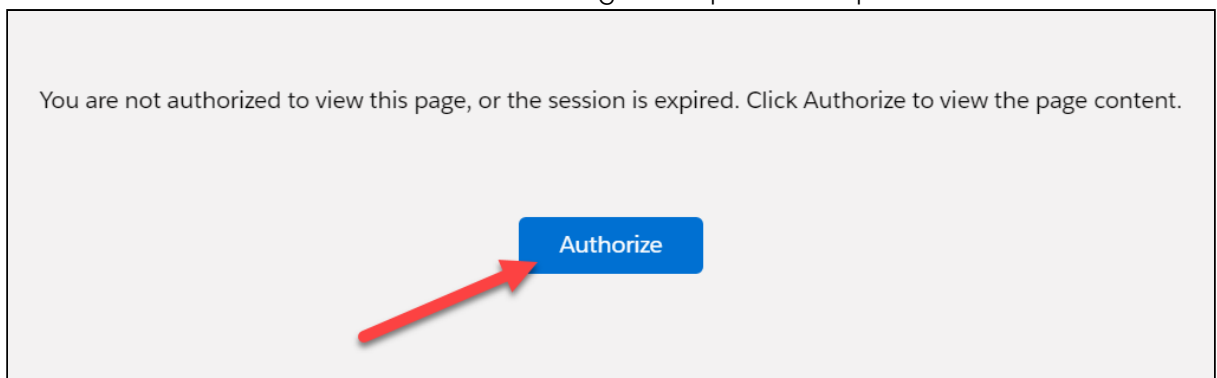
## Basic Settings and Configuring Composer

Basic settings for local template usage, template manager repositories, and solution behaviors.

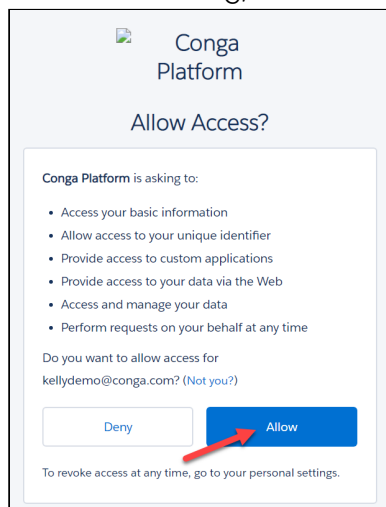
### Required Setup

 You must have Conga Composer version 8.183 or higher to use the new [Conga Composer Setup Menu](#). Any Salesforce orgs with Composer versions preceding version 8.183 must upgrade to use the new Conga Composer Setup Menu. To complete the setup process in versions preceding version 8.138, click the Conga Composer Setup tab to access the setup menu and then click Save Changes.

1. Once the installation is complete, navigate to the Conga Composer Setup tab.
2. Click **Authorize** to authorize access to Conga Composer Setup.



3. After authorizing, click **Allow** to access to Conga Composer Setup.



### If you are installing for a PAID Composer Subscription Only:

Email [onboarding@conga.com](mailto:onboarding@conga.com) with your Org ID and Account Name to notify Conga that you have completed the installation. To find your Org ID, see: [Where can I find my Salesforce Org ID?](#)

- Your initial installation is automatically provisioned as a 30-day trial with access to 10 licenses. Emailing [onboarding@conga.com](mailto:onboarding@conga.com) ensures your licenses are enabled as quickly as possible.
- Conga Onboarding provisions your licenses within three business days. You will then [Assign User Licenses](#) (Production orgs only). You have access to 10 licenses, and 1 will already be assigned to you as the Admin.

## Optional: Additional Conga Composer Setup

Navigate to the Conga Composer App in Salesforce, and then select the Conga Composer Setup Tab. You can change various user management, integration, and basic settings in Conga® by visiting the Conga Setup Tab.

Under the Composer Configuration section, you can change the default settings for local template usage, template manager repositories, and solution behaviors. You must ensure that the following items are selected in the listed sections:

- Under the Conga Template Manager: Ensure that the new, built-in Conga Template Manager is checked (this should be enabled by default).
- Legacy Configuration: Ensure all items are selected under the Legacy Configuration section. (This should be enabled by default.)

⚠ Conga Composer's user interface dynamically translates to German and French for Salesforce users that have their local language in Salesforce set to German or French. For more information, [Supported Composer User Interface Languages](#).

## Enable Email Notifications

⚠ Contact Conga Support to enable this feature.

[Notification Emails](#) is a [Composer Advanced Feature](#) that notifies users when recipients engage with documents sent with [Composer Link Service](#).

## Conga File Storage

⚠ Contact Conga Support to enable this feature.

Conga File Storage is a [Composer Advanced Feature](#) that allows users to store documents generated with [Composer Link Service](#) in the [Conga File Repository](#).

## Local Template

By default, and as a recommended practice, only users with the Salesforce System Administrator profile are permitted to upload a template from their local hard drives. This feature is always enabled for administrators because of its utility in creating and testing templates. You may enable this feature for all other user profiles by checking the box to permit users to upload a template from a local drive.


Local templates can also be enabled on a solution-by-solution basis via the Conga Composer parameter `&DS8=1`.

## Conga Template Manager

Instead of using the Salesforce Communication Templates repository, we strongly recommend you store templates in the Conga Template Manager. This custom object is included in the Conga Composer application so does not require separate installation.

## About the Legacy Configuration Section

The Legacy Configuration settings govern how Conga retrieves data and how templates behave. If your organization is new to Conga, we strongly recommend that you enable all of the Behavior Modification settings.

 If your organization is upgrading to Release 8 from Conga Composer 1.6 or earlier (or any version where you previously did not have these settings enabled), you may now enable these settings, however, they might affect your templates. Proceed with caution.

## Query Object Name Correction

This setting corrects a problem in which Conga Queries resulted in object names with double underscores. For example, a merge field name for the Amount field on an Opportunity sometimes appeared as:

`OPPORTUNITY__AMOUNT`

Rather than:

`OPPORTUNITY_AMOUNT`


This was an issue because modifying the Conga Query sometimes resulted in slightly different field names, thus preventing a merge field from working properly.

## Prevent Auto-Incrementing Column Name

This setting corrects a merge field naming problem in Reports and Queries in which column names were appended with “\_1” to delineate multiple instances of the merge field across the entire dataset.

## Word Templates: Require TableStart/TableEnd for Detail Regions

You must use TableStart/TableEnd to access detail data, which is important because some customers have unintentionally discovered that Conga Composer would generate a separate document for each row of detail data in the absence of a TableStart/TableEnd.

 In Conga Composer Release 8, all templates require TableStart and TableEnd in order to merge detail data, regardless of whether this checkbox is checked.

## Simple Report Column Names

This setting prevents Conga from altering field names used in Reports and Queries when a Salesforce ID column is also referenced in the Report or Query.

## Enable Text-based Merge Fields (Word)

This setting enables text-based merge fields in Microsoft Word templates. If this setting is not enabled only Microsoft Word traditional merge fields are available.

### Example:

- Text-based merge field: {{ACCOUNT\_NAME}}
- Traditional merge field: «ACCOUNT\_NAME»

## Conga Composer Setup Menu

The new and improved Conga Composer Setup Menu features an enhanced user interface that allows users to easily configure Conga Composer in minutes.

**⚠** You must have Conga Composer version 8.183 or higher to use the new Conga Composer Setup Menu. Any Salesforce orgs with Composer versions preceding version 8.183 must upgrade to use the new Conga Composer Setup Menu.

The screenshot displays the 'Composer Configuration' settings page. The left sidebar shows a navigation menu with categories like 'Integration Partners', 'Enterprise', and 'CONGA APPS'. The main content area is titled 'COMPOSER SETTINGS' and 'Composer Configuration'. It includes several configuration sections:

- Salesforce Org ID:** A text field containing '00D140000047MJEAU'.
- Enable Email Notifications:** A toggle switch labeled 'Email' that is currently turned on.
- Conga File Storage:** A toggle switch labeled 'File Storage' that is currently turned on.
- Local Templates:** A checkbox labeled 'Permit users to upload a template from a local directory. (Recommended)' which is currently unchecked.
- Conga Template Manager:** A section with explanatory text and a list of benefits. It includes three checkboxes:
  - Enable the new, built-in Conga Template Manager (Recommended)
  - Enable the Conga Template Manager (V4)
  - Enable Salesforce Communication Templates
- Legacy Configuration:** A section with a right-pointing arrow and the text 'Not recommended for most orgs.'

A 'Save' button is located in the top right corner of the configuration area.



## Composer Configuration

The Salesforce Org ID is conveniently displayed at the top of the page to assist with Conga Support interactions.

Conga Composer Setup also features the ability to easily enable [Composer Advanced Features: Notification Emails](#) and the [Conga File Repository](#).

Administrators can check the Local Templates checkbox field to allow users to upload and merge templates from their machines, in addition to using the available templates in the Composer user interface. Administrators also have the option to customize the Conga Template Manager to provide additional or past functionality.

## Integration Partners

The Integration Partners sections allow administrators to connect and integrate third-party applications with Composer. The following integrations are available in Conga Composer Setup:

- Email
  - SendGrid
- eSignature
  - Conga Sign
  - Adobe Sign
  - Sertifi
  - OneSpan Sign
  - DocuSign
- Document Storage
  - Google Drive
  - Amazon S3
  - Box
  - OneDrive


## Solution Migration

Use the Solution Migration Utility tool to seamlessly migrate solutions between orgs. This tool eliminates the hassle of re-building solutions when migrating from sandbox environments to production environments.

## Usage Dashboard

The Usage Dashboard allows administrators to create Composer usage reports on the fly. These reports provide critical insight into each Composer user's activity and application usage.

## Enterprise

 The features included in the Enterprise section require Composer Enterprise edition. External Data Sources configuration is not supported by the Conga Support team.

Administrators can also set up [External Data Sources](#). External Data allows customers to import data from sources outside of Salesforce into their Conga Composer solutions. Data from multiple instances of Salesforce can also be combined.

## Supported Languages for Composer Setup Menu

**Conga Composer Setup Menu** currently supports dynamic translation for German, French, Dutch, Polish, and Spanish, in addition to English as a default language. It automatically translates to German, French, Dutch, Polish, or Spanish for the Salesforce Users based on their local language set in Salesforce.

### Note

You must have Conga Composer version 8.183 or higher installed in your Salesforce org to leverage German, French, Dutch, Polish, and Spanish dynamic translations in the new *Conga Composer Setup Menu*.

To set dynamic translation

1. Under your Salesforce Profile, select **Settings**.

2. Click **Languages & Time Zone**.

The screenshot shows the 'Language & Time Zone' settings page. The 'Language' field is highlighted with a red box and set to 'Deutsch'. Other settings include 'Time Zone' (GMT-08:00 Pacific Standard Time (America/Los\_Angeles)), 'Locale' (English (United States)), and 'Email Encoding' (General US & Western Europe (ISO-8859-1, ISO-LATIN-1)).

3. Change the Language field to Deutsch (German), Français (French), Nederlands (Dutch), Polski (Polish), or Español (Spanish).
4. Click **Save**.

## Enable Conga Composer Security Settings

You must enable security settings to allow Conga Composer to interact with your data:

- For each user profile that has access to Conga Composer, ensure the API Enabled setting is active.
- If you want to use Salesforce® Reports as a data source for Composer solutions, ensure that Run Reports and Export Reports are enabled for each user profile that has access to Conga Composer.

**⚠** Do not set Lock IP Sessions in Salesforce. Doing this will disable Conga Composer from being able to communicate with Salesforce, rendering Conga Composer unusable.

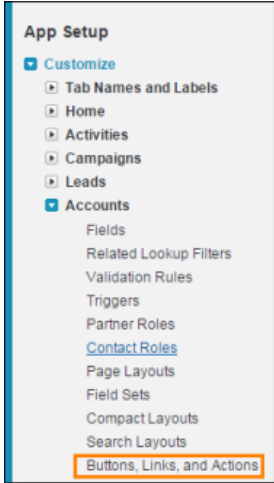
## Create a Conga Composer OAuth Button

If you want to use OAuth with Conga Composer, you must create special Conga Composer OAuth buttons. These buttons are included in the managed package. You can use the managed buttons or follow the procedure below to update the URL of an existing button.

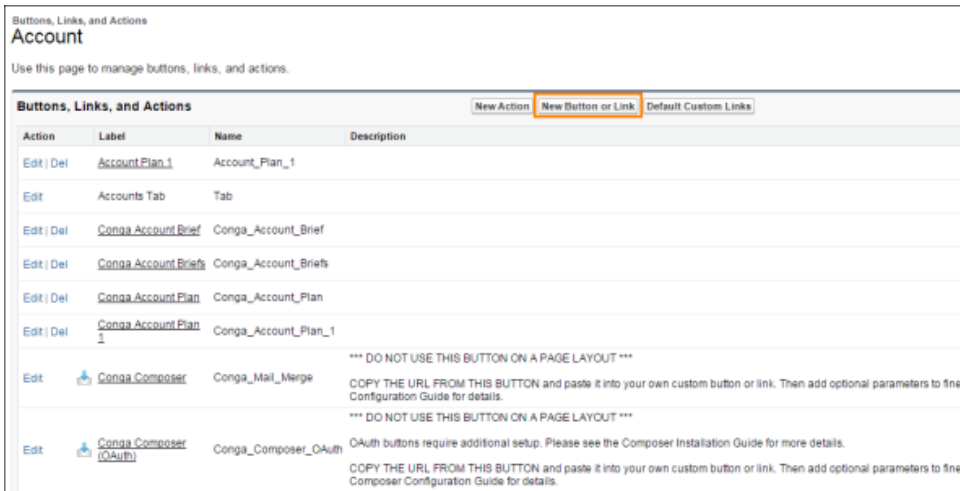
**⚠** You must use Composer OAuth buttons in Salesforce Lightning.

## To create a Conga Composer OAuth button

1. Navigate to Leads, Accounts, Contacts, Opportunities, Cases, Contracts, or Solutions > Buttons, Links and Actions > Conga Composer (OAuth) to locate a managed Conga Composer OAuth button.



2. Copy the contents of the Button or Link URL field to your clipboard.
3. Navigate to your desired standard or custom object.
4. Click **New Button or Link**.



- Complete the Label field to give the button a name.

Account Custom Button or Link  
New Button or Link

Custom Button or Link Edit Save Quick Save Preview Cancel

Label

Name

Description

Display Type  
 Detail Page Link [View example](#)  
 Detail Page Button [View example](#)  
 List Button [View example](#)

Behavior  [View Behavior Options](#)

Content Source

Select Field Type  Insert Field  Insert Operator

Functions  
 -- All Function Categories --  
 ABS  
 AND  
 BEGINS  
 BLANKVALUE  
 CASE  
 CASESAFEID

- Press **Tab** or click in the Behavior field to automatically populate the Name field.
- In the Behavior field, select Display in new window.

Account Custom Button or Link  
New Button or Link

Custom Button or Link Edit Save Quick Save Preview Cancel

Label

Name

Description

Display Type  
 Detail Page Link [View example](#)  
 Detail Page Button [View example](#)  
 List Button [View example](#)

Behavior  [View Behavior Options](#)

Content Source

Select Field Type  Insert Field  Insert Operator

Functions  
 -- All Function Categories --  
 ABS  
 AND  
 BEGINS  
 BLANKVALUE  
 CASE  
 CASESAFEID

- In the URL field, paste the Button or Link URL you copied in step 2.

Account Custom Button or Link  
New Button or Link

Custom Button or Link Edit Save Quick Save Preview Cancel

Label

Name

Description

Display Type  
 Detail Page Link [View example](#)  
 Detail Page Button [View example](#)  
 List Button [View example](#)

Behavior  [View Behavior Options](#)

Content Source

Select Field Type  Insert Field  Insert Operator

Functions  
 -- All Function Categories --  
 ABS  
 AND  
 BEGINS  
 BLANKVALUE  
 CASE  
 CASESAFEID

`/apex/APXIConges_Conges_Composer?id={!Account.Id}`

9. Click **Save**.

10. Click **Window Open Properties**, set the Height to 660 and Width to 900, and then click **Save**.
11. Add your button to the object's page layout. The button is available from the button drop-down list on the record.
12. (Optional): To make the button visible on the record itself, add the button to the Salesforce1 and Lightning Experience Actions section on the page layout. Drag and drop the button to the Salesforce1 and Lightning Actions section.

After creating the Conga Composer OAuth button, administrators must configure Conga Composer as a Connected App and grant access to specific users via Profiles or Permission sets. These steps are covered in [Configure the Conga Composer Connected App](#).

## OAuth for Conga Composer

“OAuth (an abbreviation of open authorization) is an open protocol which allows secure authorization of desktop and web applications to access APIs. A commonly used analogy is the valet key to a car, which allows the car to be driven (perhaps a limited distance), but does not give access to the glove box or trunk. In the same way, OAuth allows users to authorize applications to access resources on their behalf via an access token, rather than by handing over their actual username and password.” ([Patterson, 2011](#))

Conga has enabled Salesforce OAuth in Conga Composer Release 8. OAuth, an industry-wide accepted protocol for secure API authorization, provides several key benefits:

- The identification process occurs securely between Salesforce and Conga servers, reducing client-side security risks.
- The process makes it simple for a user (or administrators) to revoke the communication between Conga and Salesforce if desired.

If you have not authorized Conga before, when you click a Conga Composer button, a Salesforce dialog box appears asking whether to allow Conga Composer to access to various Salesforce functions. If you authorize Composer, Salesforce provides a temporary code (which Composer uses in conjunction with a protected key) to allow Composer to interact with Salesforce on your behalf.

You can revoke Conga Composer's access at any time using the revoke link found in My Settings > Personal > Connections. There may be multiple entries for Conga. If access is revoked, the Salesforce Authorization window appears the next time you use a Conga product.

To get started, see [Configure the Conga Composer Connected App](#).

You are required to use OAuth in Salesforce Lightning. Traditional Salesforce URLs that utilize a Salesforce Session ID will operate as usual in Salesforce Classic.

Salesforce reports for Conga Composer solutions using OAuth is supported only in Conga Composer version 8.2 or later. If you are running version 8.0 or 8.1, we recommend the free upgrade to version 8.2. Alternately, you can use SOQL queries instead of Salesforce reports.

## Composer for Salesforce Mobile Configuration

How to configure Composer Salesforce Mobile.

### Requirements

- Salesforce® Enterprise Edition or above
- Working Composer solution in [Background Mode \(DS7 parameter\)](#) using a supported setting
- Access to the Conga Composer® SF1 Connected App:
- Salesforce1® for your supported device
- Conga Composer 8.4.1 or greater
- Only available for certain supported objects. To see a list of objects, see [Supported Functionality in Composer for Salesforce1](#).

### Steps to set up Composer for Salesforce1

1. [Connected Apps Configuration](#)
2. [Add Global Publisher Action to desired page layout](#)
3. [Add &SF1=1 parameter to button](#)

## Integration Partners Settings

Conga Composer integrates with a number of different providers. The Integration Partners tab introduces the partners and how their products are configured to work with Conga Composer®. These include:

- Adobe eSign
- DocuSign
- Google Drive
- SendGrid
- Sertifi
- SpringCM
- Box

## Composer Activity Report Overview

You can run activity reports within the **Conga Composer Setup** menu. These reports demonstrate the total usage and Conga products that are utilized on a summary level and a granular level.

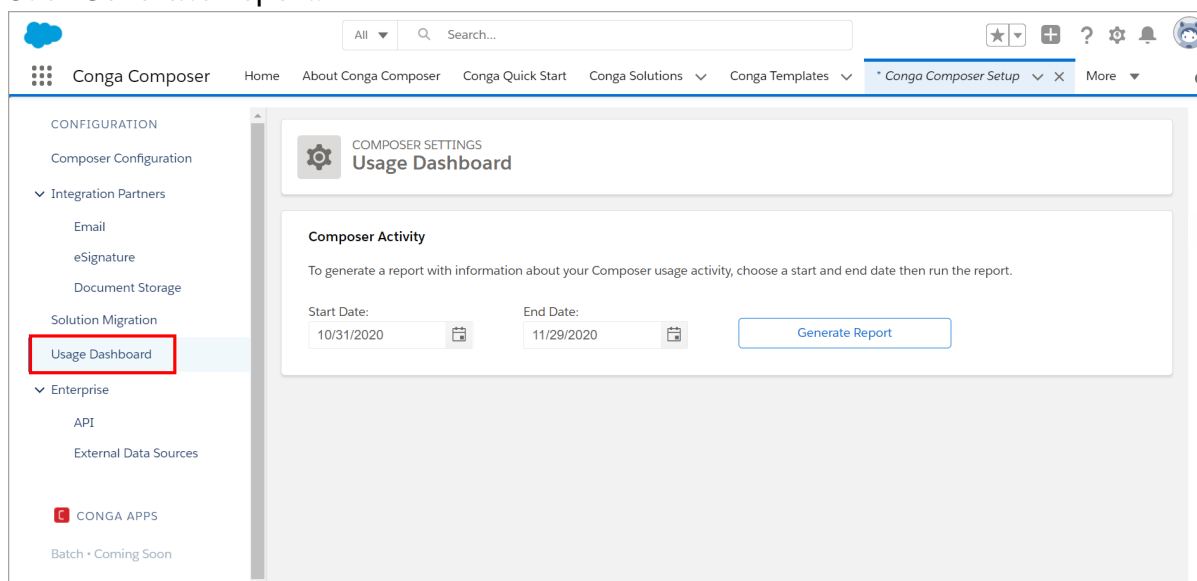
### To generate Composer Activity Report

1. Go to the **Conga Composer Setup** menu.
2. Click the **Usage Dashboard** tab.
3. Choose a start and end date.

 To run today's activity report, select "Today" as the end date.



#### 4. Click **Generate Report**.



Conga will then run a comprehensive activity report showing all of your merges within that time frame. You can then analyze the data on a summary level by the user, or look at the raw data containing each individual merge process and the product used.

## Do I Need a Conga Composer User License During My Conga Trial?

When you install the Conga Composer free 30-day trial in a production instance you are granted 10 Conga Composer user licenses for use during the trial period. One of the ten Conga Composer user licenses must be assigned to the System Administrator.

Contact your Account Executive if you require your trial to be extended. [Click here](#) to learn how to assign the Conga Composer user licenses.

## Conga Composer Add-on Trials

- **Conga Batch:** You must install Conga Batch. Anyone wishing to run Conga Batch must have an assigned Conga Composer license.
- **Conga Trigger:** You must install Conga Trigger. Conga Trigger runs under the System Administrator license.
- **Conga Mail Merge and Conga QuickMerge:** You must install one or both of these products. Anyone wishing to run either of these products must have an assigned Conga Composer license.

# How Composer Sends Emails from Salesforce

## Sending Emails with a Composer Button

The use cases below describe how emails are sent from Salesforce when using a Composer button.

1. **Using a Salesforce Classic Email Template** - Composer uses the standard Salesforce Email service to send the email, the same service as if you were to click Send Email from the activity history related list in Salesforce Classic.

 Composer does not use Salesforce Lightning Email Templates

2. **Using a Conga Email Template** - Composer uses the Salesforce Send Email API (`sendEmail(SingleEmailMessage emails[])`). This is a programmatic email send, and it counts against your daily Salesforce limit of 5,000 emails per day. For more information on the Salesforce Send Email API, see [sendEmail\(\)](#).
3. **Using Background Mode (DS7=2)** - Composer uses either of the above two options, depending on which email template type you are using (Salesforce Classic Email Template or Conga Email Template).
4. **Using Background Mode (DS7=12)** - Composer uses the Salesforce Salesforce Send Email API (`sendEmail(SingleEmailMessage emails[])`). This is a programmatic email send, and it counts against your daily Salesforce limit of 5,000 emails per day. For more information on the Salesforce Send Email API, see [sendEmail\(\)](#).

## Sending Emails with Conga Trigger and Batch

Conga uses the Salesforce Mass Email API (`sendEmail(MassEmailMessage emails[])`) when sending emails via Conga Trigger and Conga Batch.

For more information on the Salesforce Mass Email API, see [sendEmail\(\)](#).

## Sending Emails with SendGrid Enabled

If you have SendGrid enabled for your Salesforce org, all Conga emails will be routed through SendGrid services and bypass the Salesforce email functionality. Emails will not be counted against your daily Salesforce limit. For more information, see [SendGrid Integration Parameters](#) for more information.

## How to migrate Composer servers from NA to EMEA or APAC

You can migrate your Composer instance from North America (NA) to Europe, Middle East, and Africa (EMEA) or Asia-Pacific (APAC) databases.

**Requirements:** Composer 8 and the latest package.

⚠ Please test any migration in your Sandbox first.

⚠ If you do not migrate to the appropriate database, Composer may not load as quickly.

Composer is supported on EMEA and APAC databases. If your Salesforce instance is in the Europe, Middle East, and Africa (EMEA) or Asia-Pacific (APAC) locales and you would like to migrate your Composer database location, you must contact Conga support.

For more information, contact Conga support at <https://conga.com/support>.


⚠ Migration support is for region data residency only. Enhanced security is the main goal of this feature. Performance improvements are expected as well.

## Migrating Conga Composer from a Salesforce Sandbox Instance to a Salesforce Production Instance

To migrate Conga Composer from a Salesforce Sandbox Instance to a Salesforce Production Instance:

1. Copy your existing Conga Button URLs from your Sandbox instance to add to your Production instance. You will need to manually create a new button in the Production instance and can then paste into it the appropriate button URL from the Sandbox environment.
2. Recreate all your Reports (including any necessary custom report types), Queries, and custom fields in the Production instance.
3. Recreate all the Conga Template records in your Production instance. You will also need to upload the appropriate Word, Excel, PowerPoint, or PDF template file to the coordinating Conga Template record in the Production instance. If you do not have the template files saved on your local computer, intranet, etc., you may download them from the Conga Template record in the Sandbox instance.

4. Save the source code (HTML) from the Conga Email Template records in your Sandbox instance. We recommend saving each Conga Email Template's HTML source code into a .txt file. Create a new Conga Email Template record in the Production instance and paste the appropriate source code from the .txt file from the Sandbox Conga Email Template record into the Production Conga Email Template record.


 The Source button (used to access the HTML source code for a Conga Email Template) is only accessible from the Advanced Editor on the Conga Email Template record.

In the Conga Composer button URL:

- Update all IDs for example, reports, queries, templates, to the Production versions.
- Update all references to custom fields. Ensure the field has the correct merge field or API name.

Some browsers have a tendency to add spaces when copying or pasting from one org to another. Remove any additional spaces from the Button URL or errors occur when starting Composer.

## Required Profile Permissions for Salesforce Winter '20 Critical Update

 Customers using Composer versions before 8.149 **DO NOT** need to make this update. To check your version of Composer, see [Critical Update Actions for Composer Versions](#). Composer releases go all the way up to the hundreds decimal. For example, version 8.177 is a later and higher version than 8.9.

## Salesforce Winter '20 Critical Update Impact on Conga Products

The Salesforce Winter '20 Critical Update "Require Customize Application Permission for Direct Read Access to Custom Metadata Types" impacts the ability of Composer and Conga Quote Generation to read the Composer Settings custom metadata type.

The Composer Settings custom metadata determines general connection information when launching Composer and Conga Quote Generation from the /apex/APXTConga4\_\_Conga\_Composer Visualforce page. User access to this custom metadata is critical for product usage.


Users with insufficient access to the Composer Settings custom metadata type receive the following error message when attempting to access the /apex/APXTConga4\_\_Conga\_Composer Visualforce page: *An internal server error has occurred.*

For more information about this critical update, see [Require Customize Application Permission for Direct Read Access to Custom Metadata Types \(Critical Update\)](#).


Follow the steps below to avoid this error and ensure that users have sufficient access to the /apex/APXTConga4\_\_Conga\_Composer Visualforce page.

Important: The custom metadata types provide users with read access to only Composer or Conga Quote Generation. The custom metadata types do not give users access to any other areas of Salesforce outside of the Composer or Conga Quote Generation applications.

## To Add the Necessary Custom Metadata to User Profiles

 If there are multiple users with various Profiles that require the aforementioned Custom Metadata Types, admins can create a new Permission Set with access to the Custom Metadata Types and assign it to users. For more information on creating and assigning Permission Sets, see [Create Permission Sets](#) and [Assign a Permission Set to Multiple Users](#)

1. Navigate to **Salesforce Setup** → **Users** → **Profiles**.
2. Click the relevant custom profile link for your users.
3. Click **Custom Metadata Types** under the Apps section.
4. Click **Edit**.
5. Add the following Custom Metadata Types for each Conga product that the user (that is assigned this specific profile) accesses to the Enabled Custom Metadata Types section. After adding the Custom Metadata Types, click **Save**.

 Older versions of Composer may not require all of the custom metadata types listed below. Check [Critical Update Actions for Composer Versions](#) to see what custom metadata types are required for your version of Composer.

- **Composer**
  - APXTConga4.Composer Setting
  - APXTConga4.Conga Solution Export Environment
  - APXTConga4.Conga Solution Export Setting
  - APXTConga4.Connected App Setting
- **Conga Quote Generation**
  - Salesforce CPQ: Conga Quotes.APXTCFQ.Composer Setting

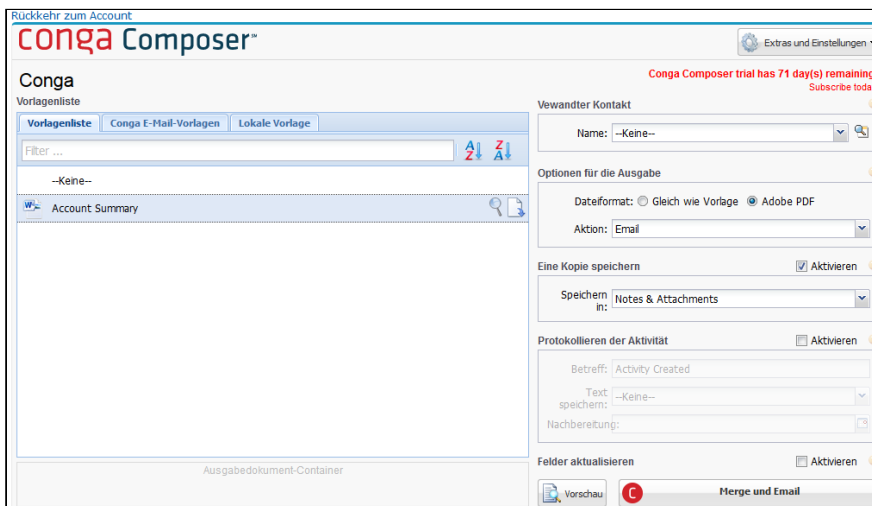
- Salesforce CPQ: Conga Quotes.APXTCFQ.Conga Solution Export Environment
- Salesforce CPQ: Conga Quotes.APXTCFQ.Conga Solution Export Setting
- Salesforce CPQ: Conga Quotes.APXTCFQ.Connected App Setting

## Supported Composer User Interface Languages

This documentation review how to enable German and French dynamic translations in the Composer user interface.

⚠ You must have Conga Composer version 8.20.71 or higher installed in your Salesforce org to leverage German, French, and Polish dynamic translations of the Composer user interface.

Conga Composer's user interface currently supports dynamic translation for German, French, and Polish, in addition to the English language default.



The Composer user interface automatically translates to German or French for Salesforce users that have their local language in Salesforce set to German, French, or Polish.

## To enable dynamic translation in the Composer user interface

1. Under your Salesforce Profile, select **Settings**.

2. Click **Languages & Time Zone**.

3. Change the Language field to Deutsch (German), Français (French), or Polski (Polish).

4. Click **Save**.

## User Management Settings (Legacy Document)

Conga strongly recommends that user management be performed with the "Manage Licenses" tool in Salesforce setup. The Manage Licenses tool found in Salesforce setup is obsolete as the process no longer works consistently. We recommend that all customers switch to the License Management App (LMA) which may require opening a case with Support. [View this article for more information](#). The information below is provided for Legacy customers.

Conga strongly recommends that user management be performed with the "Manage Licenses" tool in Salesforce setup (**Setup** → **Installed Packages** → **Manage Licenses**).

The Salesforce Manage Licenses tool standardizes how user licenses are managed with other AppExchange products and gives greater flexibility for large organizations.

To use this feature, please contact your Conga Business Analyst or Contact Support.

Alternatively, customers with Salesforce instances of 100 or fewer users (of any type) may manage user licenses with the older User Management tool. Larger organizations (more than 100 users) must use the Salesforce Manage Licenses tool.

The User Management tab facilitates the following user license activities:

- Determine whether User Management is enabled for your account.
- View the number of licenses your organization has purchased and how many of those licenses are assigned.
- Sort or search the list of users by Name, Role or Profile.

## Where can I find Salesforce Org ID?

The Salesforce Organization (Org) ID is the unique identifier for your Salesforce identity. Use the steps below to find it.

### Salesforce Classic:

1. Click **Salesforce Setup** (might be located under your name)
2. Below the Administer heading, click **Company Profile**
3. Click **Company Information**
4. Locate your [Salesforce.com](#) Organization ID

Address	913 Ocean Ave Asbury Park, NJ 07712 US	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT-07:00) Mexican Pacific Standard Time (America/Mazatlan)
Activate Multiple Currencies	<input type="checkbox"/>	Improve DATEVALUE() accuracy for DST	<input type="checkbox"/>
Newsletter	<input type="checkbox"/>	Currency Locale	English (United States) - USD
Admin Newsletter	<input type="checkbox"/>	Used Data Space	928 KB (18%) <a href="#">[View]</a>
Hide Notices About System Maintenance	<input type="checkbox"/>	Used File Space	12.7 MB (64%) <a href="#">[View]</a>
Hide Notices About System Downtime	<input type="checkbox"/>	API Requests, Last 24 Hours	12 (15,000 max)
		Streaming API Events, Last 24 Hours	0 (10,000 max)
		Restricted Logins, Current Month	0 (0 max)
		<b>Salesforce.com Organization ID</b>	<b>00Df40000047MrJ</b>
		Organization Edition	Developer Edition
		Instance	NA59
Created By	Walker Llewellyn, 10/17/2018 2:25 PM	Modified By	Walker Llewellyn, 3/20/2019 8:44 AM

### Salesforce Lightning:

1. Click **Salesforce Setup**
2. Click **Setup Home**
3. Below the Settings heading, click **Company Settings**
4. Click **Company Information**
5. Locate your [Salesforce.com](#) Organization ID.

The screenshot shows the Salesforce Lightning interface. At the top, there is a navigation bar with a search bar and several icons. A red arrow points to the gear icon (Settings) in the top right corner. Below the navigation bar, there are several tabs: Service, Home, Chatter, Talent, Contacts, Cases, Reports, Dashboards, and Conga Objects. The main content area displays a 'Quarterly Performance' chart and an 'Assistant' panel on the right.

The Org ID of your production environment is different than your Sandbox Org ID. Whenever a Sandbox is refreshed, a new Org ID is created.



## Support of Salesforce API Version with Composer

The Salesforce API version used by the Composer determines its ability to support Salesforce Standard and Custom Objects. Composer currently uses **Salesforce API Version 52**, which means it supports any Salesforce standard or custom object that requires Salesforce API Version 52 or earlier. The list of Salesforce objects and their supported API versions can be found [here](#).

### Important

- You must use the **Template Builder** to access the new Salesforce objects. You can access the new Salesforce objects within the **Document Automation Wizard** only with the **December '21** release package.
- Composer cannot merge data from Salesforce objects that require Salesforce API Version 53 or higher.
- **By June 1, 2022**, you [must update any custom Composer buttons](#) to a version higher than Salesforce API Version 20.0. Failure to do so will result in the failure of your Composer solutions. For more information, see the [Salesforce Documentation](#).
- **By June of 2023**, you must update any custom Composer buttons to a version higher than Salesforce API Version 30.0. For more information, see the [Salesforce Documentation](#).

Salesforce will deprecate API versions 7.0 through 20.0 on **June 1, 2022**. The API versions in that range are not used by the Conga Composer application. There is no need to take any action if you are using the Composer Lightning Component or OAuth buttons through the solution manager. If you use custom Composer buttons, you must update the Salesforce API version to 21.0 or higher. To avoid interruptions in your Composer service, we recommend that you review your custom buttons and make any necessary updates.



- Composer 8 supports up to version 52.0.
- Composer 7 (legacy version) supports up to version 29.0

**Example:**

Existing Button	Updated Button
<p>This button uses Salesforce API version 20.0, highlighted in red color.</p> <p><a href="https://composer.congamerge.com/?sessionId={!API.Session_ID}&amp;serverUrl={!API.Partner_Server_URL_200}&amp;Id={!Account.Id}&amp;TemplateId=0T_000UAK756438">https://composer.congamerge.com/?sessionId={!API.Session_ID}&amp;serverUrl={!API.Partner_Server_URL_200}&amp;Id={!Account.Id}&amp;TemplateId=0T_000UAK756438</a></p>	<p>Updated the button to a supported Salesforce API version 52.0.</p> <p><a href="https://composer.congamerge.com/?sessionId={!API.Session_ID}&amp;serverUrl={!API.Partner_Server_URL_520}&amp;Id={!Account.Id}&amp;TemplateId=0T_000UAK756438">https://composer.congamerge.com/?sessionId={!API.Session_ID}&amp;serverUrl={!API.Partner_Server_URL_520}&amp;Id={!Account.Id}&amp;TemplateId=0T_000UAK756438</a></p>

## Building Composer Solutions

Information covering the core functionality of Composer, including Solution Manager, Gathering Data, Salesforce Mobile, and Template Basics.

- [Using Composer](#)
- [Using the Document Automation Wizard](#)
- [Using Solution Manager](#)
- [Composer for Lightning](#)
- [Composer for Salesforce](#)
- [Composer Advanced Features](#)
- [Composer Lightning Component](#)
- [Conga Keys](#)
- [Create a Solution Collection](#)

## Using Composer

Information about building Composer button and link URLs, Conga Email Templates, and more.

Select one of the following topics for more information:

- [Creating Conga Composer Solutions](#)
- [Create a Conga Composer Solution Manually](#)
- [\(Optional\) Specify a Document or Email Template for the Solution.](#)
- [Conga Composer URL Examples](#)
- [Add a Conga Composer Button to a Page Layout Manually in Lightning](#)
- [How to Add Button to Page Layout in Classic User Interface](#)

- [Edit a Conga Composer Button Manually](#)
- [Pre-selecting a Conga Email Template in Composer](#)
- [Sending an Email using Conga Email Templates](#)
- [The Conga Email User Interface](#)

## Creating Conga Composer Solutions

Powerful Composer solutions can be created in several different ways. Quick Starts allow you to deploy pre-built solutions in just a handful of clicks. The Composer Solution Manager automates the process of creating your button or link while giving you a point-and-click interface to add parameters and customize your solution. Manual setup allows you to have complete control over the behavior of your solution.

### Quick Starts

Use the Quick Start wizard to get going with Composer solutions quickly. The wizard allows you to select and deploy pre-built solutions complete with queries, templates, and behaviors that are already configured. For more information, see [Composer Quick Starts](#).

### Solution Manager

Use Solution Manager to create solutions in an easy-to-use, point-and-click interface. Solution Manager lets you build your solution from one screen and keep a history of the solution as it changes over time. For more information, see [Create and Manage Solutions using Solution Manager](#).

Solution Manager also has a simplified method to easily create report documents that gather data from several objects, such as dashboard reports. For more information, see [Conga Global Merge](#).

**Requires:** Conga Composer version 8.8 or higher.

### Full Manual Control

For the most flexibility and customizability, you can manually build a solution's button URL. All feature parameters can be added and customized so that the solution performs exactly as you want.

Tips and tricks:

- Copy the managed package button from the Opportunity object to use as a starting point.

- Remember that Composer solutions will automatically make every field on the master object available for you to use.
- Solutions can be buttons or links.
- The Id parameter will be the Id of your master object in 99% of solutions.
- The QueryId parameter must be used in order to pull data using a Conga Query.
- Test the button URL to ensure that it performs as expected before making it available to end-users.
- Once you have created a solution, you can use the Template Builder to add merge fields to a template.
- Add repeating detail data using Microsoft Word's TableGroup and TableStart/ TableEnd parameters.
- Use IF statements to hide or show data based on criteria.
- Save the template in Conga's Template Manager for easy access.

- ✓ After you build and customize your solution, you can automate it if necessary, so that the solution runs whenever a Salesforce® Workflow event occurs or on a pre-set schedule.
  - To hide output choices from your end-users and simplify the process whenever they run a solution, use background mode parameters. For more information, see [Background Mode Parameters](#).
  - To automatically run a solution on a schedule or in batches, use Conga Batch.
  - To integrate your solutions into a workflow, use Conga Trigger.

## Create a Conga Composer Solution Manually

### (Optional) Add Salesforce Reports and Queries

- You can specify up to 50 reports or queries in any combination.
- Each report or query must have an alias.
- Alias names are used to identify datasets within the Conga Template Builder.
- Add reports and queries following this format: `&ReportID=[alias]SalesforceID` and `&QueryID=[alias]SalesforceID`.
- Additional reports and queries are separated by a comma and appear on a new line:

```

https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!ContactId}
&ReportId=[Activities]00OG0000005VR5B,
[Contracts]00OG0000005VR9T,
[Escalations]00OG0000005VQOg,
[Oppty]00OG0000005VR0y

```

Full procedure: [Add Salesforce Reports to a Conga Composer Button URL](#)

## To add Salesforce reports to a button URL

⚠ Before you begin, the button URL must exist and be open in edit mode. If not already open, click **Setup** → **Customize** → **[Object]** → **Buttons, Links, and Actions**, and select the button you want to edit. In addition, all reports or queries must exist and have a Salesforce ID. (You can run the report or query to see the Salesforce ID.)

1. Click at the end of the button URL and press **Enter** to begin a new line.
2. Type **&ReportID=**, the report alias (enclosed by brackets), and the report's Salesforce ID.
3. Type the parameter and value pair using the following format:  
`&[Parameter]=[value]`. Each parameter must include a valid value and may not include spaces. You can replace spaces in a value with the plus sign (+).
4. To add additional reports, separate each report alias and Salesforce ID with a comma.
5. Click **Save**.

⚠ Ensure that the report is visible to the end-user running the solution.

## To add queries to a button URL

⚠ Before you begin, the button URL must exist and be open in edit mode. If not already open, click **Setup** → **Customize** → **[Object]** → **Buttons, Links, and Actions**, and select the button you want to edit. In addition, all reports or queries must exist and have a Salesforce ID. (You can run the report or query to see the Salesforce ID.)

1. Click at the end of the button URL and press **Enter** to begin a new line.
2. Type **&QueryID=**, the alias (enclosed by brackets) and the query's Salesforce ID.
3. Type the parameter and value pair using the following format:  
`&[Parameter]=[value]`. Each parameter must include a valid value and may not


include spaces. You can replace spaces in a value with the plus sign (+). To add additional queries, separate each alias and Salesforce ID with a comma.

4. Click **Save**.

## Add the Button to a Layout

- Where you add the button determines the results, as well as who can see the button.
- Once you add the button to a page layout, all authorized users can see it.
- You can run the solution to view the available datasets and access Conga Template Builder.

## To add a Conga Composer button or link to a page layout

 Before you begin, you must have a button to add.



1. Navigate to **Setup** → **Customize** → **[Object]** → **Page Layouts** and click **Edit** next to the desired page layout.
2. From the Page Layout Properties screen, click **Buttons or Custom Links**.
3. Drag your new button to the Custom Button area on the page layout.
4. Click **Save**.

## Build the Document Merge Template and Save It to the Conga Template Manager

- Run the solution to load a record from the selected master object.
- When you run the solution, the datasets from the reports and queries are available.
- Build your merge template document by using the Conga Template Builder.
- Preview the merged document to ensure the template functions correctly.
- Save the template to the Conga Template Manager to make it available for Conga Composer solutions.


## To build a template using the Conga Template Builder

1. Navigate to the page layout containing the button.
2. Click the button to launch the solution in Conga Composer.
3. In the Conga Composer window, click **Tools & Settings** → **Template Builder** to launch the Template Builder. Conga Composer loads the datasets from the master object, including any detailed datasets you created with an alias by using a report or query.
4. In the Copy Field Format area, select the document type for the template.

5. In the Options area, select or clear the Use Text-Based Merge Fields box to format the fields using the text-based merge field format—such as `{{Product_Name}}`—or the traditional merge field format—such as Product Name: «Product\_Name».
6. In the Options area, select or clear the Include Label box to include or ignore the field label as merge fields are copied into the template region.
7. In the left pane of the Template Builder, locate and double-click a field to add it to the Template Canvas OR click a field name and click the right arrow to add it to the Template Canvas.
8. Repeat the previous step to add additional fields.
9. Click  to have Conga create a new template document containing all of the fields on the Template Canvas OR click individual fields or regions and copy and paste (or drag) them to an open document. You can also Click  to select all fields and then copy or paste the fields into an open document.

## Save the Template Document

Perform the following steps to add a template to the Conga Template Manager:

 **Before you begin, the document template must exist.**

1. Navigate to the Conga Templates tab.
2. Click **New**.
3. On the Conga Template Edit page, in the Name field, type a name for the template. The template will appear in the template list in Composer with this name.
4. (Optional) In the Template Group field, type a template group name (if using the TemplateGroup parameter).
5. In the Template Type field, select the type of template to add.
6. (Optional) In the Master Field to Set fields, enter up to three master fields to update whenever the template is merged. Follow the syntax as specified in the Help Text on the field (in Salesforce).
7. Click **Save** to save the template record in the Conga Template Manager. The template record is created and the Conga Template page appears.
8. On the Conga Template page, click **Attach File**. The Attach File to Conga Template page appears.
9. Click **Choose File**, select the document template to use, and then click **Open**.
10. Click **Attach File**.
11. Click **Done**.

At its simplest, a button URL includes the base URL identifying the master object. More complex solutions can use Salesforce reports and queries, preselect a document or email template, and use parameters to extend the solution's function.

These tasks are part of a larger process of creating a solution. Before creating your button or link, it is helpful to have any reports or queries already designed that are specific to the solution you're creating, as well as a mock-up of the final document you want to produce with Composer®.

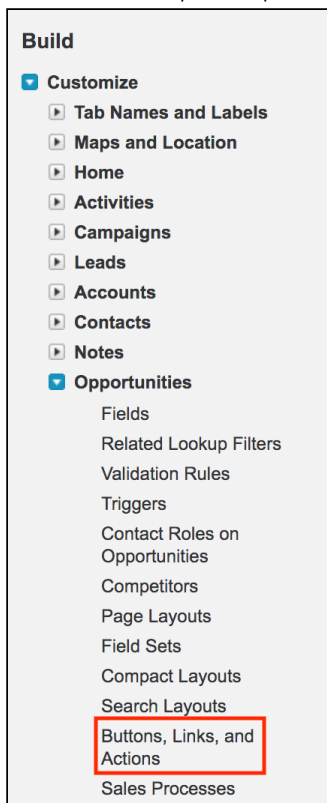
You can also [create a solution using Solution Manager](#) with Conga Composer 8.7 or newer.

## Create the Button or Link

- Add a button label that appears on page layouts.
- Copy the Conga Composer Base URL.
- Identify the master object using the Insert Merge Field drop-down list.

## To create a URL button or link

1. In Salesforce, click **Setup**.
2. In the Customize area, navigate to the object where the button should be placed and click Buttons, Links, and Actions.





- Click the link in the Label column for the sample Conga Composer button to open it.

Action	Label	Name	Description	Type
Edit	Add to Campaign	AddInfluence		
Edit	Clone	Clone		
Edit / Del	Conga_Conductor	Conga_Conductor		Detail Page Link
Edit	<a href="#">Conga Composer</a>	Conga_Mail_Merge	*** DO NOT USE THIS BUTTON ON A PAGE LAYOUT *** COPY THE URL FROM THIS BUTTON and paste it into your own custom button or link. Then add optional parameters to finely tune the button's behavior. See the Conga Configuration Guide for details. *** DO NOT USE THIS BUTTON ON A PAGE LAYOUT ***	Detail Page Button
Edit	<a href="#">Conga Composer (Auth)</a>	Conga_Composer_Auth	OAuth buttons require additional setup. Please see the Composer Installation Guide for more details. COPY THE URL FROM THIS BUTTON and paste it into your own custom button or link. Then add optional parameters to finely tune the button's behavior. See the Conga Configuration Guide for details.	Detail Page Button

- Copy to the clipboard the contents of the Button or Link URL field.  
[https://composer.congamerger.com?SessionId={!API.Session\\_ID}&ServerUrl={!\\$Api.Partner\\_Server\\_URL\\_290}the ID of the object you selected](https://composer.congamerger.com?SessionId={!API.Session_ID}&ServerUrl={!$Api.Partner_Server_URL_290}the ID of the object you selected)
- Press the back arrow to return to the Buttons, Links, and Actions page for the object and click **New Button or Link**.
- Complete the button or link fields:
- In the Label field, type a name for the button or link.
- (Optional) In the Description field, type a description of the button or link.
- Choose a Display Type.
- In the Behavior field, select a behavior option.
- In the Content Source field, accept the default, URL.

**Custom Button or Link Edit** [Save] [Quick Save] [Preview] [Cancel]

Label:

Name:

Description:

Display Type:  Detail Page Link [View example](#)  
 Detail Page Button [View example](#)  
 List Button [View example](#)

Behavior:  [View Behavior Options](#)

Content Source:

Select Field Type:  Insert Field:  Insert Operator:

Customer Receipt (Text) =  
[https://composer.congamerger.com?sessionId={!API.Session\\_ID}&serverUrl={!API.Partner\\_Server\\_URL\\_290}&id={!Opportunity.Id}](https://composer.congamerger.com?sessionId={!API.Session_ID}&serverUrl={!API.Partner_Server_URL_290}&id={!Opportunity.Id})

Functions:   
 ABS  
 AND  
 BEGINS  
 BLANKVALUE  
 CASE  
 CASESAFEID

- In the URL text box, paste the Conga Composer Base URL that you copied previously.
- Click **Save**.

## (Optional) Specify a Document or Email Template for the Solution.

Once the template is added to the Conga Template Manager, you can add it to the solution. You can add up to 10 templates to a solution. Use Conga Email Templates only for solutions where the document format is an email. DocuSign and Adobe eSign parameters are used for templates, recipients, and other settings specific to those integrations.

For more information, see [Pre-selecting a Merge Template for a Conga Composer Solution](#).

### To add a file template to a button URL

#### Before You Begin:

The button URL must exist and be open in edit mode. If it is not already open, click **Setup** → **Customize** → **[Object]** > **Buttons, Links, and Actions**, and select the button you want to edit. The template file must also be saved in the Conga Template Manager.

1. Click at the end of the button URL and press **Enter** to begin a new line.
2. Type **&TemplateID=** and the template's Salesforce ID. For example:  
`&TemplateId=01H5000000005tMS`
3. To add additional templates, separate each Salesforce ID with a comma.
4. Click **Save**.

## Conga Composer URL Examples

Sometimes it is helpful to see a complete Conga Composer URL to fully comprehend how to assemble various parameters. Here are three examples. The base button URL is shown in the normal text; the parameter examples are shown in bold.

### Simple Example

This button URL includes parameters that enable PDF encryption (**ps0=1**) and PDF printing (**ps1=1**). Additional parameters disable the Log Activity button (**ds3=1**), set the template (**TemplateId**), and lock the template choice (**ds4=1**) so that the end-user can't override it.

These examples use non-OAuth buttons. In Salesforce Lightning, you must use the [OAuth buttons](#).

**Example:**

```
https://composer.congamerger.com
?SessionId={!API.Session_ID}
&ServerUrl={!API.Partner_Server_URL_290}
&Id={!Contact.Id }
&PS0=1
&PS1=1
&DS3=1
&TemplateId=0014432553FB
&DS4=1
```

## Multiple Reports

This URL includes four reports (ReportId) with aliases, specifies a template (TemplateId), and completes an Email To field (EmailToId) dynamically using the record's field content.

**Example:**

```
https://composer.congamerger.com
?SessionId={!API.Session_ID}
&ServerUrl={!API.Partner_Server_URL_290}
&Id={!Contact.Id}
&ReportID=[Tasks]00050000001NGuW,
[Events]0005000000102gr,[Partners]00050000001NGv5,
[Escalations]00050000001NGvL
&TemplateId=01H500000005z4L
&EmailtoID={!Contact.Id}
```

## Background Mode

This example URL includes a report (filtered by the Account ID), specifies a template (TemplateId), enables the Background Mode that attaches the document (ds7=1), and forces the output to Adobe Acrobat format (fp0=1).

**Example:**

```
https://composer.congamerger.com  
?SessionId={!API.Session_ID}  
&ServerUrl={!API.Partner_Server_URL_290}  
&Id={!Opportunity.Id}  
&ReportId=[AllContacts]000500000010AA2?pv0={!Opportunity.AccountId}  
&TemplateId=01H500000005xop  
&DS7=1  
&FP0=1
```

## Add a Conga Composer Button to a Page Layout Manually in Lightning

Add a button to a page layout to make it available for end users to launch the solution.

### To add a Composer button or link to a page layout in the Lightning UI

1. Navigate to **Setup** → **Objects and Fields** → **Object Manager** → **[Object]** → **Page Layouts** and select the page layout. The Page Layout Properties screen appears.
2. Click **Mobile & Lightning Actions** or **Custom Links**.
3. Drag and drop your new button to the Salesforce Mobile and Lightning Experience Actions area on the page layout.
4. Click **Save**.

## How to Add Button to Page Layout in Classic User Interface

To add a Conga Composer button or link to a page layout in the Classic user interface:

1. Navigate to **Setup** → **Customize** → **[Object]** → **Page Layouts** and select **Edit** next to the desired page layout. The Page Layout Properties screen appears.
2. Click **Buttons** or **Custom Links**.
3. Drag and drop your new button to the Custom Button area on the page layout.
4. Click **Save**.

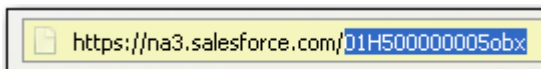
## Edit a Conga Composer Button Manually

You can edit an existing Composer button manually. Do not use the following method if you created the button with the Solution Manager.

1. Click **Setup** → **Customize** → **[Object]** → **Buttons, Links, and Actions**.
2. Select the button you want to edit.
3. Revise the button or link as necessary.
4. Click **Save**.

## Pre-selecting a Conga Email Template in Composer

You can pre-select a Conga Email Template by copying the template's record ID to the clipboard and pasting it into your Composer button's URL following the `&CongaEmailTemplateId=id` parameter. An abbreviated parameter name, `&CETID=id`, may be used instead.



https://na3.salesforce.com/01H500000005obx

⚠ You may only specify one Conga Email Template Id per Composer button / link.

If you specify a Conga Email Template Id, but not a file template id (`&TemplateId`), Composer will display the list of "Conga Email Templates" (with the selected email template) by default.

## Sending an Email using Conga Email Templates

Since Conga Email Templates (CET) can be used in conjunction with traditional File Templates, you select the CET by clicking the Conga Email Templates tab.

To use a Conga Email Template:

1. Click **Conga Email Templates** to view saved templates.
2. Select the email template to use.
3. Click the **Merge & Email button** on the Composer dialog box.

After processing is complete, you are presented with a Salesforce® (or SendGrid® email), fully populated with data and ready for review.

⚠ The Access Level email deliverability setting for your Salesforce org must be set to All Email in order to send emails with Conga Composer. If it is set to No Access or

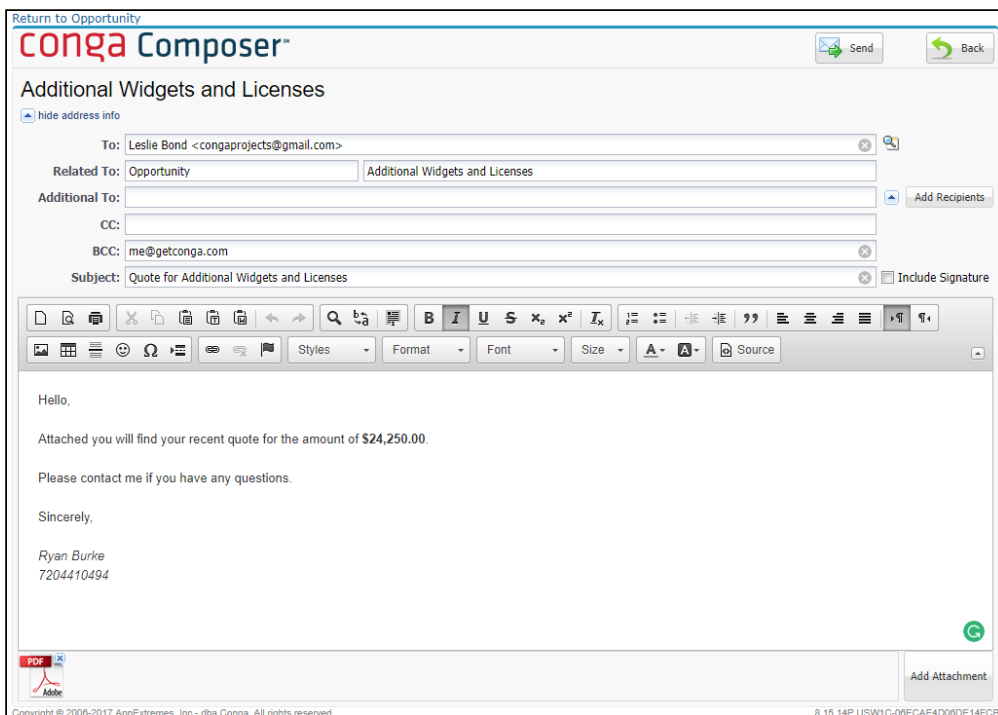
**System Email Only, an error message displays when sending an email through Composer.**

Also, Send through External Email Services must be turned off, or an error message displays when sending an email through Composer. These settings are found by going to **Salesforce Setup → Email Administration → Deliverability and Salesforce Setup → Email Administration → Send through External Email Services.**

## The Conga Email User Interface

You can customize the formatting and delivery options before sending the email by using the Conga Email User Interface.

When an outbound email is generated from a Conga HTML Email Template, SendGrid is enabled, or when merging multiple documents, Composer automatically assembles the email and presents the Conga Email User Interface. If none of those conditions are met, the standard Salesforce Email Task screen is used when emailing a Composer generated document.



## Conga Email User Interface

You can customize the formatting and delivery options before sending the email using the Conga Email User Interface. You can also make these customizations using Composer Email Parameters.

Composer will automatically change the "Related To" value to the master object record when the email is sent so that the resulting completed Task is associated with the master object record. The default "to" recipient is also the Related Contact on the Composer User Interface (typically the Primary Contact on the master record).

## Using the Document Automation Wizard

Easily create a custom Conga Composer solution with the Document Automation Wizard. If you are new to Composer or want to create a customized template, the wizard guides you through the steps to configure a simple solution.

Select one of the following topics for more information:

- [Document Automation Wizard \(DAW\)](#)
- [Behavior Settings](#)
- [Automating Solutions using Background Mode Parameters](#)
- [Composer Lightning Component FAQs](#)
- [Supported Languages for Document Automation Wizard](#)

## Document Automation Wizard (DAW)

Easily create a custom Conga Composer solution with the Document Automation Wizard. If you are new to Composer or want to create a customized template, the wizard guides you through the steps to configure a simple solution.

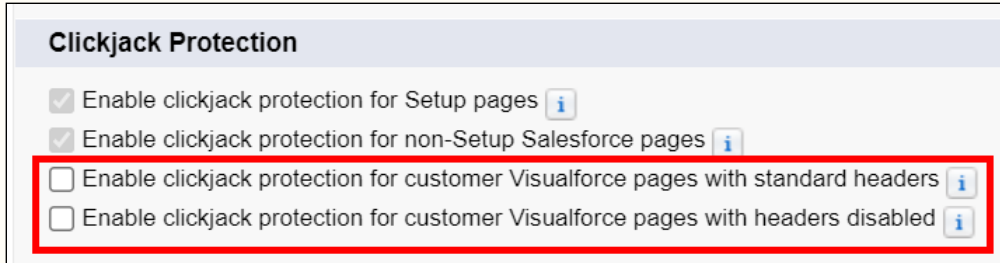
- [WATCH VIDEO](#)

### Important

Conga Composer Quick Start and the Document Automation Wizard are not currently supported (you will receive an error) in Lightning Experience with **Clickjack Protection** enabled. If clickjack protection is enabled, Conga recommends using Salesforce Classic for the best user experience.

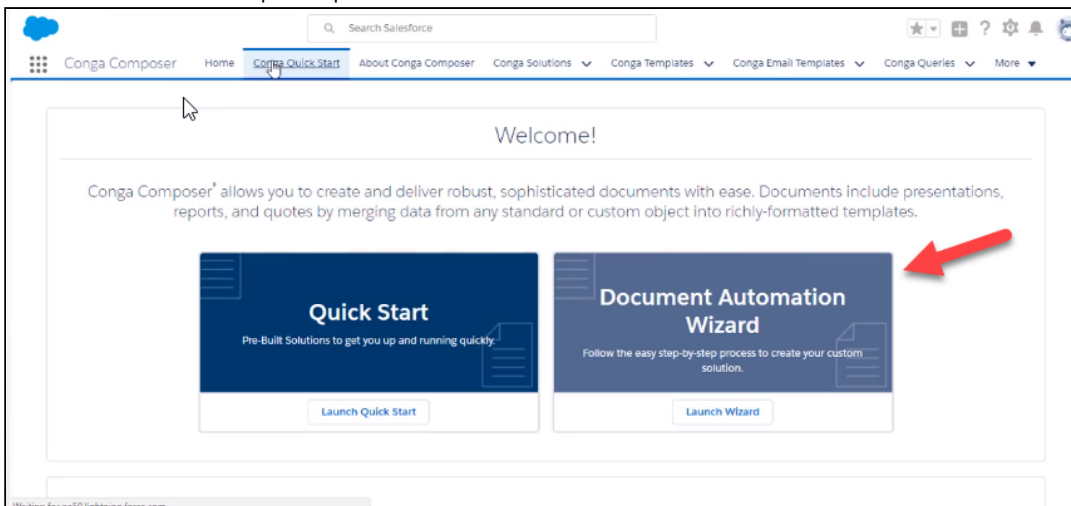
**To disable the Clickjack Protection**

1. On the Salesforce homepage, click the gear icon (⚙️) in the right corner and select **Setup**.
2. Enter **Session Settings** in the **Quick Find** box, then select **Session Settings**.
3. Uncheck the following Clickjack Protection options:



### To create a solution using the Document Automation Wizard

1. Navigate to the Conga Quick Start tab and select **Launch Wizard** on the Document Automation Wizard tile.
2. Follow the wizard prompts.



### Add Solution Details

Name your solution and select the Salesforce object you want to use as your master object, which is the object on which you want to build the Composer button and from where you want to gather master data.

⚠️ Only the first 1000 records are displayed when you search for the Sample Record.



## Add Data to a Solution

If your solution requires data related to the Salesforce Object you selected, you can create and add SOQL queries to access the related content. Locate a query and add an Alias to quickly identify that query in your solution. By default, the data from the Master object selected in the previous step is returned.

**i** Proceed to the **Add Templates to a Solution** step to see if your required fields are available. If not, the fields are not available, simply return back to the **Add Data to a Solution** step.

## Add Templates to a Solution

Select a template you want to be added to your solution. Choose any existing Conga Template and Conga Email Template, or upload your own Word, Excel, PowerPoint, and PDF files.

You can create or upload a new template, in addition to selecting an existing template from the search drop-down menu. The Document Automation Wizard also allows you to reference a formula field for [Dynamic Template Selection](#). You can add up to 10 templates only. For more information, see [Composer Templates](#). The Document Automation Wizard also allows you to create Conga Email Templates directly from the Wizard.

**⚠** Existing Conga Email Templates cannot be searched and added via the wizard. You must add them from the Conga Solution Manager.

## Behavior Settings

Assign common document behaviors to your solution. General behaviors include:

- Background Mode - which allows users to bypass the user interface and go directly into document generation.
- Generate Files - which allows you to convert your document to PDF and assign an output file name.
- Post Merge Distribution - Activity Logging, Email distribution, eSignature, and Save a Copy.
- Additional Behaviors - which allows you to add any parameter and value.

For more about these behaviors, see the [Composer Parameter Guide](#).

## Add a Solution Collection

Reference or create a Solution Collection. Solution Collections allow you to organize and reference multiple Conga Solutions from a Lightning Component.

 You only want to add solutions of the same master object to a Solution Collection.

## Add your newly created button or Lightning Component to the Page Layout.

Click **Go to Page Layout** to add the button to your desired page layout(s) so your end users can start using it to generate documents.

If you added the Solution to a Solution Collection, you have the option to add the Composer Lightning Component by following the steps below:


1. Click **Go to Lightning Builder**.
2. From the Lightning App Builder, click **New**.
3. Select Record Page and click **Next**.
4. Enter a Label name.
5. Under the Object dropdown, select the master object for your solution collection and click **Next**.
6. Under the Choose Page Template section, select a template and click **Finish**.
7. From the right-hand side of the screen, select the Conga Composer Lightning Component to add to your template by clicking and dragging it to a section labeled Add Component(s) Here.
8. On the right-hand side of the screen, select the Solution Collection you want to be included in the component.
9. Click **Save**.
10. Click **Activate**.
11. Select the **App, Record, Type and Report** tab.
12. Click **Assign to Apps, Records Types, and Profiles**.
13. Select the apps where you want to Lightning Component to display.
14. Select **Master** under Record Type and click **Next**.
15. Select the Profiles you want to have permission to access this Lightning Component and click **Next**.
16. Click **Save**.
17. Click **Save** again.

18. When you navigate to a record of the master object selected for your solution, you will see the Conga Composer Lightning Component you created. To run the solution, click on the button with the solution name in the Conga Composer Lightning Component.

## Additional Edits to Your Composer solution

If you want to make additional edits to your solution after using the Document Automation Wizard, follow these steps:

1. Navigate to the **Conga Solutions** tab.
2. Select the newly created Solution Record (identified by the name entered in Step 1 above).
3. Use the [Solution Manager](#) to make edits.

 You can return to the Document Automation Wizard by clicking **Edit (Guided)** on the solution record.

## Behavior Settings

The Document Automation Wizard allows you to set behaviors for your solution. You can set various behaviors, for example, running a merge in background mode, generate a file, log activities, email, assign an eSignature and save a copy of the solution.

You can merge your data in background mode (parameter `ds7=1`) or set automatic activity logging (parameter `AC0=1`) in your solution on the Behavior Setting page. For more information about using basic parameters with your solution, see the [Composer Parameter Guide](#).

## Automating Solutions using Background Mode

### Parameters

Background mode (also known as one-click automation) is an optional way to run Conga Composer solutions in which the user interface is skipped and the document or email template is automatically merged and delivered in the desired manner. Because the user interface is skipped, users can't choose templates, output file options, or activity logging options. Instead, the system administrator pre-defines these options in the Conga Composer solution using parameters.

Using background mode:

- Increases speed of solution by eliminating choices and clicks.

- Streamlines the feel of the solution.
- Increases security by preventing users' choices.
- Decreases errors by locking down the Composer solution.

You set up background mode by adding background mode parameters to your solution URL. The primary parameter for initiating background mode is DS7. The DS7 parameter has multiple parameter values to specify the action that happens in the background. For more information, see [Background Mode Parameters](#).

## Composer Lightning Component FAQs

Information about the Composer Lightning Component.

FEATURE	Composer 8	Composer Lightning Component
Compatible in Salesforce Classic and Lightning	Composer 8 runs in both Classic and Lightning.	The Composer Lightning Component only runs in Lightning.
Debug URL	A Debug URL is available in the Composer user interface to assist with troubleshooting.	A Debug URL is not available to assist with troubleshooting the Composer Lightning Component.
Guided Setup	<p>Users can create Composer solutions using the Conga Solution Manager.</p> <ul style="list-style-type: none"> <li>• Users must access the Composer user interface to use the Template Builder for gathering merge fields and building a new template.</li> <li>• Dynamic Template Selection is allowed in Conga Solution Manager but is not an explicit feature.</li> <li>• Cannot assign a new solution to a Solution Collection in the Conga Solution Manager.</li> </ul>	<p>Users can create Composer solutions with the Lightning Component using the Document Automation Wizard.</p> <ul style="list-style-type: none"> <li>• The Document Automation Wizard allows users to access available merge fields (similar to the Template Builder) without launching Composer.</li> <li>• Enable Dynamic Template selection is an explicit feature in the Document Automation Wizard.</li> <li>• Users can associate a new solution to a Solution Collection in the Document Automation Wizard.</li> </ul>

FEATURE	Composer 8	Composer Lightning Component
Salesforce Reports	Use Salesforce Reports in Salesforce Lightning to supplement a Composer Solution with additional data. Use SOQL Queries for additional data in Salesforce Lightning.	Reports are not an available data source for the Composer Lightning Component. SOQL Queries are used instead of Salesforce Reports.
Composer User Interface	<ul style="list-style-type: none"> <li>• After launching Composer, you can upload a new template on the fly.</li> <li>• Local Template tab</li> <li>• Users can download or click the look-up icon to navigate to a Conga Template or Conga Email Template</li> </ul>	<ul style="list-style-type: none"> <li>• Users can preview a selected document with all of the merged information before actually completing the merge process.</li> <li>• Users can upload and merge local templates in the Composer Lightning Component.</li> <li>• Users can't download or click a look-up icon to navigate to Conga Templates or Conga Email Templates.</li> </ul>
Live Edit	Available	Not Available
Conga Email Templates	Available	Available
QVar parameters	Available	Available
SelectTemplates parameter	Available	Available
TemplateGroup parameter	Available	Available

## Supported Languages for Document Automation Wizard

**Document Automation Wizard** supports dynamic translation for German, French, Dutch, Polish, and Spanish, in addition to English as a default language. It automatically

translates to German, French, Dutch, Polish, or Spanish for the Salesforce Users based on their local language set in Salesforce.

### ⚠ Important

- You must have a minimum *Conga Composer version 8.198-Longs* installed in your Salesforce org to leverage German, French, Dutch, Polish, and Spanish dynamic translations.
- Due to Salesforce Limitations, the following custom tabs and labels in corresponding child pages are not translated into German, French, Dutch, Polish, and Spanish:
  - Conga Solutions
  - Conga Templates
  - Conga Email Templates
  - Conga Queries
  - Conga Global Merges
  - Document Histories

To set dynamic translation

1. Under your Salesforce Profile, select **Settings**.
2. Click **Languages & Time Zone**.

The screenshot shows the 'Language & Time Zone' settings page in Salesforce. The page title is 'Language & Time Zone'. Below the title, there is a 'Settings' section with the following fields:

- Time Zone:** (GMT-08:00) Pacific Standard Time (America/Los\_Angeles)
- Locale:** English (United States)
- Language:** Deutsch (highlighted with a red box)
- Email Encoding:** General US & Western Europe (ISO-8859-1, ISO-LATIN-1)

At the bottom of the settings section, there are 'Save' and 'Cancel' buttons.

3. Change the Language field to Deutsch (German), Français (French), Nederlands (Dutch), Polski (Polish), or Español (Spanish).
4. Click **Save**.

## Using Solution Manager

Solution Manager is a feature in Conga Composer version 8.7 and newer that provides a point and click user interface for admins to create and manage Conga Composer solutions.

Select one of the following topics for more information:

- [Solution Manager Basics](#)
- [Creating Composer Buttons with Solution Manager](#)
- [Global Merge for Composer](#)
- [Solution Packs](#)

## Solution Manager Basics

Basics of using Solution Manager to create and manage Composer solutions.

Requires Conga Composer version 8.7 or newer.

Select one of the following topics for more information:

- [About Solution Manager](#)
- [Create and Manage Solutions with Solution Manager](#)
- [Solution Manager Button URLs](#)
- [About the Button Syntax and Test Launcher Area in Solution Manager](#)
- [About the Conga Solution Query Screen](#)
- [Using Solution Manager with Salesforce Professional Edition](#)
- [Conga Composer Solution Migration Utility Tool](#)

## About Solution Manager

Conga Composer Solution Manager dramatically accelerates the creation of document automation solutions. Conga Composer's Solution Manager provides an all-in-one workspace to create and manage solutions.

To effectively and fully use Solution Manager, Unlimited Edition and Enterprise Edition Salesforce users must enable the Modify All Data and Customize Application permissions at the Salesforce Profile level. Failure to enable these permissions prevents you from creating or updating your buttons.

From one Solution Manager record, you can:

- Add the [reports](#) and [queries](#) that gather data for the solution
- Add new and [existing templates](#) to style the merged documents
- Customize your solution with [parameters](#) to fine-tune and expand the solution

- Test your solution using a [sample record](#) from the dataset (without exposing the solution button to end-users)
- Add the [solution button to a page layout](#)

In addition, Solution Manager maintains an archive of the button changes, so you can review a button URL's history and can easily revert to a previous version. Solution Manager is included in Conga Composer release 8.7 and above. Solution Manager also includes an integrated Global Merge (formerly called QuickMerge), to quickly produce reports and gather information from multiple objects. You can also build button URLs manually. For more information, see [Build Button and Link URLs Manually](#).

## Create and Manage Solutions with Solution Manager

### Create the Solution Record

- Create the container (the solution record) that organizes the solution settings.
- Give the solution a name.
- Identify the master object.
- Create the button.

To create a new Solution record


1. Click the **Conga Solutions** tab.
2. Click **Create New** and select **Conga Solution**.
3. In the Conga Solution Name field, enter a name for the solution.
4. In the Master Object field, select a master object from the list.
5. Click **Create Record**. The new solution opens in the Conga Solution screen.

### Create the Button

To create the button in Solution Manager, click **Create** on the Conga Solution window.

(Optional) Specify the data needed for the merged documents. Some solutions can receive all necessary data from the master object. If your solution requires data from other objects, you can include reports or queries to gather that data and make it available.

- You can specify up to 50 reports or queries in any combination.
- Each report or query must have an alias.
- Alias names are used to identify datasets within the Conga Template Builder.

 Although Conga Composer supports up to 50 reports or queries, numerous or complicated reports can exceed the timeout of 60 seconds in Salesforce. There are



other limitations that you should consider. For more information on limits, see [Product Data Limits](#).

## Add an Existing Query

To add an existing query to a solution using Solution Manager:

**⚠ Before you begin, you must have a solution record created and open.**

1. In Solution Manager, click **Add Query Data**.
2. Select a query. The query name and ID appear in the boxes to the right.
3. In the Alias field, type an alias for the query.
4. (Optional) In the Description field, type a text description of the query.
5. (Optional) To modify the default filter of the query, click **Advanced Filter Configuration**.
6. Click **Add Conga Query to Solution Record**. Solution Manager adds the query to the record. The query appears in the Selected Queries area.
7. (Optional) Add additional existing queries to the record. Each solution can access up to 50 entries, in a combination of reports and queries.
8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

## Create and Add a New Query

To create a new query and add it to a Solution Record:

- Watch [a video](#) of this task

**⚠ Before you begin, you must have a solution record created and open.**

1. On the Conga Solution screen, click **Add Query Data**.
2. On the Conga Solution Query screen, click **Create New Conga Query Record**. The Conga Query Edit screen opens in a new window.
3. In the Name field, type a name for the query.
4. (Optional) In the Description field, type a text description of the query.
5. Click **Save** to save the query.
6. In the SOQL Select Statement box, type the select statement for the query.

**Or**

Use the Conga Query Builder button to build a SOQL statement.

1. Click **Save** to save the query. Solution Manager adds the query and displays the Conga Query screen with the record loaded.
2. Return to the Conga Solution Query screen and click **Refresh** to view the new query.
3. In the All Available Conga Queries area, click the query to select it. The report name and ID appear in the boxes to the right.
4. In the Alias field, type an alias for the report.
5. (Optional) In the Description field, type a text description of the report.
6. (Optional) To modify the default filter of the query, click Advanced Filter Configuration.
7. Click **Add Conga Query to Solution Record**. Solution Manager adds the query to the record. The query appears in the Selected Queries area.
8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

#### **Important**

While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

## Add an Existing Report

To add an existing report to a solution using Solution Manager:

- Watch a [video](#) of this task.

#### Before you begin, you must have a solution record created and open.

1. In Solution Manager, click **Add Report Data**.
2. Select a report. The report name and Id appear in the boxes to the right.
3. In the Alias field, type an alias for the report.
4. (Optional) In the Description field, type a text description of the report.
5. (Optional) To modify the default filter of the report, click **Advanced Filter Configuration**.
6. Click **Add Conga Report to Solution Record**.  
Each solution can access up to 50 entries, in a combination of reports and queries. Solution Manager adds the report to the record. The report appears in the Selected Reports area.
7. (Optional) Add additional existing reports to the record.

8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

## Create and Add a New Report

To create a new report and add it to a Solution Record:

 **Before you begin, you must have a solution record created and open.**

1. On the Conga Solution screen, click **Add Report Data**.
2. On the Conga Solution Report screen, click **Create New Report**. The Create New Report screen opens in a new window.
3. In the Select Report Type area, select a report type. A preview of the report is shown in the right pane.
4. Click **Create**. The standard Salesforce Report screen appears with the unsaved report information. Use this screen to build and save your report.
5. Click **Help for this Page** in the upper right of the screen to [access Salesforce help on using the Report Builder](#).
6. (Optional) Click **Run Report** to run the report to view the results.
7. Return to the Conga Solution Report screen and click **Refresh** (icon) to view the new report.
8. In the All Available Reports area, click the report to select it. The report name and ID appear in the boxes to the right.
9. In the Alias field, type an alias for the report.
10. (Optional) In the Description field, type a text description of the report.
11. (Optional) To modify the default filter of the report, click **Advanced Filter Configuration**.
12. Click **Add Conga Report to Solution Record**. Solution Manager adds the report to the record. The report appears in the Selected Reports area.
13. (Optional) Add additional existing reports to the record. Each solution can access up to 50 entries, in a combination of reports and queries.
14. Click **Return to Solution Record**.
15. Click **Update Button** to add your changes to the current button URL. The report name and ID appear in the boxes to the right.

## Select a Sample Record and Build the Document Merge Template.

Click **Access Template Builder** and then click **Select Sample Record** to load a record from the selected master object.

When you select a sample record, the datasets from the reports or query are available.

To select a sample record in Solution Manager:

**⚠ Before you begin, you must have a solution record created and open. In addition, you must have created the button.**

1. On the Solution Manager screen, click **Access Template Builder**. The Conga Solution Sample Record screen appears, showing a list of sample records.
2. In the Sample Opportunity Records box, click a record to select it. Solution Manager updates the Sample Record ID field.
3. Click **Select Sample Record**.
4. Click **Return to Solution Record** to return to the solution record.

**Or**

Click **Access Template Builder** to open the Conga Composer for the selected record to preview the merged document or create a new template. If no existing template is available, you can build one.

To build a template using the Conga Template Builder:


- Watch [a video](#) of this task.

**⚠ Before you begin, the solution record must be loaded and a sample record selected.**

1. On the Conga Solution Record, click **Access Template Builder**.
2. Verify that a sample record is selected.
3. Click **Access Template Builder** to launch the solution in Conga Composer.
4. In the Conga Composer window, click **Tools & Settings** → **Template Builder** to launch Template Builder. Conga Composer loads the datasets from the master object, including any you created with an alias by using a report or query.
5. In the Copy Field Format area, select the document type for the template.
6. In the Options area, select or clear the **Use Text-Based Merge Fields** box to format the fields using the text-based format –such as `{{Product_Name}}` – or the traditional merge field format—such as Product Name: «Product\_Name».
7. In the Options area, select or clear the **Include Label** box to include or ignore field label as merge fields are copied into the template region.
8. In the left pane of the Template Builder, locate and double-click a field to add it to the right pane.

**Or**

In the left pane of the Template Builder, click a field name and click the right arrow to add it to the right pane.

1. Repeat the previous step to add additional fields.
2. Click  to have Conga create a new template document containing all of the fields in the right pane.

Or

Click individual fields or regions and copy and paste (or drag) them to an open document.

Or

Click  to select all fields and then copy or paste the fields into an open document.

1. Save the template document.
2. Click **Back** to return to the Conga Composer window. To add a template to the Conga Template Manager:
3. On the Conga Solution page, in the Useful Links area, click **Create New Conga Template Record**.
4. On the Conga Template Edit page, in the Name field, type a name for the template.
5. (Optional) In the Template Group field, type a template group name (if using the TemplateGroup parameter).
6. In the Template Type field, select the type of template to add.
7. (Optional) In the Master Field to Set fields, enter up to three master fields to update whenever the template is merged.
8. Click **Save** to save the template record in the Conga Template Manager. The template record is created and the Conga Template page appears.
9. On the Conga Template page, click **Attach File**. The Attach File to Conga Template page appears.
10. Click **Choose File**, select the document template to use, and then click **Open**.
11. Click **Attach File**.
12. Click **Done**.

## Select a Merge Template for the Solution.

- Once the template has been added to the Conga Template Manager, you can add it to the solution.
- Use Conga Email Templates only for solutions where the document format is an email.
- You can add up to 10 file templates using Solution Manager and only one email template.

### Add an Existing File Template to the Solution Record

To add an existing template to a solution using Solution Manager:

- Watch [a video](#) of this task.

**⚠ Before you begin, you must have a solution record created and open.**

1. In Solution Manager, click **Add Templates**.
2. Click a template to select it. The template name and ID appear in the boxes to the right.
3. (Optional) In the Description field, type a text description of the template.
4. Click **Add Conga Template to Solution Record**. Solution Manager adds the template to the record. The template appears in the Selected Templates area.
5. Click **Return to Solution Record**.
6. Click **Update Button** to add your changes to the current button URL.

## (Optional) Add an Email Template

To add a Conga email template:

**⚠ Before you begin, you must be editing a solution record.**

1. On the Conga Solutions page, click **Add Conga Email Template**.
2. Select the Conga Email Template you want to add to your solution. You can only have one Conga Email Template per solution.
3. (Optional) In the Description field, add a free-form text description.
4. Click **Add Conga Email Template** to select the desired template and add it to the solution.
5. (Optional) Click **Save** to save your solution record settings.

**Or**

If the button has been created, click **Update Button** to update the button URL with the current settings.

## Customize the Solution and Add Functionality with Parameters

- Parameters add flexibility and power to a solution.
- You can use multiple parameters in a solution.
- Some parameters require other parameters to function.
- Parameters require a value.

## Add Parameters

To add parameters to a solution:

**⚠ Before you begin, you must have a solution record created and open.**

1. Click **Customize** with Parameters to load the Conga Composer Parameter screen. The parameter list is divided out into sections based on functionality. Click the plus sign to expand a section and view the parameters in that category. Click again to hide the parameters in the category.
2. Click a parameter to see information and settings for the parameter. The description pane lists parameters and other settings that are required to use the selected parameter.
3. Click **Select Parameter** to select the parameter. The Solution Manager Parameter Picker auto-fills the Parameter Name field.
4. Enter a value in the Parameter Value field. The Parameter Picker will enter a default value of 1 automatically for parameters with On/Off functionality. This is a required field.
5. (Optional) To add a merge field to your parameter, click **Merge Field Helper** to view the available merge fields.
6. Click **Add Parameter** to Solution Record to save the selected parameter to your solution.
7. Repeat to add additional parameters.
8. Click **Return to Solution Record** to return to the solution record where you can update the button URL.

## Test the Solution

- Use the Launch button to run the solution before adding it to a layout.
- Make any changes prior to publishing the solution to end users.

To test a solution:

1. Click **Select Sample Record**.
2. Click the record you want to use as the sample record for testing. It will auto-populate the Sample Record ID field.
3. Click **Submit & Return to Solution Record**. This will add the sample record to your solution and allow you to test your solution against this record. The Launch button is now enabled in the Button Syntax & Test Launcher area.
4. Click **Launch** to launch Conga Composer. You can now run a merge, check your settings, open the Template Builder, and access any other parts of Composer necessary to adequately test your solution.

## Add the Button to a Layout

Where you add the button determines the results, as well as who can see the button.

To add or remove a Solution Manager button on a page layout:

**⚠ Before you begin, you must have a solution record open with a button created.**

1. In Solution Manager, click **Manage Page Layouts**.
2. In the Available Page Layouts area, select a check box next to a page layout to add the solution button to it. You can add buttons to more than one page layout.
3. Click **Select All** to add the button to all available layouts.
4. Click **Add to Selected Layout(s)** to add the selected buttons to the layout.

Or

Click **Remove from Selected Layout(s)** to remove the buttons from the layout. (Optional)

Click **Return to Solution Record**

## Solution Manager Button URLs

Overview of Solution Manager UI.



If you edit your button URL directly from the button, those changes will not be saved in the Solution record. If you go back into the Solution Manager, make additional changes, and then click Update Button, only the changes made in the Solution record will be updated and pushed out to the active button (however, the changes made on the button itself will be saved in the archive as a Note on the Solution Manager record).

1	A real-time view of the solution as it is constructed thus far from the Solution Manager record. As you work through a solution and update, the Button Configuration in Progress area will update in real time, but the URL will not become part of the button until you click Create Button or Update Button.
2	A view of the currently saved button. The current button configuration appears after you click Create Button and updates after each click of Update Button.
3	Launches Conga Composer® so that you can test the currently saved button.



## About the Button Syntax and Test Launcher Area in Solution Manager

As you have added the report, queries, parameters, etc., in the previous steps, your solution has been dynamically building. You can see the dynamic solution displayed in the Button Syntax & Test Launcher section.

Once you have added all the desired elements to your solution, you may create the Conga Composer button.

As you have added the report, queries, parameters, etc., in the previous steps, your solution has been dynamically building. You can see the dynamic solution displayed in the Button Syntax & Test Launcher section.

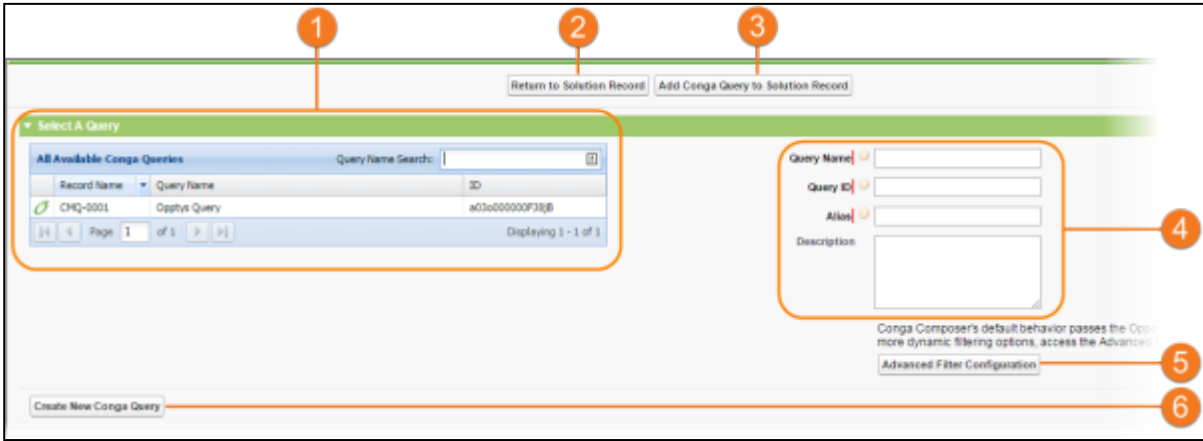


1	A real-time view of the solution as it is constructed thus far from the Solution Manager record. As you work through a solution and update, the Button Configuration in Progress area will update in real time, but the URL will not become part of the button until you click Create Button or Update Button.
2	A view of the currently saved button. The current button configuration appears after you click Create Button and updates after each click of Update Button.
3	Launches Conga Composer so that you can test the currently saved button.

## About the Conga Solution Query Screen

Conga® Solution Query screen.

In addition to using Salesforce® reports to gather information from Salesforce, you may also use SOQL Queries stored in the Conga® Query Manager.



1	A list of all queries available to add to your solution. Queries are stored in and retrieved from the Conga Query Manager.
2	Click <b>Return to Solution Record</b> to close the Add Queries screen and return to the solution record.
3	Click <b>Add Conga Query to Solution Record</b> to add a selected query to the solution.
4	Once you select a query, the Query Name and Query ID fields auto-populate. Add an Alias (required) and a Description (optional).
5	(Optional) Click <b>Advanced Filter Configuration</b> to add advanced filters (pv0, pv1, pv2) to the selected query.
6	(Optional) Click the <b>Create New Conga Query</b> button to open the Conga Query Manager in a new window, allowing you to create an ad-hoc query.

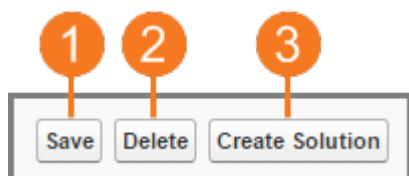
## Using Solution Manager with Salesforce Professional Edition

Not all features are available in the Professional Edition license of Salesforce®, but you can still build the button syntax, and then copy and paste it into the button created on the Master Object.

⚠ Not all features are available in the Professional Edition license of Salesforce®. Professional Edition users do not have access to creating the button and adding it to a Page Layout. However, you can build the button syntax and then copy and paste it into the button created on the Master Object. Thus, you will be creating a solution with the Solution Manager and then manually creating a Conga button.

For more information about this process, see [Create a Conga Composer Solution Manually](#). This edition does not support the Solution Wizard.

Below are actions that you can perform using the available buttons:



1	Save: Saves the solution from Solution Manager.
2	Delete: Deletes the solution from Solution Manager. This will delete only the Solution Manager record, not the button or the button URL from the Buttons, Links and Actions menu on your Master Object.
3	Create Solution : Creates a new solution.

## Conga Composer Solution Migration Utility Tool

The Conga Composer Solution Migration Utility tool allows you to export and import solution records between Salesforce Orgs. For example, you can export a solution record you created in a Sandbox Org and import it into your Production Org.

**⚠ The solution must be built with the Solution Manager or Document Automation Wizard.**

The Solution Migration Utility allows a solution to be migrated from one org to another. Upon import in the target org, the solution is created as new and does not map to or update existing solutions. The solution adds the Conga\_Composer\_ object to every new solution and you cannot change it or the solution breaks.

**⚠ The Solution Migration Utility Tool currently does not migrate Conga Queries referenced in QVar parameters and their retrieved record Ids, such as Conga Template and Conga Email Template Ids.**

### To migrate a solution

1. Log in to the Salesforce org where the solution is being migrated from.
2. Navigate to the Solution Record you want to export.
3. Click on the **Export Solution button**.
4. Click **Download** and navigate to the folder where you want to save the .zip file.

❗ **Salesforce CRM Content must be enabled in both Salesforce orgs for the Export Solution to work.**

## To import a Solution

1. Log in to the Salesforce org where the solution is being migrated to.
2. Navigate to Conga Composer Setup.
3. Click on the **Solution Migration** tab.
4. Click **Browse** and navigate to the .zip file you want to upload.
5. Click **Import**.

If the solution migration fails, you can see in-depth error messages with details on why the Solution Migration fails from the browser console panel.

## To troubleshoot Error Messages within the Browser Console Panel

1. Right click on the browser page.
2. Click **Inspect**. (Shortcut key: *Ctrl+Shift +I*)
3. Click the **Console** tab.
4. Review error message details on why the solution migration is failing.

## Creating Composer Buttons with Solution Manager

Information about using Solution Manager, including adding and removing components as well as running solutions.

Select one of the following topics for more information:


- [Create a Solution Record in Solution Manager](#)
- [Create the Conga Composer Button with Solution Manager](#)
- [Update a Button URL in Solution Manager](#)
- [Adding Reports](#)
- [Add an Existing Report to a Solution using Solution Manager](#)
- [Add a New Report to a Solution using Solution Manager](#)
- [Add Advanced Filter Configuration for a Report Entry in Solution Manager](#)
- [Add an Existing Query to a Solution using Solution Manager](#)
- [Add a New Query to a Solution using Solution Manager](#)
- [Adding File Templates](#)
- [Add an Existing Template to a Solution using Solution Manager](#)
- [Add a New Template to a Solution using Solution Manager](#)
- [Adding Email Templates](#)

- [Add a New Conga Email Template to a Solution using Solution Manager](#)
- [Add an Existing Conga Email Template to a Solution using Solution Manager](#)
- [Select a Sample Record in Solution Manager](#)
- [The Conga Solution Parameter Screen](#)
- [Add a Parameter to a Solution using Solution Manager](#)
- [View Solution and Button Archive Components](#)
- [Add a Note to the Button Archive using Solution Manager](#)
- [Remove Solution Components in Solution Manager](#)
- [Add or Remove Buttons on Page Layouts](#)
- [Run a Solution from Solution Manager](#)
- [Remove a Button from a Page Layout in Solution Manager](#)
- [Best Practice for Editing a Button in Solution Manager](#)
- [Manually Adding a Parameter with Solution Manager](#)
- [Remove a Template Entry in Solution Manager](#)

## Create a Solution Record in Solution Manager

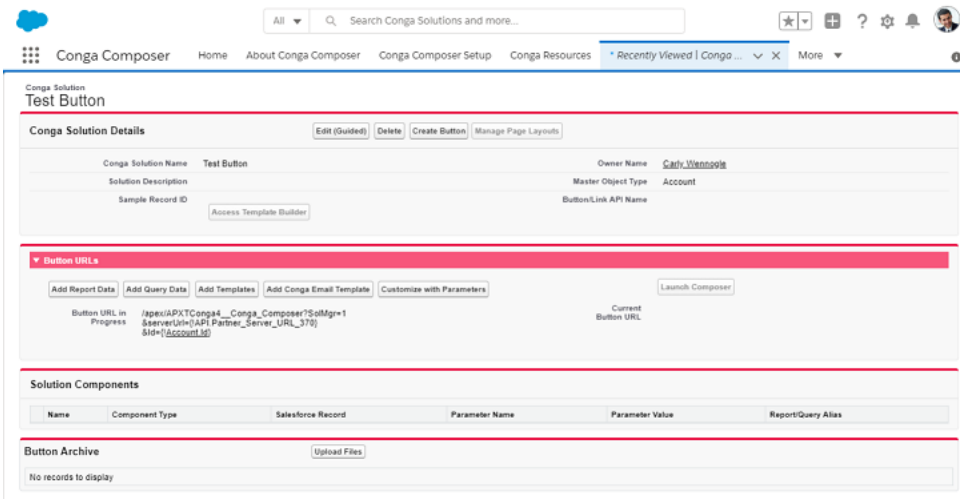
Before you can build a solution in Solution Manager, you must first create a solution record. The record is a container for solution information, including the button URL history and notes.

To create a new solution record in Solution Manager:

 To fully utilize solution records, you should be a System Administrator (or have the Modify All Data and Customize Installed Applications profile settings).

1. Click the **Conga Solutions** tab.
2. Click **New** to create a new Conga Solution record.
3. In the Conga Solution Name field, enter a name for the solution.
4. In the Master Object field, select a master object from the list.
5. Click **Create Record**.

The new solution opens in the Conga Solution screen.



- Add report data (using existing or creating new reports) or query data (using existing or creating new queries).

## Create the Conga Composer Button with Solution Manager

Once you have added elements to your solution, you can create the Conga Composer button. As you build the solution and before you create the button, the button URL changes are shown in the Button URL in Progress area.

After you create the button, Solution Manager copies the URL into the Current Button URL area. In addition, Create Button changes to Update Button, which you can use to update the button URL whenever you make changes to the solution.

To create the button in Solution Manager:


1. On the Conga Solution screen, click **Create Button**.

After you build the button, the Manage Page Layouts button enables, so that you can add the button to the page layout.

## Update a Button URL in Solution Manager

Whenever the Button URL in Progress is different from the Current Button URL, you can quickly update the current button to match the button in progress.

To update a button URL in Solution Manager:

 **Before you begin, you must have a solution record created and open.**

1. On the Solution Record screen, click **Update Button**.

Solution Manager copies the URL from the Button URL in Progress field to the Current Button URL field and adds an entry in the Button Archive area. The Button Archive section retains a history of all button URLs changed in Solution Manager.

Solution Manager is not bi-directional. While the Current Button Configuration in Solution Manager will always display the actual button configuration, any changes made directly in button URL will not update Solution Manager. Update Button in Solution Manager will override any changes made directly to the button URL.

If changes have been made directly to the button URL in Buttons, Links and Actions, an informative message appears when the record is accessed in Solution Manager:

This message serves as an indicator that the button URL has changed outside of the Solution Manager. This message disappears after the button has been updated in Solution Manager.

## Adding Reports

About Conga Solution Report screen.

**⚠ While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.**

Specifying reports in your Conga Composer solution allows you to gather data in addition to the Master Object data for use in the merge process.

## The Conga Solution Report Screen

1	<p>The Select a Report area displays a list of all reports available to add to your solution. Once you select a report, the Report Name and Report ID fields automatically populate. Add an Alias (required) and a Description (optional).</p> <p>Solution Manager shows all reports in your instance, including those saved into the Conga Composer Reports folder.</p>
2	<p>Click Return to Solution Record to close the Add Reports screen and return to the solution record.</p>
3	<p>Click Add Report to Solution Record to add a selected report to the solution.</p>

4	(Optional) Click Advanced Filter Configuration to add advanced filters (pv0, pv1, pv2) to the selected report.  By default, Conga Composer will pass the Master Object ID to the first filter on the report.
5	The Create New Report button opens Salesforce reports in a new window, allowing you to create a report on the fly, if needed.
6	All reports that have been added to the solution appear in the Selected Reports area.

## Add an Existing Report to a Solution using Solution Manager

You can add new or existing reports to your solution. If the report exists, you need only select it and assign an alias. If the report you need isn't available, you can create one within Solution Manager.

**⚠ While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.**

By default, Conga Composer passes the Master Object's Record ID to the first filter of your report. You can modify and add dynamic filtering options using [Advanced Filter Configuration](#).

To add an existing report to a solution using Solution Manager:

- [Watch a video of this task](#)


**⚠ You must have a solution record created and open.**

1. In Solution Manager, click **Add Report Data**. The Conga Solution Report screen appears. All available reports appear in the list.
2. Click a report to select it. The report name and ID appear in the boxes to the right.
3. In the Alias field, type an alias for the report.
4. (Optional) In the Description field, type a text description of the report.
5. (Optional) To modify the default filter of the report, click **Advanced Filter Configuration**. For details, see [Add Advanced Filter Configuration for a Report Entry in Solution Manager](#).
6. Click **Add Conga Report to Solution Record**. Solution Manager adds the report to the record. The report appears in the Selected reports area.



7. (Optional) Add additional existing reports to the record. Each solution can access up to 50 entries, in a combination of reports and queries.
8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

## Add a New Report to a Solution using Solution Manager

 While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

If the report you need isn't already available, you can create a new report within Solution Manager. If the report is available, you can [add the existing report to the solution record](#).


To create a new report and add it to a Solution Record:

 You must have a solution record created and open.

1. On the Conga Solution screen, click **Add Report Data**.
2. On the Conga Solution Report screen, click **Create New Report**. The Create New Report screen opens in a new window.
3. Search for the report type. Type in the search box to display matching report types.
4. Select a report type.
5. Click **Create** to create the new report. The standard Salesforce Report screen appears with the unsaved report information. (Optional) Click **Help for this Page** in the upper right of the screen to [access Salesforce help on using the Report Builder](#).
6. (Optional) Click **Run Report** to run the report to view the results.
7. Return to the Conga Solution Report screen and click **Refresh** to view the new report.
8. In the All Available Reports area, click the report to select it.

 The report name and ID appear in the boxes to the right.

9. In the Alias field, type an alias for the report.
10. (Optional) In the Description field, type a text description of the report.
11. (Optional) To modify the default filter of the report, click **Advanced Filter Configuration**. For details, see [Add Advanced Filter Configuration for a Report Entry in Solution Manager](#).
12. Click **Add Conga Report to Solution Record**.

 Solution Manager adds the report to the record. The report appears in the Selected reports area.

13. (Optional) Add additional existing reports to the record.

⚠ Each solution can access up to 50 entries, in a combination of reports and queries.

14. Click **Return to Solution Record**.
15. Click **Update Button** to add your changes to the current button URL.
16. The report name and ID appear in the boxes to the right.

## Add Advanced Filter Configuration for a Report Entry in Solution Manager

How to add advanced filter configuration for a Report Entry in Solution Manager. You can enable advanced filters (pv values) in reports using Solution Manager.

⚠ While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

You can enable advanced filters (pv values) in reports using Solution Manager.

To enable advanced filter configuration in a report:

1. Click **Advanced Filter Configuration**. The page will refresh and Advanced Filter Configuration will become available under Reports.


⚠ Be aware of the filters in your selected report and ensure that these same filters are added as pv values in the Solution Manager. If the filters in the report and filters in Solution Manager do not match exactly then the solution will not work correctly.

2. Click the field name you want added to the pv0 Value field. The field will flash green indicating the selected Merge Field has been added to that filter. The pv value fields are added in the order that you click them. For example, the first click populates pv0 Value, second click populates pv1 Value, and so on.) It is important to click fields in the order that you want them to populate the corresponding value. More than one parameter value (pv) is rarely used except in very advanced solutions.
3. Click **Add Report** to add the report to your solution. The report you added appears in the Selected Reports section.

## Add an Existing Query to a Solution using Solution Manager

You can add new or existing queries to your solution. If the query exists, you need only select it and assign an alias to the query. If the query you need doesn't exist, you can create one within Solution Manager. By default, Conga Composer passes the Master Object's Record ID to the first filter of your query. You can modify and add dynamic filtering options using [Advanced Filter Configuration](#).

To add an existing query to a solution using Solution Manager:


 Before you begin, you must have a solution record created and open.

1. In Solution Manager, click **Add Query Data**. The Conga Solution Query screen appears. All available queries appear in the list.
2. Click a query to select it. The query name and ID appear in the boxes to the right.
3. In the Alias field, type an alias for the query.
4. (Optional) In the Description field, type a text description of the query.
5. (Optional) To modify the default filter of the query, click **Advanced Filter Configuration**. For details, see [Add Advanced Filter Configuration for a Report Entry in Solution Manager](#).
6. Click **Add Conga Query to Solution Record**. Solution Manager adds the query to the record. The query appears in the Selected Queries area.
7. (Optional) Add additional existing queries to the record. Each solution can access up to 50 entries, in a combination of reports and queries.
8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

## Add a New Query to a Solution using Solution Manager

You can create a new query for your solution.

To create a new query and add it to a Solution Record:


 Before you begin, you must have a solution record created and open.

1. On the Conga Solution screen, click **Add Query Data**.
2. On the Conga Solution Query screen, click **Create New Conga Query Record**. The Conga Query Edit screen opens in a new window.
3. In the Name field, type a name for the query.
4. (Optional) In the Description field, type a text description of the query.
5. Click **Save** to save the query.

6. In the SOQL Select Statement box, type the select statement for the query OR use the Conga Query Builder button to build a SOQL statement.

To use the Conga Query Builder:


1. Click **Save** to save the query. Solution Manager adds the query and displays the Conga Query screen with the record loaded.
2. Return to the Conga Solution Query screen and click **Refresh** to view the new query.
3. In the All Available Conga Queries area, click the query to select it. The report name and ID appear in the boxes to the right.
4. In the Alias field, type an alias for the report.
5. (Optional) In the Description field, type a text description of the report.
6. (Optional) To modify the default filter of the query, click **Advanced Filter Configuration**.
7. Click **Add Conga Query to Solution Record**. Solution Manager adds the query to the record. The query appears in the Selected Queries area.
8. Click **Return to Solution Record**.
9. Click **Update Button** to add your changes to the current button URL.

 The Conga Query record must contain a valid SOQL statement so it can appear in the All Available Conga Queries area.

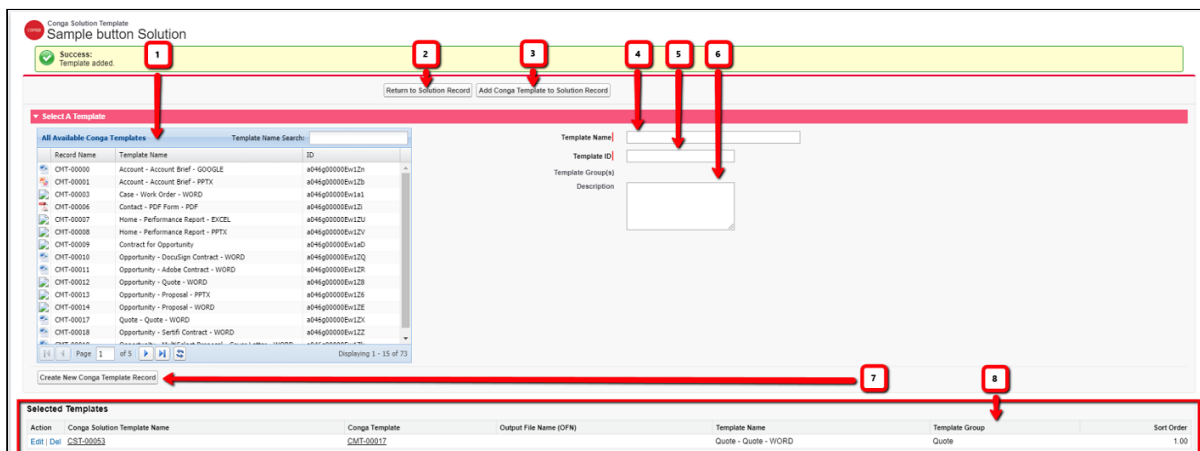
## Adding File Templates

In Conga Composer, you create file templates that auto-populate with Salesforce® data when running the solution. You must configure these templates with the correct merge fields for them to populate correctly.

For more in-depth information on templates, refer to the template guides for each template type: Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Adobe PDF Forms, and HTML email.

 You can add a maximum of 10 Conga Templates using the Conga Solution Manager. To add and merge more than 10 Conga Templates in a Conga Solution, contact your Customer Success Manager and then add the Conga Templates to your Composer Solution manually in Salesforce Setup.

## About the File Templates Screen



1	A list of all templates available to add to your solution.
2	Click <b>Return to Solution Record</b> to close the Add Templates screen and return to the solution record.
3	Click <b>Add Conga Template to Solution Record</b> to add a selected template to the solution.
4	Once you select a template, the Template Name field auto-populates.
5	Once you select a template, the Template ID field auto-populates.
6	If the TemplateGroup parameter has been added to this Solution, the values of the TemplateGroup are shown here and only Conga Templates that match those values will appear in the table.  The Template Group parameter must be added to your button for this functionality to work.
7	Click <b>Create New Conga Template Record</b> to open the Conga Template Manager in a new window.
8	All templates that have been added to the solution display under Selected Templates.


## Add an Existing Template to a Solution using Solution Manager

You can add new or existing templates to your solution. If the template exists, you need only select it and assign an alias to the template. If the template you need doesn't exist, you can create one within Solution Manager.

To add an existing template to a solution using Solution Manager:

 **Before you begin, you must have a solution record created and open.**


1. In Solution Manager, click **Add Templates**. The Conga Solution template screen appears. All available templates appear in the list.
2. Click a template to select it. The template name and ID appear in the boxes to the right.
3. (Optional) In the Description field, type a text description of the template.

 You can add only 10 templates through Conga Solution Manager UI. If you want to add more than 10 templates, you must use the [Manual Button](#) creation and editing option.

4. Click **Add Conga Template to Solution Record**. Solution Manager adds the template to the record. The template appears in the Selected Templates area.
5. Click **Return to Solution Record**.
6. Click the **Update Button** to add your changes to the current button URL.

## Add a New Template to a Solution using Solution Manager

How to add a new template to a solution record in Solution Manager.

 **If the template you want to use for the solution is not yet saved in Salesforce, you can add it in Solution Manager. You must have a solution record created and open. Additionally, you should have a template file already created.**

To add a new template to a solution record in Solution Manager:

1. In Solution Manager, click **Add Templates**. The Conga Solution Template screen appears.
2. Click **Create New Conga Template Record**. The Conga Template Edit screen appears.
3. In the Name field, type a name for the template.

4. (Optional) If the template is part of a group, in the Template Group box, type the name of the group.
5. In the Template Type box, select the type.
6. (Optional) In the Master Field to Set boxes, enter the master field, an equals sign, and the value for each of the master fields you want to set.
7. (Optional) Select the Label Template: Use Detail Data box to use the report or query data as a data source for the template label. This option applies only if the Template Type field is set to Label.
8. (Optional) In the Description box, type a description for the template.
9. Click **Save**. The Conga Template screen now appears. You can now attach a template to this template record. If you do not already have a template designed, [Create a Template with the Template Builder](#).
10. Click the **Related tab** on the new Conga Template record.
11. Click **Upload Files** and select a file from your machine.
12. Click **Done** once the file is uploaded.
13. Navigate back to the Conga Solution Template screen.
14. Click the refresh icon in the All Available Conga Templates box.
15. Select the new Conga Template record and click **Add Conga Template to Solution Record**.

## Adding Email Templates

With Conga Email, you create a template record in the new custom object included when you install Conga from the AppExchange. Then you build a merge template in the record's rich-text field using the Conga Advanced Editor.

Name	CET Name	ID
CET-00000	Opportunity - Receipt	a026g0000063JF9
CET-00001	Opportunity - Proposal	a026g0000063JF5
CET-00002	Quote - Quote	a026g0000063JF6
CET-00003	Opportunity - Invoice	a026g0000063JF8
CET-00004	Opportunity - Contract - Standard	a026g0000063JF0
CET-00005	QuickMerge - Dashboard Report	a026g0000063JFP
CET-00006	Opportunity - Contract - Priority Shipping	a026g0000063JFQ
CET-00007	Account - Account Brief (Classic)	a026g0000063JFS
CET-00008	Account - Account Brief (Lightning)	a026g0000063JFT
CET-00009	Follow up Lead Email	a026g0000063JFR
CET-00010	Contact - Change Request	a026g0000063JFG
CET-00012	Order - Invoice	a026g0000063JFA
CET-00013	Opportunity - Enhanced Proposal	a026g0000063JFV
CET-00014	WetSignature Contract Email Template	a026g0000063JFH


- 1 The Template Group parameter must be added to your button for this functionality to work.

2	Click <b>Return to Solution Record</b> to close the Add Templates screen and return to the solution record.
3	Click <b>Add Conga Email Template to Solution Record</b> to add a selected email template to the solution.
4	Once you select a template, the Template Name and Template ID fields auto-populate. Optionally, you can enter a template output file name (Template OFN).
5	If the TemplateGroup parameter has been added to this Solution, the values of the TemplateGroup are shown here and only Conga Templates that match those values will appear in the table. The Template Group parameter must be added to your button for this functionality to work.
6	You can add an optional Description for your email template.
7	Click <b>Create New Conga Email Template Record</b> to open the Conga Email Template Manager in a new window.

## Add a New Conga Email Template to a Solution using Solution Manager

If a solution created using Solution Manager requires a new HTML email template, you can create one within Solution Manager.

To create a new Conga email template using Solution Manager:

 Before you begin, you must have a solution record created and open.

1. On the Conga Solution screen, click **Add Conga Email Template**. The Conga Solution Email Template screen appears.
2. On the Conga Solution Email Template screen, click **Create New Conga Email Template Record**. The Conga Email Template Edit screen opens in a new window.
3. In the Name field, enter a name for the email template.
4. (Optional) In the Template Group field, enter the name of the template group containing this template.
5. (Optional) In the Description field, enter a description of the template.
6. (Optional) If you want to store the body of the template as an attachment rather than a rich-text field, select the Use attachment for template body box.



7. In the Email Content area, in the Subject field, enter a text string that will become the subject line of the merged email. You can include dynamic content in the Subject field. For example, Invoice for `{{OPPORTUNITY_NAME}}`.
8. In the HTML Body field, enter the contents of the body of the email. You can use text formatting and attach images and files by using the text formatting tools in the HTML Body toolbar OR in the Text Body field, type body text without using rich-text editing tools.
9. Click **Save** to save the template.
10. Close the window to return to the Conga Solution Email Template screen.
11. Select the template from the All Available Conga Email Templates table and click Add Conga Email Template to Solution Record to add the template to the solution.
12. (Optional) Click **Update Button** to add your changes to the current button URL.

## Add an Existing Conga Email Template to a Solution using Solution Manager

You can add email templates to the Conga Template Manager from within Solution Manager.

To add a Conga email template:


 **Before you begin, you must be editing a solution record.**

1. On the Conga Solutions page, click **Add Conga Email Template**.
2. Select the Conga Email Template you want to add to your solution. You can only have one Conga Email Template per solution.
3. (Optional) In the Description field, add a free-form text description.
4. Click **Add Conga Email Template** to select the desired template and add it to the solution.
5. (Optional) Click **Save** to save your solution record settings or If the button has been created, click **Update Button** to update the button URL with the current settings.

## Select a Sample Record in Solution Manager

One of the features of Solution Manager is the ability to create and test a solution before adding the button or link to a page. In Solution Manager, you can select a sample record from your dataset to build a template or preview a merged document.

To select a sample record in Solution Manager:

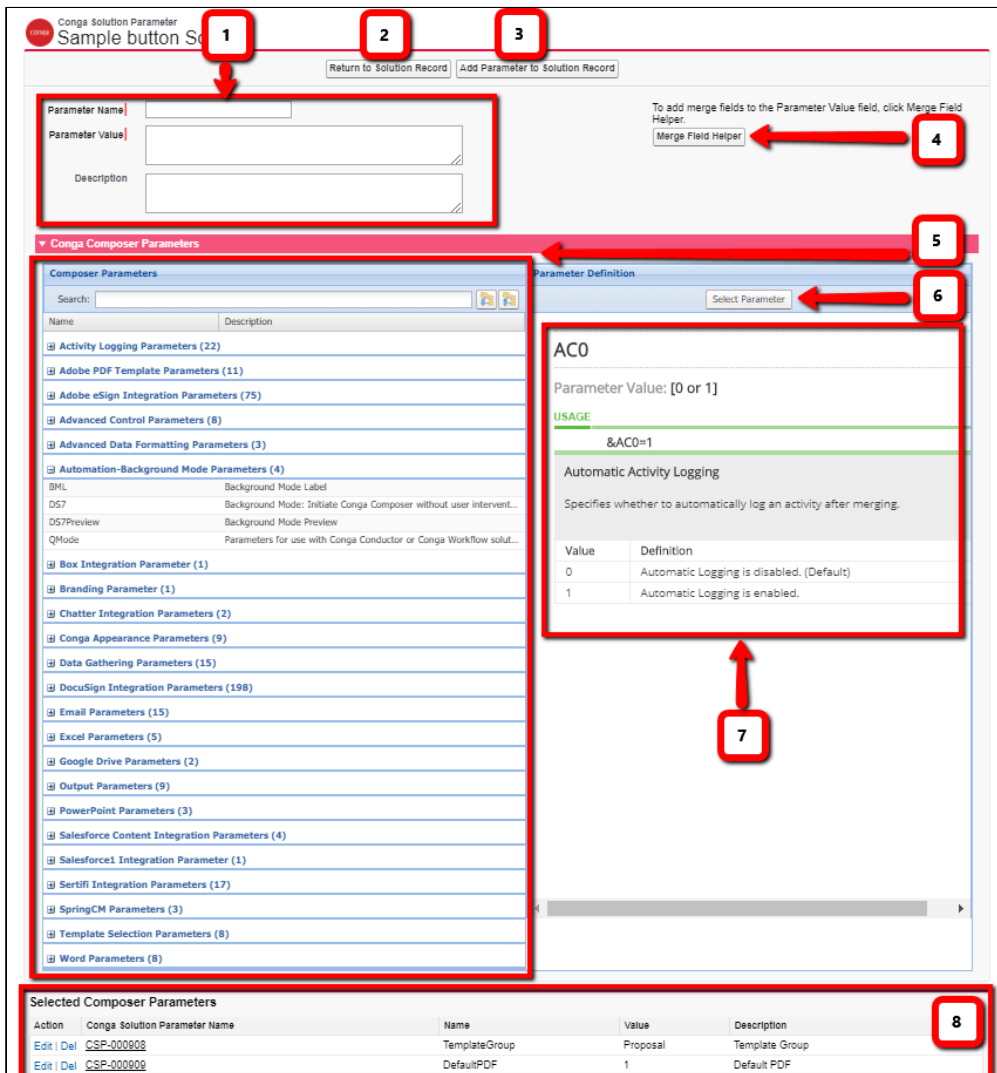
 Before you begin, you must have a solution record created and open. In addition, you must have created the button.

1. On the Solution Manager screen, click **Access Template Builder**. The Conga Solution Sample Record screen appears, showing a list of sample records.
2. In the Sample Opportunity Records box, click a record to select it. Solution Manager updates the Sample Record ID field.
3. Click **Select Sample Record**.
4. Click **Return to Solution Record** to return to the solution record or click **Access Template Builder** to open the Conga Composer for the selected record to preview the merged document or create a new template.

## The Conga Solution Parameter Screen

Parameters allow users to customize the behavior of their solutions.

## The Conga Solution Parameter Screen



1	Parameter name, value, and description. The name field (and in some cases, the value field) auto-fills after you select a parameter. A value is required; description is optional.
2	Returns you to the solution record. (Click this after you have added the parameters you want.)
3	Adds the parameter/value pair to the solution's button URL. (Click this after each parameter you add.)
4	Displays a list of fields from the master object so that you can use a dynamic value for the parameter.

5	<p>Displays a list of Conga Composer® parameters, organized by category.</p> <ul style="list-style-type: none"> <li>• Use the search field to quickly locate a parameter by name, function, or keyword.</li> <li>• Click the plus sign to expand the list of parameters within a category. (Click again to collapse the list.)</li> <li>• Click a parameter to view details in the right pane.</li> </ul>
6	<p>Click to select a parameter and auto-fill the parameter name field (and a default value, if applicable).</p>
7	<p>Displays the parameter description.</p> <p>Auto-populates after a parameter is selected. You can also type additional information and notes here; after the parameter is added to the solution, this information is available in the Selected Parameters area.</p> <p>Description information is not added to the URL.</p>
8	<p>The Selected Composer Parameters area displays a list of parameters that have been added to the solution.</p>

## Add a Parameter to a Solution using Solution Manager

 You must have a solution record created and open.

You can add multiple parameters to a solution within Solution Manager. The Parameter Picker within Solution Manager presents all available parameters and provides detailed information about the accepted parameter values and each parameter's functionality.

To add parameters to a solution:


1. Click **Customize with Parameters** to load the Conga Composer Parameter screen. For information on this screen, see [Adding Parameters](#). The parameter list is divided out into sections based on functionality. Click the plus sign to expand a section and view the parameters in that category. Click again to hide the parameters in the category.
2. Click a parameter to see information and settings for the parameter. If the parameter you want does not appear, see [Manually Adding a Parameter with Solution Manager](#). The description pane lists parameters and other settings that are required to use the selected parameter.
3. Click **Select Parameter** to select the parameter. The Solution Manager Parameter Picker auto-fills the Parameter Name field.

4. Enter a value in the Parameter Value field. The Parameter Picker will enter a default value of 1 automatically for parameters with On/Off functionality. This is a required field.
5. (Optional) To add a merge field to your parameter, click **Merge Field Helper** to view the available merge fields.
6. Click **Add Parameter to Solution Record** to save the selected parameter to your solution.
7. Repeat to add additional parameters.
8. Click **Return to Solution Record** to return to the solution record where you can update the button URL.

## View Solution and Button Archive Components

You can view solution components as well as archived button URLs and notes for a solution in Solution Manager.

 **Conga does not recommend deleting buttons from the archive.**

 Before you begin, you must have a solution record created and open.


To view solution components and button archive entries, on the Solution Record screen, scroll to the Solution Components or Button Archive area.

You can edit or remove components from the solution from this screen.

## Add a Note to the Button Archive using Solution Manager

Solution Manager automatically saves button URLs whenever you update a button. You can also add a note to the archive, for example to document user requests or annotate why you made changes.

To add a note to a button archive:

 Before you begin, you must have an existing solution record open in Solution Manager.

1. In the Button Archive area of the solution record, click **New Note**. The Note Edit screen appears.
2. (Optional) To make the note private, select the Private check box.
3. In the Title box, type a brief title for the note.
4. In the Body box, type the contents of your note.

5. (Optional) Click Check Spelling to check the spelling of the note text. The spell-checker opens in a new pop-up window. Follow the onscreen instructions to find and correct misspellings.
6. Click **Save**.

The Conga Solution screen appears, with the new note in the Button Archive area.

## Remove Solution Components in Solution Manager


You can easily edit or remove solution components that you have added to a solution record.

To remove solution components from a solution record:

 **Before you begin, you must have a solution record created and open.**

1. On the Solution Record screen, scroll to the Solution Components area.
2. In the row containing the component you want to remove, click **Del**.
3. In the confirmation dialog box, click **OK**.

## Add or Remove Buttons on Page Layouts

 **You must have a solution record open with a button created.**

You can add an existing Solution Manager button to a page layout. You can also remove buttons from page layouts.

To add or remove a Solution Manager button on a page layout:

1. In Solution Manager, click **Manage Page Layouts**.
2. In the Available Page Layouts area, select a page layout to add the solution button to it.
  - a. You can add buttons to more than one page layout. Click **Select All** to add the button to all available layouts.
3. Click **Add to Selected Layout(s)** to add the selected buttons to the layout or **Click Remove from Selected Layout(s)** to remove the buttons from the layout. A confirmation screen appears.
4. (Optional) Click **Return to Solution Record**.

## Run a Solution from Solution Manager

Once you have created a solution, you can test it using a sample record to make sure it works properly before adding it to a page layout.

To test a solution:

1. Click **Select Sample Record**.
2. Click the record you want to use as the sample record for testing. This will auto-populate the Sample Record ID field.
3. Click **Submit & Return to Solution Record**. This will add the sample record to your solution and allow you to test your solution against this record. The Launch button is now enabled in the Button Syntax & Test Launcher area.
4. Click **Launch** to launch Conga Composer.

You can now run a merge, check your settings, open the Template Builder, and access any other parts of Composer necessary to adequately test your solution.

## Remove a Button from a Page Layout in Solution Manager

You can easily remove buttons from page layouts in Solution Manager.

To remove a button from a page layout:

1. Click **Manage Page Layout(s)**.
2. Select the layouts where you want to remove the button.
3. Click **Remove From Selected**. The button has now been removed from the desired page layout.

You can add the button to the layout again by clicking Manage Page Layout(s) and following the instructions on adding a button to a page layout above.

## Best Practice for Editing a Button in Solution Manager

- When you need to make changes to the button or add to or remove components from your solution, use the Update Button to save those changes.
- As you are making updates and changes, you can view the solution as it is being built in the Button Syntax & Test Launcher section by looking at the Button Configuration in Progress area.
- Whenever you update a button, a record of the previous button is added to the Button/ Formula Archive section. This section retains a history of all button URLs changed in Solution Manager.
- Solution Manager is not bi-directional. While the Current Button Configuration in Solution Manager will always display the actual button configuration, any changes


made directly in button URL will not push back into Solution Manager. As soon as you click Update Button in Solution Manager this will override any changes made directly to the button URL.

- We recommend as a best practice that the Salesforce URL for the button or link be copy/pasted into the Description field of the Solution Manager record.
- If changes have been made directly to the button URL in Buttons, Links and Actions, an informative message appears when the record is accessed in Solution Manager. This message serves as an indicator that the button URL has changed outside of the Solution Manager. This message disappears after the button has been updated in Solution Manager.

## Manually Adding a Parameter with Solution Manager

If you need to add a parameter to your solution that does not appear on the Customize with Parameters screen in Solution Manager, follow these steps:

1. From your Solution Record, click **Customize** with Parameters.
2. Type the name of the parameter into the Parameter Name field. For example, to use the Live Edit feature you would type LiveEditVisible.


 Do not include an ampersand (&) because Solution Manager will add it for you.

3. Type the parameter value into the Parameter Value field. For example, to turn on the LiveEditVisible parameter, you would enter 1.
4. Click **Add Parameter to Solution Record**.
5. Repeat for each parameter you want to add that does not appear on the Customize with Parameters screen.

## Remove a Template Entry in Solution Manager

You can remove a template from your solution record in Solution Manager. Removing the template does not delete it from the Conga Template Manager or Salesforce.

To remove an existing template:

 Before you begin, you must have a solution record created and open. In addition, the solution must have at least one template.

1. In Solution Manager, click **Add Templates**. The Conga Solution Template screen appears. Templates added to the solution record appear in the Selected Templates area.



2. In the Selected Templates area, click **Del** in the row containing the template you want to remove.
3. In the confirmation dialog box, click **OK**. Solution Manager removes the template from the solution record.
4. Click **Return to Solution Record**.
5. Click **Update Button**.

## Global Merge for Composer

Information about the Global Merge feature in Solution Manager for Conga Composer.

- [About Conga Global Merge for Solution Manager](#)
- [Add a Global Merge Record using Solution Manager](#)
- [How to Create a Conga Global Merge Solution](#)
- [Set up a Global Merge Record in Salesforce Professional Edition](#)
- [Automate a Global Merge Solution with Conga Batch](#)
- [Converting Conga Courier Solutions to Global Merge Solutions Scheduled with Conga Batch](#)

## About Conga Global Merge for Solution Manager

Solution Manager includes the Global Merge feature. Global Merge allows you to gather data from multiple objects or records.

The features formerly included in the Conga QuickMerge optional add-on have been integrated directly into Conga Composer 8.8 and higher.

Conga Global Merge records are simple to create and can be managed like any solution using Solution Manager. Global Merge solutions differ from other Conga Composer solutions in that they are not attached to a standard Salesforce object, but are attached to the Conga Global Merge object. You can create fields within the record Button URL and those fields become Master fields on the Global Merge record. Master fields are available on your master data set.

You launch Conga Global Merge solutions from the Conga Global Merges tab.

To allow end users to launch Global Merge solutions, you must grant access to the Global Merge Visualforce page at the Profile level. Go to **Setup** → **Develop** → **Visualforce Pages** → **Composer Solution Launcher** → **Security** → **Add Profile**.

## Add a Global Merge Record using Solution Manager

Solution Manager has the built-in functionality to create a Global Merge record and will be automatically added to the Conga Global Merge Records object.

To create a Global Merge solution record in Solution Manager:

1. Click the **Conga Solutions tab**.
2. Click **Create New** and select **Conga Solution**.
3. In the Conga Solution Name field, enter a name for the solution.
4. In the Master Object field, select **Conga Global Merge (APTXConga4\_Composer\_QuickMerge\_c)**.
5. Click **Create Record**. The new solution opens in the Conga Solution screen.
6. Add solution components, such as reports, queries, templates, and parameters to your Solution Manager record. Because the Master Object is a Global Merge custom object, you must specify the datasets that populate the templates for your solution.
7. Click **Create Conga Global Merge** to move the Button URL from the in Progress area to the Current Button URL area.


You can test, revise, and update the solution record using the buttons in the Button URLs section. For more information on adding components to the solution, see [Creating and Managing Solutions using Solution Manager](#).

When Solution Manager creates the Global Merge record, it also creates a record ID and associates the button with the Global Merge record. You can launch the solution directly from the Conga Solution screen by clicking Launch Composer or the Global Merge object. Global Merge solutions appear on the Conga Global Merges tab.

## How to Create a Conga Global Merge Solution

Create an org-wide reporting solution with Conga Global Merge.

Global Merge is a feature included with Composer version 8.7 and newer. Global merge provides a custom object from which to launch Composer solutions that perform record-independent merges (such as org-wide or dashboard reporting). The steps to create a Global Merge solution are listed below.

 **Master Field To Set (MFTS) does not work with Global Merge Solutions.**

### To create a new Conga Global Merge Solution

1. Click the **Conga Solutions tab** and click **New**.

2. In the Conga Solution Name field, enter a name for the solution.
3. In the Master Object field, select Conga Global Merge (APTXConga4\_Composer\_QuickMerge\_c).
4. Click **Create Record**.
5. Click **Create Conga Global Merge** to move the Button URL from the in Progress area to the Current Button URL area. This action will create a Global Merge record that is linked to the Solution Manager record.
6. Create the Salesforce reports or SOQL queries needed for your report.
7. Select **Add Report Data** or **Add Query Data**.
8. Select the Report or Query from the list.

 Queries are highly recommended in Lightning.

9. Add an alias.
10. Click **Advanced Filter Configuration** and set the pv0 value to + (this will ensure the filters you set for your report or query are retained).
11. Click **Add Report or Query to Solution Record**.
12. Repeat Steps 7 - 11 for all the reports or queries you want to add to the Global Merge Solution.
13. Return to the Solution Record and click **Update Conga Global Merge**. This will add the reports or queries to your Global Merge solution.
14. The next step is to create the Conga template. Access the Template Builder by clicking Launch Composer.
15. Click **Tools and Settings** and select **Template Builder** from the drop-down.
16. The report and query data are listed in the Template Builder (look for the alias name you used).
17. Begin creating your template by adding merge fields from the Template Builder.
18. (Optional) To test your template, click **Back** in the Template Builder interface then select the Local Template tab from the Composer User Interface. Select your template (saved locally on your computer) and click Preview to merge your report and query data into the template.
19. When satisfied with your template, upload it to Salesforce by clicking **Add Templates** from the Global Merge solution record.
20. Click **Create New Template Record**, click **New**, and upload your template. For help, see [Add Template to Conga Template Manager](#).
21. Return to the Add Templates screen and refresh the table. Your newly uploaded template will appear. Select it and click **Add Template to Solution Record**. For more information, see [Add a Template Using Solution Manager](#)
22. The final step is to customize your Global Merge solution with the Conga parameters. See [Adding Parameters](#) for more information. For a list of parameters available to use, see the [Composer Parameter Guide](#).

23. When you are done adding parameters, click **Update Conga Global Merge** from the Solution record.
24. To run the finished solution as an end-user, click on the Conga Global Merge Record ID to navigate to the Global Merge record that was created from the Solution record.
25. From the Global Merge record click **Launch**. This will launch the Conga Global Merge solution and generate the output file using the selected template.

## Set up a Global Merge Record in Salesforce Professional Edition

In Professional Edition, the process for setting up a Global Merge record is slightly different than the basic method described earlier, and requires a few additional steps.

To set up a Global Merge Record in Salesforce Professional Edition:

1. Click the **Conga Solutions tab**.
2. Click **Create New** and select **Conga Solution**.
3. In the Conga Solution Name field, enter a name for the solution.
4. In the Master Object field, select **Conga Global Merge (APTXConga4\_Composer\_QuickMerge\_c)**.
5. Click **Create Record**. The new solution opens in the Conga Solution screen.
6. Add solution components, such as reports, queries, templates, and parameters to your Solution Manager record.
7. Click **Create Conga Global Merge** to create the Conga Global Merge object. When Solution Manager creates the Global Merge record, it also creates a record ID and associates the button with the Global Merge record. Global Merge solutions appear on the Conga Global Merges tab.
8. Copy the URL that has been created.
9. Navigate to the Global Merge object: **Setup** → **Create** → **Objects** → **Buttons, Links and Actions** → **New Button or Link**.
10. Create a new button using the URL that you copied from the Solution Manager.
11. Save and then copy the Global Merge solution ID from the URL into the clipboard.
12. Navigate to the Conga Global Merge Records tab under All Tabs. Your Global Merge record will display. However, in Professional Edition you cannot launch a button without completing a few additional steps.
13. Click the Global Merge record to access the details.
14. Click **Edit Layout**. Notice the Button ID and Button Name fields are duplicates of those already on the page.
15. Add each button to the Weblink Details area, even though they are already there. The buttons already in that section are not editable so you are unable to paste the

Record ID. Best Practice is to remove the un-editable button name and button id fields from the page layout to keep the page clean.

16. Once you have added the editable buttons, click **Save**.
17. Double-click the Button Name field and enter the button name.
18. Double-click the Button ID field and paste the Button ID copied earlier. The URL fields that already existed will now update with the information added in the editable fields.
19. Click **Launch** to launch the button.

## Automate a Global Merge Solution with Conga Batch

As with any other Conga Composer solution, a Conga Global Merge solution can be scheduled to run automatically using [Conga Batch](#), an optional, paid add-on to Conga Composer.

First, [create your Global Merge solution](#) and ensure it works just the way you want. Note the solution must be compatible with [Background Mode](#) (but not necessarily in Background Mode) in order to automate with Conga Batch. This means you must pre-select the merge template(s) as well as the desired delivery method (download, email, etc.).

### To automate a Global Merge Solution with Conga Batch

1. Install Conga Batch (if you have not already).
2. Navigate to the **Conga Batches** tab.
3. Click **New**.
4. Name your solution (we recommend using the same name as your Global Merge solution - i.e. Sales Pipeline Report or Quarterly Business Review).
5. Click **Save**.

6. Next, navigate to the Global Merge record you want to automate and copy the record id from the address bar (the last fifteen characters of the URL).

7. Paste the Global Merge record id into the Record ID field on the Conga Batch record you just created and click **Save**.

Conga Conductor-0001

Scheduled Conductor History

**Conga Conductor Detail**

Conductor Name: Conductor-0001 | Title: QBR | Description: | Owner: Ryan Burke (Change) | Last Modified By: Ryan Burke, 12/1/2016 10:28 AM

▼ Master Object Source

Report ID	URL Field Name
Query ID	Version: C8
<b>Record ID: a073600000219dK</b>	Conductor Environment: prod-

▼ Schedule

Schedule Description | Content Workspace ID | Next Run Date

**Scheduled Conductor History**

No records to display

Always show me more records per related list

8. Click **Formula Builder**.

Conga Conductor-0001

Scheduled Conductor History

**Conga Conductor Detail**

Conductor Name: Conductor-0001 | Title: QBR | Description: | Owner: Ryan Burke (Change) | Last Modified By: Ryan Burke, 12/1/2016 10:28 AM

▼ Master Object Source

Report ID	URL Field Name
Query ID	Version: C8
Record ID: a073600000219dK	Conductor Environment: prod-

▼ Schedule

Schedule Description | Content Workspace ID | Next Run Date

**Scheduled Conductor History**

No records to display

Always show me more records per related list

9. Conga Global Merge will populate as the Master Object. Select the desired Output Mode (this will append the corresponding QMode to the formula. field).

Return to Conductor Record

### Formula Builder

Progress: Select Object → Select Output Mode → Select Button → Review Formula → Enter Formula Name → Create Field

Master Object: Conga Global Merge | **Output Mode: Download**

**Button / Link**

Name: | Show API Name:  | Formula / URL: | Show Converted Formula:

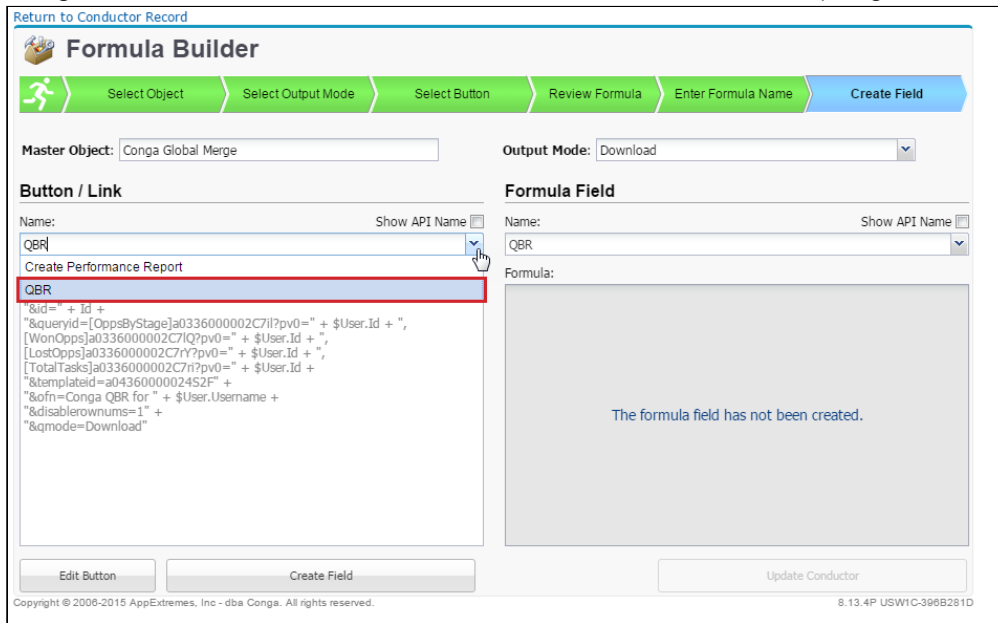
**Formula Field**

Name: | Show API Name:  | Formula: | The formula field has not been created.

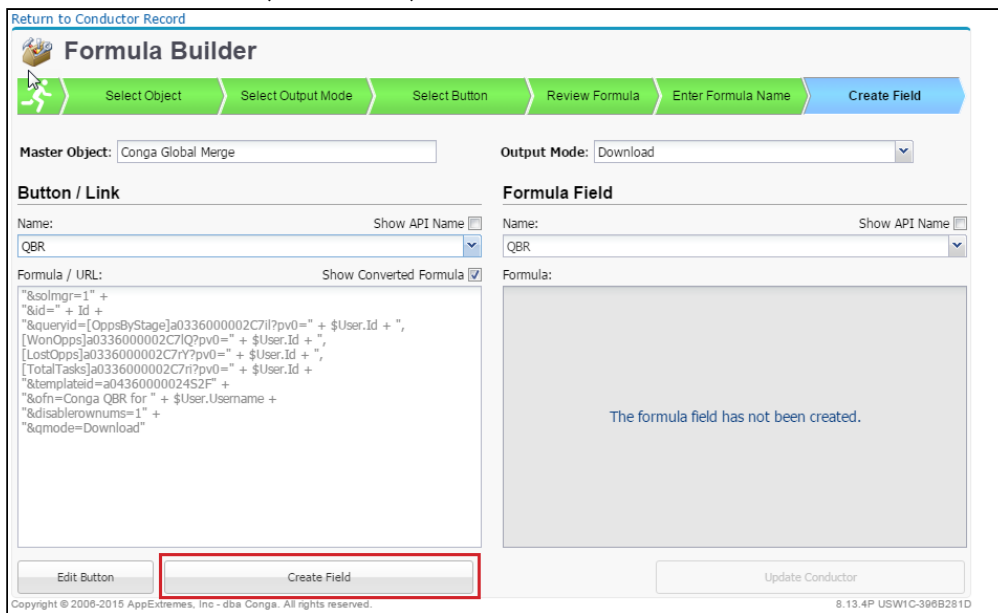
Buttons: Edit Button, Create Field, Update Conductor

Copyright © 2006-2015 AppExtremes, Inc. - dba Conga. All rights reserved. 8.13.4P USW1C-398B281D

- Next, select the button from the Name drop-down list (this is the name of the Global Merge solution - see below). The converted formula will display.

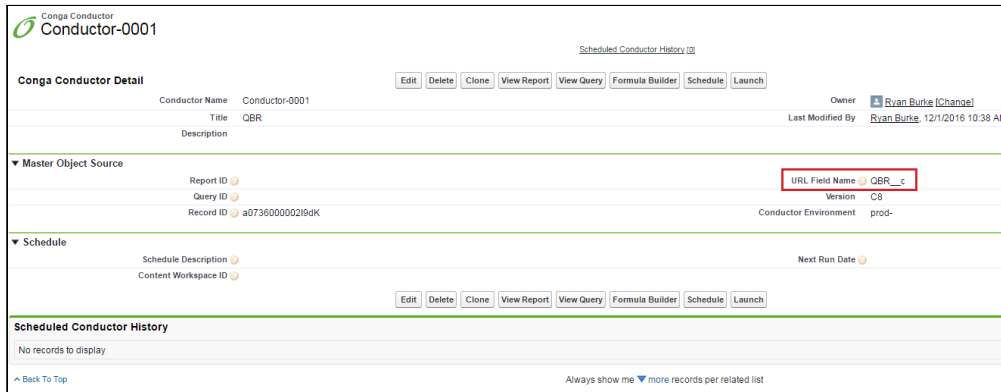


- Click **Create Field** (see below).

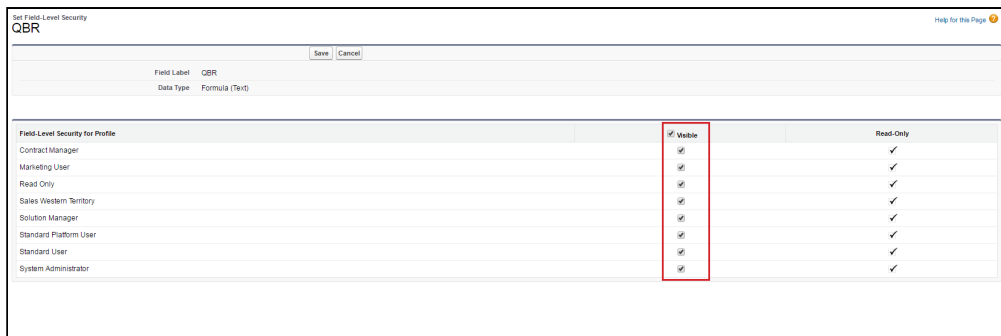


- A success message will display. Click **Yes** on the success message to update the Conga Batch record with the API Name of the newly created formula field (as in the

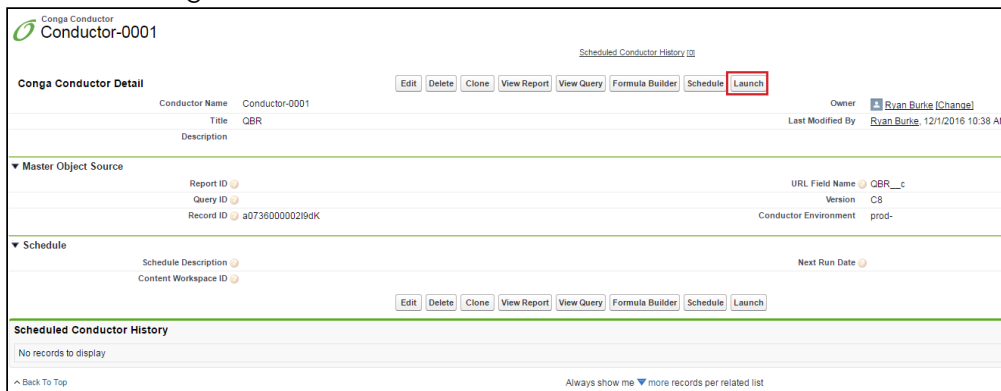
screenshot below).



13. Click **OK**.
14. Lastly, ensure the formula field is visible to you and any other user who might run this Conga Batch solution. To do so, return to the Formula Builder and click Edit Field Level Security.
15. Check the Visible box next to all of the Profiles of users who need to run the Conga Batch solution (below).

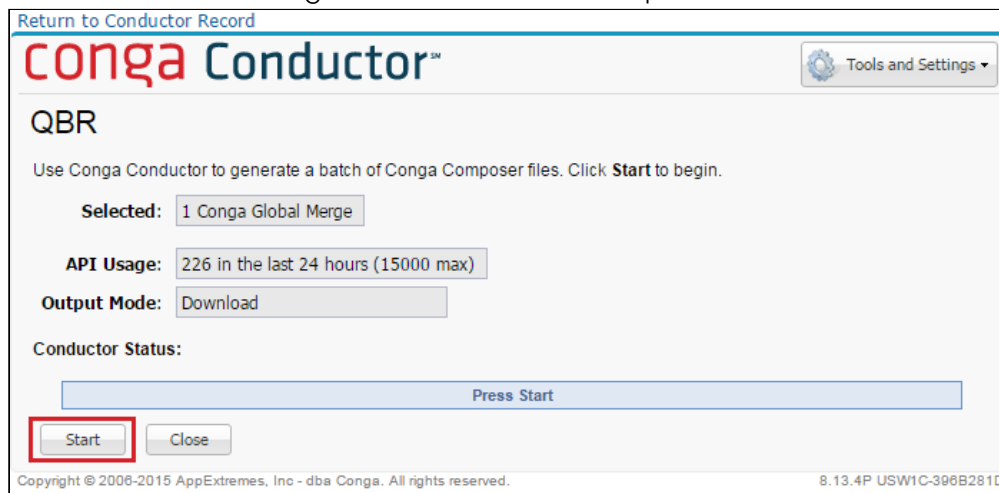


16. Click **Save** and close the tab.
17. We recommend testing your solution before you schedule it. To do so, click **Launch** from the Conga Batch record.





- Click **Start** in the Conga Batch window that opens.



- Ensure your Global Merge solution runs and delivers the document(s) as expected. You can review the results and any error messages on the Conga Batch Dashboard tab.
- (Optional) The final step is to schedule the Conga Batch solution on whatever recurring intervals you prefer. To do so, follow the steps in [Schedule Conga Batch to Run Automatically](#).

That's it - you have automated a Global Merge solution using Conga Batch.

## Converting Conga Courier Solutions to Global Merge Solutions Scheduled with Conga Batch

Use a Global Merge solution with Conga Batch to schedule and send reports. Convert your Conga Courier Solutions to a Global Merge solution and schedule to send the output files automatically with Conga Batch.

Global Merge allows you to gather data from multiple objects or records and supports Word, Excel, PowerPoint, and PDF templates. Use Conga Batch to schedule your Global Merge solutions on a daily, weekly, or monthly basis.


### Prerequisites

- [Install Conga Composer](#) from the Salesforce AppExchange.
- [Install Conga Batch](#) from the Conga Support Site.
- [Assign Conga Composer user licenses](#).
- [Convert data gathering Salesforce Reports to Conga Queries](#).

### To convert Conga Courier solutions to scheduled Global Merge solutions

1. [Create a Conga Global Merge Solution](#).

2. Add your data gathering [Conga Query](#) to the Global Merge solution and [use the Template Builder](#) to create a template.

 It is recommended to use an Excel template for the Global Merge solution.

3. [Create a Conga Email Template](#) to use with the Global Merge solution.
4. Determine the recipients from the [Conga Courier schedule](#) and add their email addresses to the Global Merge solution by leveraging the [EmailAdditionalTo parameter](#).
5. Add the Conga Template and Conga Email Template to the Global Merge solution, along with any additional behavior parameters (such as DefaultPDF and activity logging parameters).
6. [Create a Conga Batch record](#) and use the Record Id from your Global Merge solution as the **Master Object Source** on the Conga Batch record.
7. [Convert your Global Merge solution to a formula field using the Formula Builder](#) and set the QMode to Send Email.
8. Locate the newly created formula field on the Global Merge custom object in Salesforce Setup.
9. Set the field level security of the formula field to visible for all user profiles.

 Skipping this step will result in an error when the Conga Batch runs.

10. Enable the [Conga Batch Master Switch](#) and [create the Salesforce token](#) in Conga Batch Setup.
11. Navigate back to the Conga Batch record and set its schedule.

## Solution Packs

Solution Packs are installable packages containing pre-built Composer solutions. Requires Solution Manager (Composer version 8.7 or newer).

Select one of the following topics for more information:

- [About Solution Packs](#)
- [Install a Solution Pack](#)
- [The Composer Solution Pack](#)
- [The DocuSign Solution Pack](#)
- [The Adobe eSign Solution Pack](#)
- [View a Solution Pack Solution](#)
- [Delete a Solution Pack Solution](#)

## About Solution Packs

Solution Packs are a great way to get a solution up and running in minutes. They are pre-built solutions that gather the necessary data (using reports and queries), output into a custom template (which you can customize with your logo, font choice, and so on), and have parameters applied to provide functionality depending on the solution you're running.

Simply install the Solution Pack, unpack the solutions, add your logo and styles to the included template, and then add the button to a page layout.

Or, use them as a training tool or to get a head start, then use [Solution Manager](#) to customize and expand the functionality of these solutions.


We currently offer the following solution packs:

- Composer
  - Account Brief (Account)
  - Invoice (Opportunity)
  - Proposal (Opportunity)
  - [Install](#)
- DocuSign
  - DocuSign Contract (Opportunity)
  - DocuSign Quote (Opportunity)
  - [Install](#)
- Adobe eSign
  - Adobe eSign Contract (Opportunity)
  - Adobe eSign Quote (Opportunity)
  - [Install](#)

## Install a Solution Pack

You can install a Solution Pack in an organization that has Conga Composer version 8.7 or above.

To install a Solution Pack:

 Before you begin, you must have Conga Composer Release 8.7 or newer and be logged into Salesforce.

1. Navigate to the installation link for the Solution Pack you want to install.
  - We currently offer the following solution packs:
    - Composer
      - Account Brief (Account)

- Invoice (Opportunity)
  - Proposal (Opportunity)
  - [Install](#)
  - DocuSign
    - DocuSign Contract (Opportunity)
    - DocuSign Quote (Opportunity)
    - [Install](#)
  - Adobe eSign
    - Adobe eSign Contract (Opportunity)
    - Adobe eSign Quote (Opportunity)
    - [Install](#)
2. On the Install Composer Solution Pack screen, select **Install for All Users** and click **Install**.
    - a. On the Install Composer Solution Pack screen, select **Install for All Users** and click **Install**. The installer installs the Solution Pack.
  3. On the Installation Complete screen, click **Unpack Composer Solutions** to unpack all solutions or clear check boxes for any solutions you do not want to install, and then click **Unpack Composer Solutions** to install the selected solutions.
  4. Click **Add Remote Site** to add a required remote site.
  5. After installing the remote site, close the tab.
  6. Click **Create Composer Buttons**.
  7. On the Installation Complete! screen, click **Done**.

## The Composer Solution Pack

To install a Composer Solution pack, see [installing solution packs](#).

### Account Brief

The Account Brief solution provides a way to quickly understand all of the activity related to any Account. It uses Salesforce Reports to gather data from an Account's related Contacts, Opportunities, Activities, and Cases into a Word® document that is converted to a PDF file upon merging. After merging, the file is available for immediate download.

Data	Fields from the Account, Contact, Opportunity, Activity, Case, and Primary Contact (&ReportID)
Template	Word template converted to PDF output (&DefaultPDF)

Features	<p>Output file name includes Account Name and today's date (&amp;OFN)</p> <p>Output file is automatically saved to the Account's Notes and Attachments upon merging (&amp;SC0, &amp;SC1)</p> <p>A completed Activity is logged against the Account upon merging (&amp;AC0, &amp;AC1)</p> <p>If an Account does not have either related Contacts, Opportunities, Activities, or Cases, the corresponding table(s) is not shown in the output file (TableHide)</p>
----------	--

## Invoice

The Invoice solution allows you to create and email a simple invoice to a prospect or customer from any Opportunity. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word document that is converted to a PDF file upon merging. The solution attaches the merged PDF file to a Conga® HTML Email, which also displays data from the Opportunity and its related Opportunity Line Items. The default email recipient is the Opportunity's Primary Contact, but the email and attachment can be sent directly to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, Opportunity Owner, Primary Contact, and Organization
Template	Word template converted to PDF output (&DefaultPDF), Conga HTML Email template
Features	<p>Output file name includes Opportunity Name and today's date (&amp;OFN)</p> <p>Output file is automatically saved to the Account's Notes and Attachments upon merging (&amp;SC0, &amp;SC1)</p>

## Proposal

The Proposal solution offers a way to create and email a basic proposal to a prospect or customer from any Opportunity. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word document that is converted to a PDF file upon merging. The solution attaches the merged PDF file to a Conga HTML Email, which also displays data from the Opportunity and its related Opportunity Line Items. The default email recipient is the Opportunity's Primary Contact, but the email and attachment can be sent directly to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, User, Primary Contact, and Organization
Template	Word template converted to PDF output (&DefaultPDF), Conga HTML Email template
Features	Output file name includes Opportunity Name and today's date (&OFN)  Output file is automatically saved to the Account's Notes and Attachments upon merging (&SC0, &SC1)

## The DocuSign Solution Pack

DocuSign for Salesforce must be installed and configured in your target environment. To install the DocuSign Solution pack into your production environment, click [Install the DocuSign Solution Pack](#). To install DocuSign into a Sandbox, click [Install DocuSign into a Sandbox](#).

## DocuSign Contract

The DocuSign Contract solution provides a way to send a simple contract to a single recipient for electronic signature. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word® document that is converted to a PDF file upon merging. The contract template includes sample payment terms and terms and conditions that can be easily replaced with language specific to your organization. The default DocuSign recipient is the Opportunity's Primary Contact, but the contract can be sent to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, User, Primary Contact, and Organization
Template	Word template converted to PDF output (&DefaultPDF)  Word template includes DocuSign automatic anchor tags for the recipient's signature and the date signed

Features	<p>Output file name includes Opportunity Name and today's date (&amp;OFN)</p> <p>DocuSign recipient is defaulted to Opportunity Primary Contact (&amp;DocuSignR1Id)</p> <p>DocuSign recipient type is set to 'Signer' (&amp;DocuSignR1Type)</p> <p>DocuSign recipient role is set to 'Signer 1' (&amp;DocuSignR1Role)</p> <p>DocuSign recipient routing order is set to '1' (&amp;DocuSignR1RoutingOrder)</p>
----------	---

## DocuSign Quote

The DocuSign Quote solution provides a way to send a basic quote to a single recipient for electronic signature. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word document that is converted to a PDF file upon merging. The default DocuSign recipient is the Opportunity's Primary Contact, but the quote can be sent to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, User, Primary Contact, and Organization
Template	<p>Word template converted to PDF output (&amp;DefaultPDF)</p> <p>Word template includes DocuSign automatic anchor tags for the recipient's signature and the date signed</p>
Features	<p>Output file name includes Opportunity Name and today's date (&amp;OFN)</p> <p>DocuSign recipient is defaulted to Opportunity Primary Contact (&amp;DocuSignR1Id)</p> <p>DocuSign recipient type is set to 'Signer' (&amp;DocuSignR1Type)</p> <p>DocuSign recipient role is set to 'Signer 1' (&amp;DocuSignR1Role)</p> <p>DocuSign recipient routing order is set to '1' (&amp;DocuSignR1RoutingOrder)</p>

## The Adobe eSign Solution Pack

[Install the Adobe eSign Solution Pack \(link\)](#)

Adobe Document Cloud eSign Services for Salesforce must be installed and configured in your target environment.

## Adobe Document Cloud eSign Contract

The Adobe Document Cloud eSign Contract solution provides a way to send a simple contract to a single recipient for electronic signature. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word document that is converted to a PDF file upon merging. The contract template includes sample payment terms and terms and conditions that can be easily replaced with language specific to your organization. The default recipient is the Opportunity's Primary Contact, but the contract can be sent to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, User, Primary Contact, and Organization
Template	Word template converted to PDF output (&DefaultPDF) Word template includes Adobe text tags for the recipient's signature and the date signed
Features	Output file name includes Opportunity Name and today's date (&OFN) Adobe eSign Services agreement name includes Opportunity Name (&ESAgreementName) Adobe eSign Services recipient is defaulted to Opportunity Primary Contact (&ESRecipient1) Adobe eSign Services agreement signature type is defaulted to 'e-Signature' (&ESSignatureType)

## Adobe Document Cloud eSign Quote

The Adobe Document Cloud eSign Quote solution provides a way to send a basic quote to a single recipient for electronic signature. It merges data from an Opportunity, as well as its related Opportunity Line Items, into a Word document that is converted to a PDF file upon merging. The default recipient is the Opportunity's Primary Contact, but the quote can be sent to anyone.

Data	Fields from the Opportunity, Account, Opportunity Line Item, User, Primary Contact, and Organization
------	--



Template	Word template converted to PDF output (&DefaultPDF) Word template includes Adobe text tags for the recipient's signature and the date signed
Features	Output file name includes Opportunity Name and today's date (&OFN) Adobe eSign Services agreement name includes Opportunity Name (&ESAgreementName) Adobe eSign Services recipient is defaulted to Opportunity Primary Contact (&ESRecipient1) Adobe eSign Services agreement signature type is defaulted to 'e-Signature' (&ESSignatureType)

## View a Solution Pack Solution

Solution Pack solutions always include the Solution Pack name in the solution name. This makes them easy to locate in the Recent Items list.

In addition, the solution record includes a description that identifies the solution as being created by Solution Pack. Because the Solution Pack is part of Solution Manager, the button URL also includes a parameter identifying it as a solution managed by Solution Manager.

In all other aspects, you edit and launch the solution as you would any other Solution Manager solution.

## Delete a Solution Pack Solution

You can delete a Composer Solution Pack solution. Deleting a solution deletes the button and removes it from any page layout where it was added. All underlying reports, queries, and templates are not removed unless deleted manually.

To delete a Composer Solution Pack solution:

 Before you begin you must have a Composer Solution Pack solution installed.

1. Open the solution record for the solution you want to delete.
2. Click **Delete**.
3. When the confirmation dialog box appears, click **OK**.

# Composer for Lightning

Information on configuring Conga Composer for Salesforce Lightning. Try the [Lightning Migration Assistant](#).

- [Configure Conga Composer for Lightning](#)
- [Lightning Readiness Items for Conga Composer](#)
- [Convert Non-OAuth Buttons to OAuth](#)
- [Convert Data Gathering Reports to Conga Queries](#)
- [Add the Query to the Composer Solution Using an Alias](#)
- [Update Affected Templates](#)
- [Other Considerations](#)

## Configure Conga Composer for Lightning

### Background

The Salesforce Lightning Experience combines the new Lightning Design System, Lightning App Builder and Lightning Components to enable anyone to quickly and easily create modern enterprise apps within Salesforce. The changes to the Salesforce interface will require customers to update their Conga Composer solutions when they switch to Lightning.


For more information about Lightning, we recommend reviewing the [Lightning Trail](#) on Salesforce Trailhead.

### How will Migrating to Lightning Affect My Conga Solutions?

Currently, Salesforce customers can switch between Salesforce Classic and the Salesforce Lightning Experience. When you decide your organization is ready to switch to Lightning, the following items will need to be updated so your Conga Composer solutions continue to work:

- If your Salesforce Org has Lightning enabled, converting Salesforce Reports to Conga Queries is not required, but it is highly recommended due to better data gathering performance, versatility, and stability.
- If you replace your Salesforce Reports with Conga Queries, review all your templates as changing reports to queries changes your merge field names.
- Non-OAuth buttons must be converted to OAuth buttons.
- Conga Composer must be configured as a Connected App.

For more information and guided steps on how to address the above items, see the [Lightning Migration Assistant](#).


 **Warning: JavaScript buttons do not work in Salesforce Lightning.**

## Helpful Links


- [\\$Api in the Visualforce Developer Guide](#)
- [Lightning: is direct API access on the roadmap?](#)

## Lightning Readiness Items for Conga Composer

The following items must be addressed to make your Conga Composer solutions Lightning Ready. Use the navigation in this window to walk through the steps.

Item	What to Do
Configure Conga Composer as a Connected Apps	<a href="#">Configure the Conga Composer Connected App</a>
Non-OAuth Conga buttons	<a href="#">Convert non-OAuth buttons to OAuth</a>
Reports used to gather data for Conga templates	Convert reports to queries. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Converting reports to queries is highly suggested but is not required.           </div>
Templates	Converting reports to queries results in new field naming convention; all templates should be reviewed
Home Page Composer links	Move to Global Merge record or use the Utility Builder to design a new layout.
JavaScript Buttons (such as Composer or Conga Batch List View Buttons)	Choose a different implementation technique or engage Conga Services to build a custom solution.
Composer User Permission Set	Assign the Composer User Permission Set to Conga Composer users.

Item	What to Do
Winter '20 Critical Update related permissions	Assign the necessary permissions or permission set related to the Salesforce Winter '20 Critical Update. This is required for orgs that have Composer versions preceding Composer 8.179.

 As each Composer implementation is unique, additional steps might be required to make your specific Composer solution Lightning ready.

## Convert Non-OAuth Buttons to OAuth

Composer buttons that do not use OAuth security (Open Authorization) must be converted to OAuth buttons in order to work in Salesforce Lightning.

### How Do I Know if My Composer Buttons Need to be Converted?

1. Navigate to the button in question and view the URL. If the button was created using Solution Manager in Conga Composer Release 8, you can view the URL by navigating to the Solution record in the Conga Solutions tab. If not, you can view the button URL in Salesforce Setup or by launching the button and generating a Debug URL from the Composer user interface.
2. If the first three lines of your URL look like the one below, it is a Non-OAuth button and needs to be converted.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
```

3. If the first line of the URL looks like the one below, the button does NOT need to be converted.

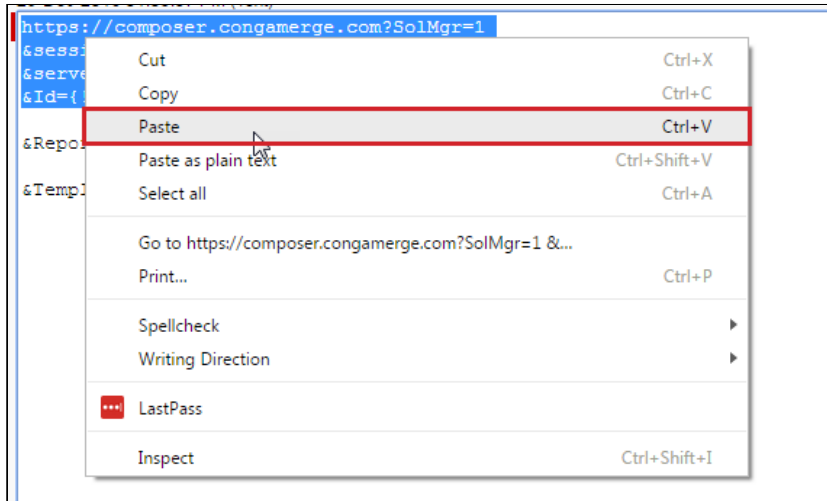
```
/apex/APXTConga4__Conga_Composer
```

### Convert to an OAuth Button

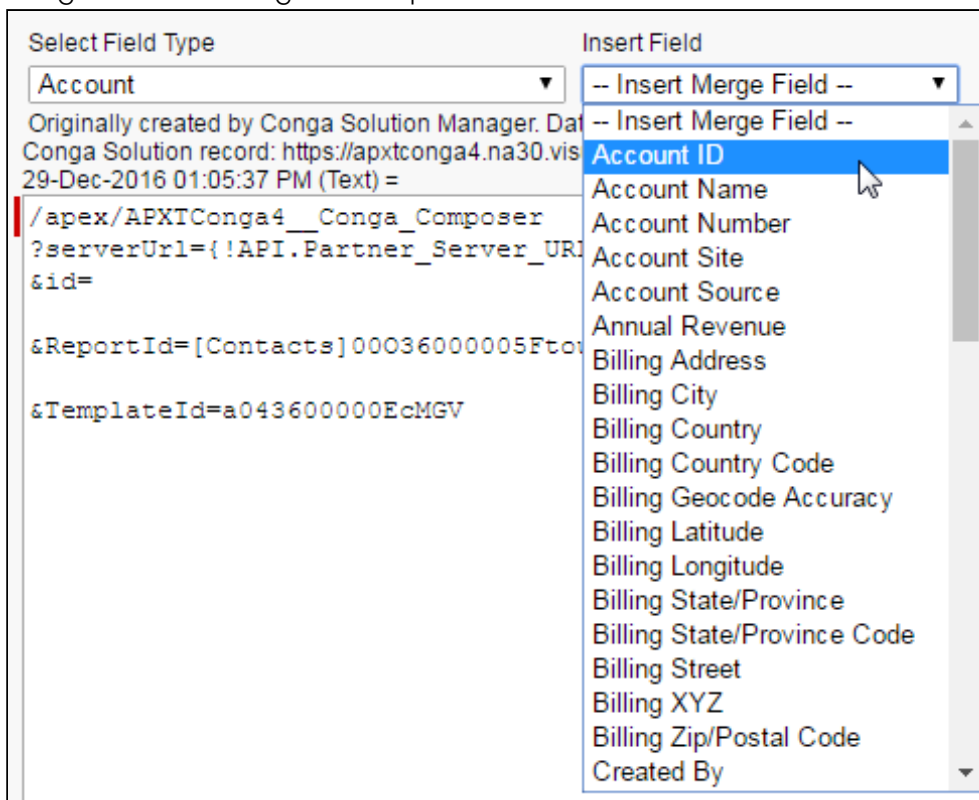
To convert your Non-OAuth button to an OAuth button, follow the steps below.

1. Copy the following URL: `/apex/APXTConga4__Conga_Composer?serverUrl={!API.Partner_Server_URL_370}&id=`

2. Navigate to your button in Salesforce **Setup** → **Customize** → Master Object (i.e. Account) → **Buttons, Links, and Actions**
3. Click **Edit**.
4. Replace the first four lines of your button URL with the three lines you just copied above. A quick way to do this is to highlight the first four lines of the URL, right click, and select **Paste**.



5. In the third line of the URL, enter the ID of your Master Object (i.e. the object on which the button was built, such as Account, Opportunity, Contact, etc.). You can do this using the Insert Merge Field option.



For a button built on the Account object, the first three lines of the URL would look like

this: /apex/APXTConga4\_\_Conga\_Composer?serverUrl={!API.Partner\_Server\_URL\_370}&id={!Account.Id}

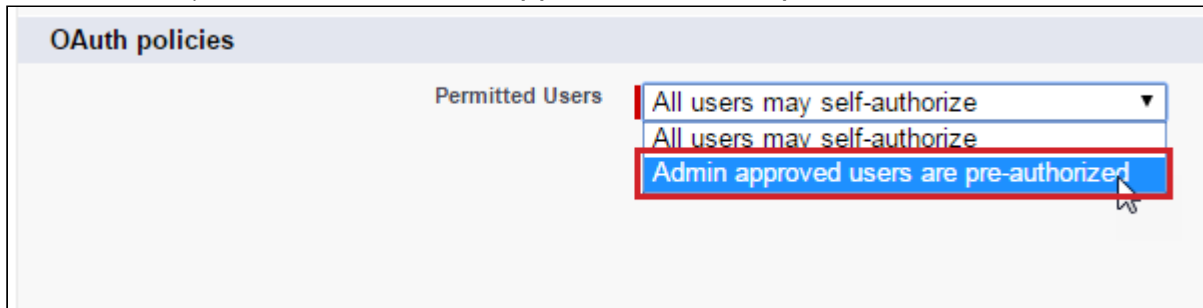
6. Click **Save** to save the updated URL.

## Configure Composer as a Connected App

Now you need to configure Composer as a Connected App to complete the process of converting to OAuth buttons.

Follow the steps below to do so.

1. Navigate to Connected Apps in Salesforce Setup.
2. Click **Edit** next to Conga Composer.
3. Under OAuth policies, select **Admin approved users are pre-authorized**.



4. Click **Save**. This will return you to the Connected Apps page.
5. Lastly, designate who can connect to Conga Composer via OAuth by adding Profiles or Permission Sets. To do so, click back into Conga Composer from the Connected Apps screen (by clicking Edit again), scroll down, and click Manage Profiles or Manage Permission Sets and then add the necessary Profiles or Permission Sets.

Users who have been pre-approved by the admin can now launch Composer OAuth buttons. Users who have not will be denied access.

**⚠ To use Composer OAuth buttons, users must have APXTConga4.Conga\_Composer Visualforce page added in their profile's Enabled Visualforce Page Access section. If this is not included, they will get an Insufficient Privileges error.**

## Convert Data Gathering Reports to Conga Queries

While Salesforce Reports are used to gather data for Conga templates, it is recommended that they be converted to Conga Queries to work in Salesforce Lightning. Conga templates that merge data from those reports must also be updated.

Converting reports to queries involves three steps:

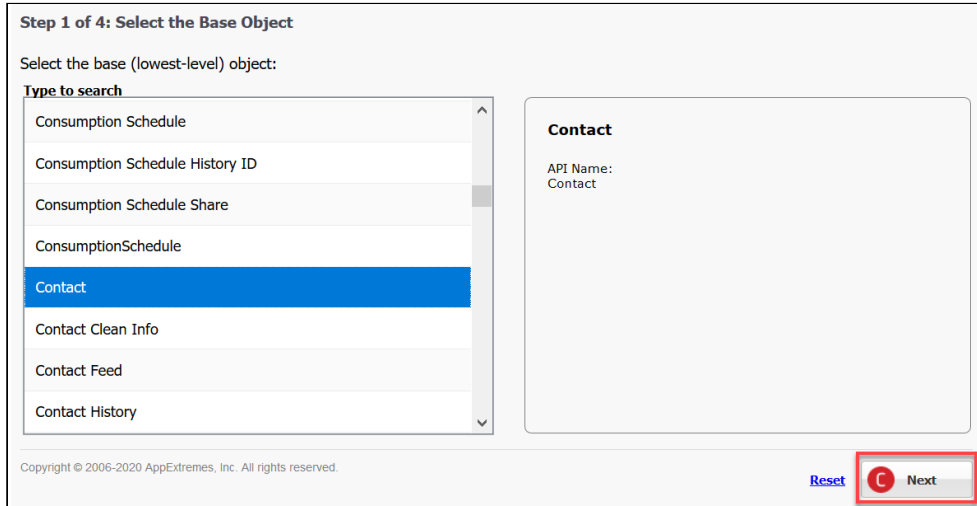
1. Use the Conga Query Builder to re-create the report.
2. Add the query to the solution using an alias.
3. Update affected templates

**i** In this example, we are going to convert a Salesforce report that gathers data about Contacts related to an Account for an Account Summary solution. In general, you can use these same steps to convert any reports you are using to gather data for the solution.

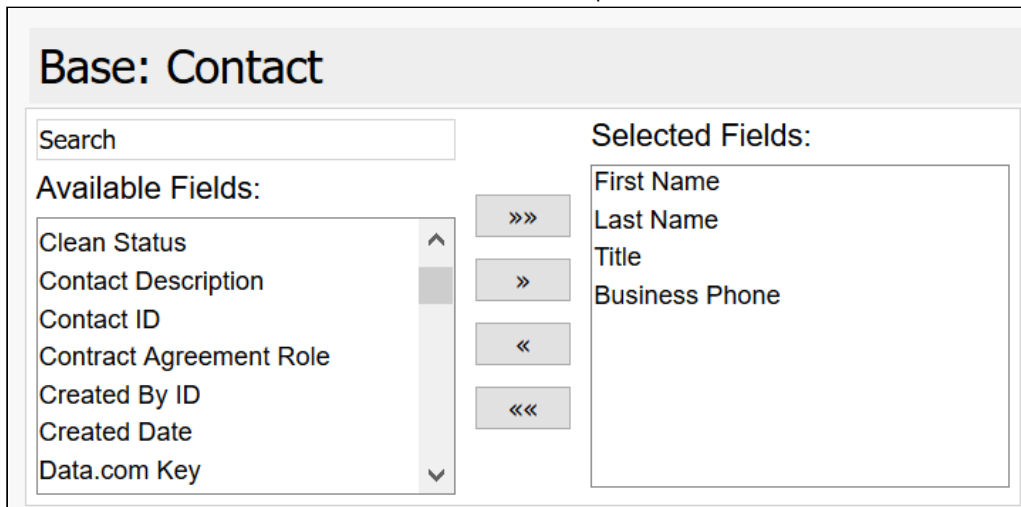
### Use the Conga Query Builder to Recreate the Report

The Conga Query Manager and Conga Query Builder are custom objects installed with the package. They assist with the creation of SOQL queries for use with . The steps below outline how to write a query using the Conga Query Builder.

1. Navigate to the Conga Queries object (go to All Tabs, which is the "+" in the ribbon, and then select Conga Queries from the list).
2. Click **New**.
3. Give the new query record a name. In our example, we are converting a report that gathers Contact data for an Account Summary solution, so we will name our query *Account Summary - [Contacts]*.
4. Click **Save**.
5. Next, click **Conga Query Builder**. The Conga Query Builder is going to walk you through four steps to write the query.
6. Select the base object. This is the lowest-level object. For our example, we are gathering Contact data related to an Account, so Contact is our base object. Select Contact and click **Next**.



7. Select columns. These are the fields you want to merge into your Conga template. In our example, the report we are replacing currently merges First Name, Last Name, Title, and Business Phone, so we are going to add those same four columns. Search for and select the columns from the list, then click the right-facing arrow to move each one over to the Selected Fields pane. Click **Next**.



8. Choose a selection criteria. This is the equivalent of adding a blank Master Object ID filter to your report. Choose your Master Object Id from the first field drop-down list. In our example, it is Account ID. As you do with reports, leave the Operator as equals and the value blank. Click **Next**.



Enter the selection criteria:

Field	Operator	Value	
Account ID	equals		AND
--None--	equals		AND
--None--	equals		AND
--None--	equals		AND
--None--	equals		

Choose the sort order:

Field	Direction
--None--	ASC
--None--	ASC
--None--	ASC

Number of rows to display:

**Back** **Next**

- (Optional) Test the results. If you would like to test the results of the query, you can paste a sample Master Object ID into the pv0 field. We will test our query later, so we will skip this step and instead click **Save**.

Notice your new SOQL statement populated in the SOQL Select Statement field on the Conga Query record.

## Add the Query to the Composer Solution Using an Alias

Now that the Conga Query has been created, you need to replace the ReportId parameter in the button URL with the QueryId parameter. Then you need to update the value of the QueryId parameter to the Id of the new Conga Query record.

Follow the steps below to complete these tasks.

- Navigate to the button in Salesforce Setup. **Setup** → **Customize** → Your Master Object (i.e. Account) → **Buttons, Links, and Actions**.
- Click **Edit**.
- Delete the ReportID parameter.
- Copy the QueryId parameter below and paste it into the URL where the ReportId parameter was.
  - `&QueryId=`
- Click **Quick Save**.
- In a separate tab, navigate to the Conga Query record you created in the earlier step and copy the last fifteen characters of the URL, which is the Query record Id.

7. Return to the tab with your button URL and paste the Id as the value of the QueryId parameter.
8. Add the alias, which is the name of the query dataset as it will appear in the Template Builder and be referenced in your Conga templates. For our example, we will call ours Contacts. Add the alias by typing the name between the "=" and the first character in the query Id, then surround the name with "[" and "]". Ensure there are no spaces.
9. Click **Save** to save the updated button URL.

## Test the Query

1. Navigate to a record with your Composer button you are working on. Click the button to launch the solution. For our example, we will navigate to an Account record and launch our Account Summary button.
2. From the Composer user interface that opens, click the Tools & Settings drop-down and select the Template Builder.
3. If the query was added correctly, you will see a new dataset named with your alias name. Expand the dataset to see the query data that Composer has retrieved. The fields available should be the same Columns you added to the query in the earlier step. In our example, we have First Name, Last Name, Title, and Phone, as well as some additional fields.

Congratulations! You have completed the process of converting your report to a query. The next step is to update affected templates by mapping these new merge fields to your template. Go to the next step.

## Update Affected Templates

After replacing reports with queries and adding them to the Composer solution using aliases, the final step is to update affected templates.

Reports and queries retrieve different values (reports retrieve Field Labels, while queries retrieve the API Name of the field along with the object prefix), which is why you must update any templates that contain merge fields from the reports you replaced with queries. This is as easy as copying the new merge fields from the Template Builder and pasting them into your template.

Follow the steps below to complete this task.

## Identify Affected Templates and Locate New Merge Fields

1. Identify which templates have been affected (i.e. those used with solutions that now use queries instead of reports). Then, identify which merge fields in the template need to be replaced (merge fields from the master dataset are not affected, but merge fields from detail datasets are).

**i** Merge fields from detail datasets are surrounded by `{{TableStart}}` and `{{TableEnd}}` merge fields. Look for these markers to help identify which merge fields need to be replaced.

For example, our Account Summary template has merge fields from the master dataset as well as the Contacts detail dataset, which is the data that came from the report we replaced with a query. These will need to be replaced.

Account Summary for: `{{ACCOUNT_NAME}}`

DATE: `{{TODAY \@ MM/DD/YY}}`  
 ADDRESS: `{{ACCOUNT_BILLINGSTREET}}` `{{ACCOUNT_BILLINGCITY}}`,  
`{{ACCOUNT_BILLINGSTATE}}` `{{ACCOUNT_BILLINGPOSTALCODE}}`  
 INDUSTRY: `{{ACCOUNT_INDUSTRY}}`  
 ANNUAL REVENUE: `{{ACCOUNT_ANNUALREVENUE \@# CURRENCY}}`

**Master data**

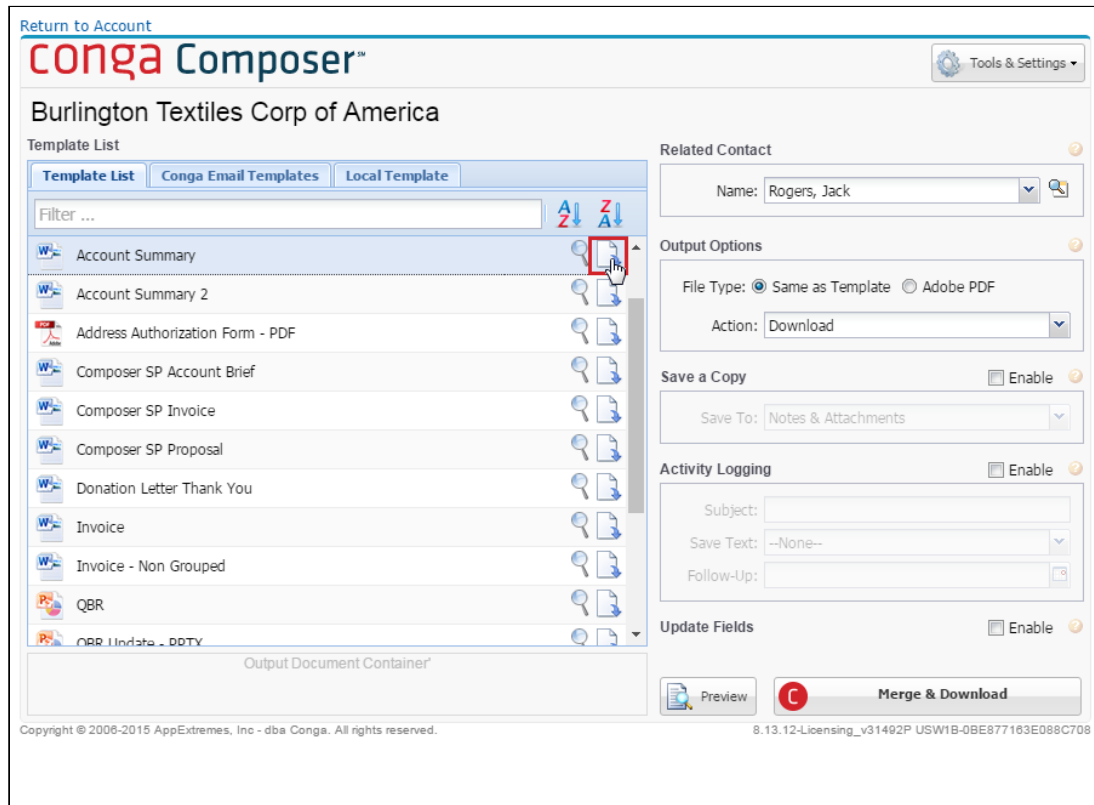
Contacts

First Name	Last Name	Phone	Title
<code>{{TableStart:Contacts}}<code>{{First_Name}}</code></code>	<code>{{Last_Name}}</code>	<code>{{Phone}}</code>	<code>{{Title}}</code> <code>{{TableEnd:Contacts}}</code>

**Detail data**

`{{ IF {{ACCOUNT_ANNUALREVENUE}} > 2000000 "THIS IS A HIGH VALUE ACCOUNT WITH ANNUAL REVENUE OF {{ACCOUNT_ANNUALREVENUE \@# CURRENCY}}." "THIS IS NOT A HIGH VALUE ACCOUNT." }}`

**i** A quick way to open your template as a local file is to download it from the Composer user interface. It will then appear in your downloads folder. To do so, click the Download icon next to the template you want to update (see below).




2. Next, navigate to a record that has the Composer button on it and launch the button.
3. From the Composer user interface that opens, select the Tools & Settings drop-down and select the Template Builder.
4. Locate the new query dataset you created for the solution and the new merge fields inside the dataset. These are the fields you will use in your template.

## Update Template

There are several options for updating the merge fields in your template. If the data is in a table, as in our example, it might be easiest to replace the entire table using the Template Builder. You can also replace merge fields one at a time if you do not want to re-create the entire table. This is the approach we will take in our example.

To update the template, we will use the same techniques used when building templates with the Template Builder. Follow the steps below.

1. To copy an individual merge field to your template, select the field and click the Copy Selected Item icon next to the field. This copies the field to your clipboard.

 To copy more than one field at a time, double-click each merge field to move it over to the Template Canvas. Then you can copy and paste (or drag and drop) all of the fields from the Template Canvas into your template at once.


2. In your template, replace the old merge field with the new one you just copied to your clipboard. Repeat for all the merge fields in your detail dataset.
3. Lastly, update the `{{TableStart}}` and `{{TableEnd}}` merge fields. For a reminder about how these works, see [Using Detail Data in Word](#).
4. When you are done, save your template locally.
5. Test your template before completing the next step. To do so, click **Back** to return to the Composer user interface and then run a test merge by choosing your template from the Local Templates tab. If the new merge fields merge correctly, you can continue to the next step. If not, correct any mistakes before continuing.

## Update the Conga Template Record

Finally, upload the new template to the same template record. This will prevent you from having to update the button URL because the `&TemplateId` parameter can continue to reference the same Id.

Follow the steps below to complete this final task.

1. Navigate to the template record for the template you just updated.

 A quick way to do this is from the Composer user interface. Click the magnifying glass icon next to the template to open the template record in a new tab (see below). The other option is to navigate to the Conga Templates object and find the template record from there.

2. Click **Attach File**.
3. Click **Choose File**. Locate the updated template you saved locally.
4. Click **Attach File**.
5. Click **Done** and close the window.

You do not need to delete the old template because Composer will automatically use the newest attachment on the record. However, you can delete it if you like. You have now converted your reports to queries. Run the Composer solution one more time to confirm everything looks correct with the new query and the new template.

## Other Considerations

At this point, the main concerns of migrating Conga Composer to Salesforce Lightning have been addressed. However, you may have other considerations based on your specific implementation of Conga Composer. Some of these include:

- Home Page Links
- JavaScript Buttons
- Conga Mail Merge and Conga QuickMerge
- Documents and Solution Packs
- Home Page Links

You cannot create Home Page components for Composer links in Salesforce Lightning. If you have any of these, we recommend moving them to a [Global Merge](#) record or using the Utility Builder to design a new layout.

## JavaScript Buttons

If you use Conga List View buttons or if you have created a custom Conga JavaScript button, these will not work in Salesforce Lightning. We recommend you choose a different implementation technique or engage Conga Services to build a custom solution for your situation.

## Conga Mail Merge and Conga QuickMerge

Conga Mail Merge and Conga QuickMerge currently do not work in Salesforce Lightning. Global Merge can replicate the PointMerge capabilities of Conga QuickMerge, but not the MassMerge capabilities of Conga QuickMerge. There is not currently a workaround.

## Documents and Solution Packs

Documents are not supported in Salesforce Lightning. If you are using existing Conga Solution Packs (which use Documents) or are using Conga templates saved as Documents, you must convert them to Salesforce Files. In Composer 8.15 and newer, you can utilize Quick Starts.

## Composer for Salesforce1

Select one of the following topics for more information:

- [About Conga Composer for Salesforce1](#)
- [Composer for Salesforce1 Configuration](#)
- [Configuring the Connected App Settings for Composer for Salesforce1](#)
- [Global Publisher Action for Composer for Salesforce1](#)
- [Supported DS7 Values for Salesforce1](#)
- [Supported Functionality in Composer for Salesforce1](#)
- [Run a Composer for Salesforce1 Solution](#)
- [Adding Composer for Salesforce1 Parameter to the Button URL](#)
- [Best Practices for Conga Composer for Salesforce1](#)
- [Configure the Composer Lightning Component for Salesforce Mobile](#)

## About Conga Composer for Salesforce1

Leveraging the power of the Salesforce1 Platform, Conga Composer for Salesforce1 enables Salesforce customers to create and deliver customized documents, presentations and reports from mobile devices. The thousands of solutions created by our customers can now be launched right from their user's phone or tablet.

**⚠ Users must launch a Composer button with an OAuth base URL in the standard Salesforce user interface before launching it in Salesforce Mobile. This establishes the OAuth connection that allows users to launch the Composer button in Salesforce Mobile.**

Before setting up a Composer solution to work with Salesforce1, first make sure it works as intended in the browser.

## Composer for Salesforce1 Configuration

How to configure Composer Salesforce1.

### Prerequisites

- Salesforce Enterprise Edition or above
- Working Composer solution in [Background Mode \(DS7 parameter\)](#) using a supported setting.
- Access to the Conga Composer SF1 Connected App
- Salesforce1 for your supported device
- Conga Composer 8.4.1 or greater
- Only available for certain supported objects. To see a list of objects, see [Supported Functionality in Composer for Salesforce1](#).

## Steps to set up Composer for Salesforce1

1. [Connected Apps Configuration](#)
2. [Add Global Publisher Action to desired page layout](#)
3. [Add &SF1=1 parameter to button](#)

## Configuring the Connected App Settings for Composer for Salesforce1

In order to use Composer for Salesforce1, you must assign access to the Conga Composer SF1 Connected App for specific Salesforce users or profiles.

To Configure Composer for Salesforce1:

1. Navigate to **Setup** → **Administer** → **Manage Apps** → **Connected Apps** → **Conga Composer SF1**.
2. Edit the record and change OAuth policies-Permitted Users to Admin approved users are pre-authorized.
3. Save changes.
4. Assign the desired profiles to the Connected App or create a Permission Set and assign that to the desired user(s).

Assigning profiles to the connected app or creating a permission set is required in order to use Composer for Salesforce1.

Now enable access to the app for non-Admin users.

 This token is not required for Conga Quote Generation.

To enable access to non-admin users:

1. Navigate to Installed Packages.
2. Navigate to the Conga Composer installed package.
3. Click **Configure**.
4. On the Basic tab in the Offline Access area, click **Create Salesforce Token** (or Update Salesforce Token if already created).

A button or link does not have to be on a page layout in order to access it in Salesforce1.



## Global Publisher Action for Composer for Salesforce

Add the Conga Composer Global Publisher Action to the target object page layout to set up Composer solutions for Salesforce.

To add the Conga Composer Global Publisher Action to the target object page layout:

1. Edit the Page Layout of the object where you've created solutions for Composer for Salesforce.
2. Click on Actions.
3. Add the Conga Composer action to the Publisher Action layout by dragging and dropping the action to the Publisher Actions section.
4. Click **Save**.

It is highly recommended that administrators do not include the Conga Composer Global Publisher Action as the first Publisher Action that appears in the Action Bar. The first Publisher Action in the Action Layout will open by default in the Chatter Feed for desktop users, which can be undesirable considering Composer for Salesforce solutions are optimized for mobile devices.

To ensure Composer appears in a browser in Salesforce:

1. In Account Layout, click **Quick → Actions**.
2. In the Quick Actions in the Salesforce Classic Publisher section, add the Conga Composer action to the layout.

To ensure Composer appears in a mobile device in Salesforce:

1. In Account Layout, click **Quick → Actions**.
2. In the Salesforce and Lightning Experience Actions section, add the Conga Composer action to the layout.

## Supported DS7 Values for Salesforce

The following is a table of use cases with the corresponding DS7 (background automation) parameter value(s) that is supported as well as unsupported in Composer for Salesforce.

### Supported DS7 Values

DS7 Functionality	DS7 Parameter Values
Email	12

DS7 Functionality	DS7 Parameter Values
eSignature	4, 7, 14, 17, 301, 302
Store in Salesforce	1, 5, 6
Store in SpringCM	8
Google Drive®	9, 19
Chatter®	401

### Unsupported DS7 Values

DS7 Functionality	DS7 Parameter Values
Email	2
Adobe eSign	24
Merge & Download	3, 13

## Supported Functionality in Composer for Salesforce1

Functionality supported in Composer for Salesforce1.

### Email

- Conga Email Templates (CETs) and Salesforce® Email Templates are supported.
- SendGrid is supported.
- During the sending process, the user does not have access to the email editor.

### eSignature

- Auto-delivery is recommended.
- Adobe eSign Agreement Templates are supported.

## Supported Vs Unsupported Objects

Global Actions are not supported on all Standard Salesforce objects. Since you need to add the Conga Composer global publisher action to the object page layout, Composer for SF1 will only work on the following:

- Account
- Campaign
- Case
- Contact
- Custom objects
- Group
- Lead
- Opportunity
- Quote
- Orders

Composer for SF1 does not work on the following Objects:

- Contracts
- Products

## Lightning Components

Composer for SF1 does not work when lightning components are enabled.

## Storing in Salesforce

Below is a table that outlines how well Salesforce1 handles each method to store and what is recommended when setting up a Composer for Salesforce1 solution.

Chatter (Recommended)	Notes & Attachments (applies to any use of &SC1=Attachments)
<ul style="list-style-type: none"> <li>• Well supported in Salesforce1</li> <li>• Best Practice to use Chatter to store in Salesforce when using Composer for Salesforce1</li> <li>• Files can be downloaded to your mobile device</li> </ul>	<ul style="list-style-type: none"> <li>• Not well supported in Salesforce1</li> <li>• No capability to download PDF or .docx files on mobile devices</li> </ul>

## Google Integration

If Google Drive is going to be used, a refresh token must be set up.

To set up Google Drive Integration:

1. Navigate to Conga Composer Setup.
2. Click **Integration Partners**.
3. Select the Google Drive tab.
4. Click **Update Google Token**.
5. Navigate to the Conga Composer Setup tab.
6. Follow the steps to update the token.

## Run a Composer for Salesforce Solution

Using Composer for Salesforce is simple and streamlined.

To run a Composer for Salesforce Solution:

1. From a desired Master object record, tap the Conga Composer action at the bottom of the screen.
2. The buttons available for Salesforce for that Master Object appear.
3. Tap or select a button to run the solution.

⚠ If an expected button does not appear in the list, ensure that the solution uses a supported DS7 value and includes the Salesforce parameter (&SF1=1).

4. Depending on the configuration of the solution, there may be a document to view and download once the process has completed.

⚠ To save the document to the device, open the document in a new browser and then save as desired. Users must launch a Composer button with an OAuth base URL in the standard Salesforce user interface before launching it in Salesforce Mobile. This establishes the OAuth connection that allows users to launch the Composer button in Salesforce Mobile.

## Adding Composer for Salesforce Parameter to the Button URL

For a solution to appear as an available button on the mobile device, the button URL must contain the SF1 parameter.

To add a parameter to enable Salesforce integration:

1. Navigate to the Master Object.
2. Edit the desired button.
3. Add the &SF1=1 parameter/value pair to the URL.


## Best Practices for Conga Composer for Salesforce

- Before setting up a Composer solution to work with Salesforce, make sure it works as intended in the browser.
- It's highly recommended that administrators do not include the Conga Composer Global Publisher Action as the first Publisher Action that appears in the Action Bar.
- **WHY?** The first Publisher Action in the Action Layout will open by default in the Chatter Feed for desktop users, which can be undesirable considering Composer for Salesforce solutions are optimized for mobile devices.

## Configure the Composer Lightning Component for Salesforce Mobile

Configure the Composer Lightning Component as a Global Action to easily generate documents from Salesforce Mobile. Initiating the Composer Lightning Component from a Global Action in Salesforce Mobile allows users to easily generate files from mobile devices such as smartphones and tablets.

Follow the steps below to configure and use the Composer Lightning Component as a Global Action in Salesforce Mobile. For more information about the Composer Lightning Component, see [Composer Lightning Component](#).

 The Lightning Component's Global Action works in both Salesforce Mobile and Lightning.

### Prerequisites

- Assign the necessary [profile permissions](#) to user profiles using the Composer Lightning Component. Additionally, assign the **Modify Metadata Through API Functions** system permission to the user profiles.

⚠ Users are required to have the Modify Metadata Through API Functions system permission to use the Composer Lightning Component in Salesforce Mobile, but it is not required to use the Composer Lightning Component in Salesforce Lightning.

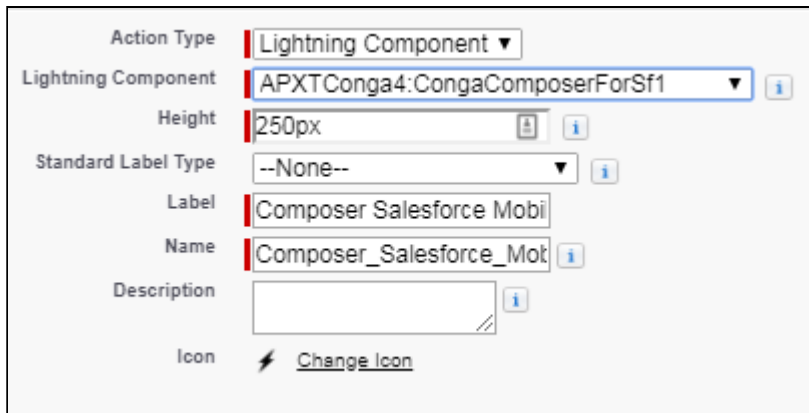
- [Create a Solution Collection](#).
- Create a Composer Solution using the [Conga Solution Manager](#) or the [Document Automation Wizard](#).

**Important**

iOS users who use Composer & Sign in Salesforce1 must *Enable Cross-Website Tracking* using the steps outlined in the [Salesforce article](#) to ensure that Composer & Sign is successfully executed.

## Configure the Lightning Component as a Global Action

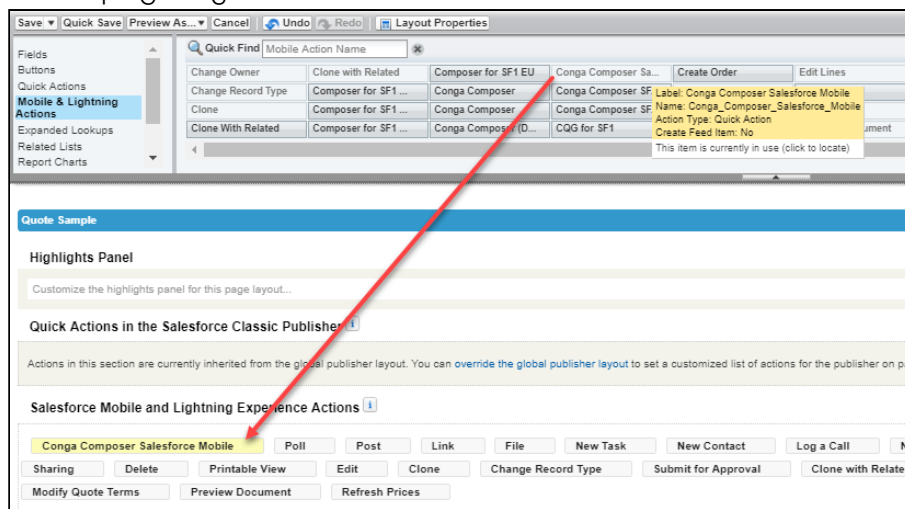
1. Navigate to an existing Conga Solution Collection.
2. Toggle Set As Salesforce1 Collection to Yes.
3. Navigate to Salesforce Setup.
4. Search for Global Actions in the Quick Find Search Box and click **Global Actions**.
5. Click **New Action** to create a new Global Action.
  - Choose Lightning\_Component.
  - Choose APXTConga4:CongaComposerForSf1.
  - Fill out the Label and Name fields to name the Global Action.



Action Type	Lightning Component
Lightning Component	APXTConga4:CongaComposerForSf1
Height	250px
Standard Label Type	--None--
Label	Composer Salesforce Mobi
Name	Composer_Salesforce_Mot
Description	
Icon	⚡ <a href="#">Change Icon</a>

- Click **Save**.

6. Navigate back to Salesforce Setup and click **Object Manager** under the Objects and Fields section.
7. Click the Conga Solution Collection's corresponding object in the Object Manager.
  - For example, if the Conga Solution is built for the Account object, click the Account object in the Object Manager.
8. Click **Page Layouts**.
9. In the Object Layout menu, click **Mobile & Lightning Actions**.
10. Drag the new Global Action onto the Salesforce Mobile and Lightning Experience Action page layout








11. Click **Save**.

## Initiate the Lightning Component from a Global Action in Salesforce Mobile

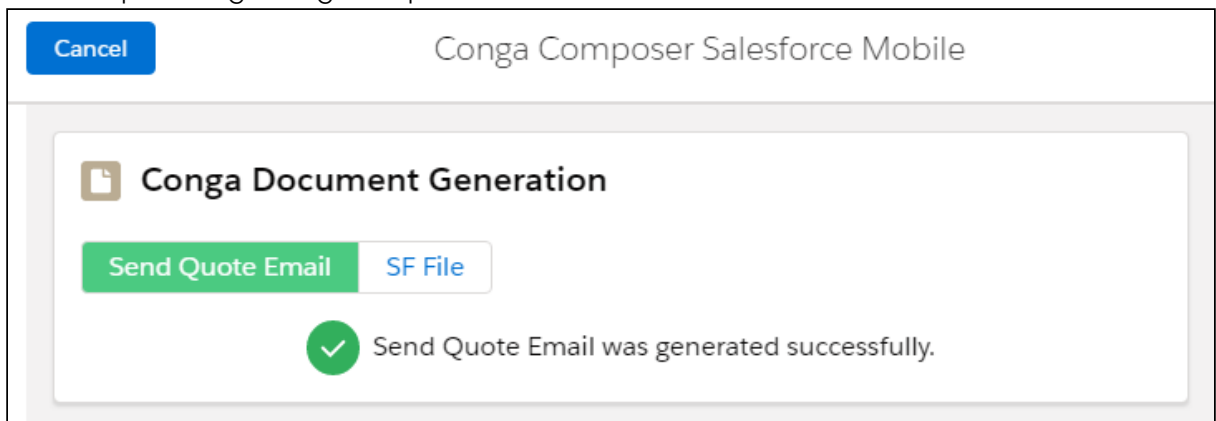
1. Navigate to an object record in Salesforce Mobile where the Composer Lightning Component's Global Action is present.

2. Click the recently created Global Action to initiate and access the Composer Lightning Component.

Additional Disc. (%)		List Amount	
\$0.00		\$0.00	
Avg. Customer Disc. (%)		Regular Amount	
0.0%		\$0.00	
Total Customer Disc. Amount		Customer Amount	
\$0.00		\$0.00	
Addl. Disc. Amount		Net Amount	
\$0.00		\$0.00	

3. Choose the desired Solution from the Solution Collection to launch Composer through the Composer Lightning Component.



## Composer Advanced Features

Composer Advanced Features is a new functionality that provides users with access to Composer Link Service, the Conga File Repository, Notification Emails, and the Conga PDF Viewer.

[Composer Link Service](#) allows users to use Composer to easily generate and send documents in an email directly from Salesforce. Each document generated with Composer Link Service is given a unique, shareable link and is associated with a Document History record in Salesforce.

Documents generated with Composer Link Service are securely stored in the proprietary [Conga File Repository](#).




Users and Recipients can then easily view documents generated by Composer Link Service in the [Conga PDF Viewer](#). The Conga PDF Viewer provides users and recipients access to documents without having to log into Salesforce.

Composer Link Service also provides users with insightful [Notifications Emails](#) to inform users when recipients interact with documents sent with Composer Link Service.

- [Composer Link Service](#)
- [Notification Emails](#)
- [Conga File Repository](#)
- [Document History and Document History Detail Objects](#)
- [Conga PDF Viewer](#)

## Composer Link Service

Composer Link Service is a Composer Advanced Feature that allows users to create and send documents through email. Composer solutions using Composer Link Service create a unique link for each document, making documents easily shareable and accessible. Word, Excel, and PowerPoint files sent with Composer Link Service are downloaded to the browser tray once the recipient clicks the document link in the email. PDF files are automatically displayed in the [Conga PDF Viewer](#).

 Composer Link Service supports sending PDF, Word, Excel, and PowerPoint files. The [Conga PDF Viewer](#) only supports PDF files.

The Composer document link is kept on the [Document History](#) object in Salesforce. Each online document generated with Composer Link Service has its own related Document History record. The document is stored in Salesforce on the backend within the proprietary [Conga File Repository](#).

Composer Link Service also provides users with the ability to receive customized [Notification Emails](#) when recipients interact with documents sent with Composer Link Service.

To enable Composer Link Service, see [Enabling Composer Link Service and Conga File Repository](#).

Setup for Composer Link Service includes adding the [LinkService parameter](#) to a Composer solution that has a designated Conga Email Template (using the [CongaEmailTemplateId parameter](#)). Then add the `{Conga_Doc_Link}` merge field in the body of the Conga Email Template to dynamically generate the link in the sent email.

⚠ The {Conga\_Doc\_Link} merge tag generates the unique link to the document and is required in the Conga Email Template to leverage Composer Link Service.

For more detailed information on setting up Conga Link Service, see the following documentation:

- [Setting Up Composer Link Service for New Solutions in the Document Automation Wizard](#)
- [Manually Setting Up Composer Link Service for New Solutions](#)
- [Setting Up Composer Link Service for Existing Solutions in the Document Automation Wizard](#)
- [Manually Setting Up Composer Link Service for Existing Solutions](#)
- [Composer Link Service Parameters](#)

## Manually Setting Up Composer Link Service for Existing Solutions

Use Conga Link Service in existing Composer solutions. You must enable [Composer Link Service and the Conga File Repository](#) to setup and use Composer Link Service. Additionally, you must [enable Notification Emails](#) in Conga Composer Setup to send Notification Emails.

⚠ The solution must have a Conga Email Template (CongaEmailTemplateID parameter) to send a document with Composer Link Service in an email.


Modify an existing Composer solution in Salesforce Setup to leverage Composer Link Service.

1. Navigate to the object that the Composer solution exists on in Salesforce Setup.
2. Click **Buttons, Link, and Actions**.
3. Click the link to a specific Button, Link, or Action, and then click **Edit**.
4. Add the following parameters to the Composer URL.
  - a. [TemplateID](#)
  - b. [LinkService](#)
  - c. [CongaEmailTemplateID](#). Additionally, add the required `{Conga_Doc_Link}` merge tag in the email body of the Conga Email Template referenced in the solution.

⚠ The CongaEmailTemplateID must have the {Conga\_Doc\_Link} merge tag in the email body to generate the document link.

5. Add the following parameter to send [Notification Emails](#).

- a. [NETId](#)
  - b. [EmailTold](#) or a different [Conga Email Template parameter](#) to designate recipients
6. (Optional) Add [DS7](#) to run the solution in background mode.
  7. (Optional) Add the [DefaultPDF](#) parameter to convert the file to a PDF and view it in the [Conga PDF Viewer](#).

 The Conga PDF Viewer only supports PDF files. Word, Excel, and PowerPoint are automatically downloaded to the browser tray once a recipient clicks the document link.

8. Click **Save** after adding the parameters.

## Setting Up Composer Link Service for Existing Solutions in the Document Automation Wizard

Set up Composer Link Service for an existing Composer solution in the Document Automation Wizard.

You must enable [Composer Link Service](#) and the [Conga File Repository](#) to setup and use Composer Link Service. Additionally, you must [enable Notification Emails](#) in Conga Composer Setup to send Notification Emails.

 The Conga Solution must have a Conga Email Template (CongaEmailTemplateID parameter) to send a document with Composer Link Service in an email.

Modify an existing solution to leverage Composer Link Service.

1. Navigate to Conga Solutions and select a Conga Solution record.
2. Click **Edit (Guided)**.
3. Click **Assign Behaviors**.




4. Under the Link Management section, click **Link Configuration**.
5. Click the Link Service toggle to switch it to *Yes* and click **Save**.
  - This automatically adds the [&LinkService=1 parameter](#) to the Conga Solution.
6. Enable Email Notifications for Composer Link Service.
  - a. Click **Notifications** under the Link Management section.
  - b. Click the Email Notifications toggle to switch it to *Yes*.
  - c. Select a Conga Email Template record to use as the Notification Email Template.
    - This automatically adds the NETId parameter to the Conga Solution.

- d. Click **Save**.
- 7. Add the required Email parameters to send a document in an email with Composer Link Service.

- a. Click **Add Templates**.



- b. Click **Create Conga Email Template** and fill out the fields to customize and create the Conga Email Template.
- c. Add the required `{Conga_Doc_Link}` merge tag in the email body.

 The CongaEmailTemplatelD must have the `{Conga_Doc_Link}` merge tag in the email body to generate the document link.

- d. Click **Save**.
- e. Click **Assign Behaviors**.



- f. Click **Email** in the Post Merge Distribution section.
  - g. Select a value for the *To* field to specify who receives the email.
  - h. If applicable, select values for the *Additional To*, *CC*, and *BCC* fields to further customize the email.
- 8. Under the General section, click **Generated File**.
  - 9. (Optional) Click the Convert to PDF toggle to switch the value from *No* to *Yes*. PDF files are automatically opened up in the [Conga PDF Viewer](#) when a recipient clicks a document link.

 The Conga PDF Viewer only supports PDF files. Word, Excel, and PowerPoint are automatically downloaded to the browser tray once a recipient clicks the document link.


- 10. (Optional) Enable Background Mode under the Behavior Settings section.
  - a. Click **Assign Behaviors**.
  - b. Click **Background Mode** and select from the following options. Once selected, click **Save**.
    - Link & Email with Preview (DS7=550)
    - Link & Email Auto Delivery (DS7=551)
    - Link without Email Auto Delivery (DS7=552)
- 11. Click **Save**.

## Manually Setting Up Composer Link Service for New Solutions


Manually set up a Composer Link Service in a new Composer solution. You must [enable the Conga File Repository](#) and [enable Notification Emails](#) in Conga Composer Setup to use Composer Link Service.

In Salesforce Setup, manually create a new Composer solution to online documents with Composer Link Service.

1. [Create a Composer OAuth Button](#).
2. Add the following parameters to the Composer URL.
  - a. [TemplateID](#)
  - b. [LinkService](#)
  - c. [CongaEmailTemplateID](#). Additionally, add the required {Conga\_Doc\_Link} merge tag in the email body of the Conga Email Template referenced in the solution.
  - d. [EmailToId](#) or a different [Conga Email Template parameter](#) to designate recipients.

 The CongaEmailTemplateID must have the {Conga\_Doc\_Link} merge tag in the email body to generate the document link.

3. Add the following parameter to send [Notification Emails](#).
  - a. [NETId](#)
4. (Optional) Add [DS7](#) to run the solution in background mode.
5. (Optional) Add the [DefaultPDF](#) parameter to convert the file to a PDF and view it in the [Conga PDF Viewer](#).

 The Conga PDF Viewer only supports PDF files. Word, Excel, and PowerPoint are automatically downloaded to the browser tray once a recipient clicks the document link.

6. Click **Save**.
7. Add the button to a Page Layout.

## Setting Up Composer Link Service for New Solutions in the Document Automation Wizard

You must enable [Composer Link Service](#) and the [Conga File Repository](#) to setup and use Composer Link Service. Additionally, you must [enable Notification Emails](#) in Conga Composer Setup to send Notification Emails.

**⚠ The Conga Solution must have a Conga Email Template (CETID parameter) to send a document with Composer Link Service in an email.**

Create a new solution that uses Composer Link Service to create and send documents.

1. Click the Conga Quick Start tab.
2. Click **Launch Wizard** to launch the Document Automation Wizard
3. Add Solution Details.
  - a. Fill out the Solution Name field to name the solution.
  - b. Select a Salesforce Object to launch the solution from.
  - c. Find a sample record to test with the solution.
4. Click **Create**.
5. Create or add SOQL Queries to gather additional data and then click **Save**.
6. Create, upload, or choose an existing **Conga Template** to determine that the document that is generated and sent.
7. Click **Create Conga Email Template** and fill out the fields below.
  - a. Fill out the Email Template Name to name the Conga Email Template.
  - b. Fill out the Subject to determine the email subject's text.
  - c. Add the required `{Conga_Doc_Link}` merge tag in the email body of the Conga Email Template referenced in the solution.


**⚠ The CongaEmailTemplatelD must have the `{Conga_Doc_Link}` merge tag in the email body to generate the document link.**

8. Click **Save**.
9. Click Assign Behaviors.



10. Under the Link Management section, click **Link Configuration**.
11. Click the Link Service toggle to switch it to *Yes* and click **Save**.
  - This automatically adds the `&LinkService=1` parameter to the Conga Solution.
12. Enable Notification Emails for Composer Link Service.
  - a. Click **Notifications** under the Link Management section.
  - b. Click the Email Notifications toggle to switch it to *Yes*.
  - c. Select a Conga Email Template record to use as the Notification Email Template.
    - This automatically adds the `NETId` parameter to the Conga Solution.
  - d. Click **Save**.
13. Under the General section, click **Generated File**.

14. (Optional) Click the Convert to PDF toggle to switch the value from *No* to *Yes*. PDF files are automatically opened up in the [Conga PDF Viewer](#) when a recipient clicks a document link.

 The Conga PDF Viewer only supports PDF files. Word, Excel, and PowerPoint are automatically downloaded to the browser tray once a recipient clicks the document link.

15. (Optional) Enable Background Mode.
  - a. Click **Background Mode** and select from the following options. Once selected, click **Save**.
    - Link & Email with Preview (DS7=550)
    - Link & Email Auto Delivery (DS7=551)
    - Link without Email Auto Delivery (DS7=552)
16. Add the solution to a Solution Collection.
 

If a Solution Collection is not yet created, [create a Solution Collection](#).

  - a. Click a Solution Collection under Available Solution Collections.
  - b. Click the + symbol next to the new solution and then click **Save**.
17. Add the solution to your Page Layout.
  - Click Go to Lightning App Builder to [add the Composer Lightning Component to a Lightning Page Layout](#).
  - Click Go to Page Layouts to add the solution as a button, link, or action on a Page Layout.

## Composer Link Service Parameters

Use these parameters to generate and send a document with Composer Link Service.

### Parameters for Composer Link Service

- [LinkService](#)
  - Enables Conga Link Service and generates a Document History record with a unique document link
- [DefaultPDF](#)
  - Converts the output file to PDF and allows recipients to view the document in the Conga PDF Viewer.
- [CongaEmailTemplateId](#)
  - Important: The Conga Email Template must contain the required `{Conga_Doc_Link}` merge tag to generate the document link in the email.
  - Identifies the Conga Email Template that is sent with the document link.

⚠ You must also identify the recipient with either the [EmailTold](#) or other additional [Conga Email Template parameters](#). If the `CongaEmailTemplateId` parameter is not included in the Composer solution, a Document History record with a unique document link is generated and no email is sent.

- [NETId \(Notification Email Template\)](#)
  - Identifies the Conga Email Template that is sent to users when recipients interact with the document
- [DS7](#)
  - Enables background for Composer Link Service
  - The following values are designated for Composer Link Service: 550, 551, and 552

## Notification Emails

Notification Emails are emails sent to the running Composer user when recipients interact with documents sent from Composer Link Service.

To begin using Notification Emails with Composer Link Service, the Salesforce Org must have Composer Advanced Features provisioned and the Email Notifications feature enabled in Conga Composer Setup.

⚠ Contact your Conga Representative to enable the [Composer Advanced Features](#) in order to use **Email Notifications**.

Once provisioned and enabled, add the [NETId parameter](#) to a Composer solution that also has the [LinkService parameter](#) and other required [Composer Link Service parameters](#) to send Notification Emails.

⚠ The NETId parameter value must be the 15 or 18 digit record Id of a Conga Email Template record. The Conga Email Template subject and body must contain content or the Notification Email will not send.

A Notification Email is sent to the Composer user that executes and sends the document. A Notification Email is sent to the user the first time the document is opened. Additional Notification Emails are not sent to the user after the initial notification email.



⚠ If a Composer solution uses Composer Link Service to deliver a document and the NETId parameter is not present, no Notification Email is sent to the user.

- [Enabling Notification Emails](#)

## Enabling Notification Emails

⚠ Contact your Conga Representative to enable the [Composer Advanced Features](#) in order to use **Email Notifications**.

Perform the following steps to enable Notification Emails in Conga Composer Setup.

1. Click the **Conga Composer Setup** tab.
2. Toggle Email from *Disabled* to *Enabled* in the *Enable Email Notifications* section.
3. Click **Save**.

## Conga File Repository

The Conga File Repository is a proprietary repository used to store documents generated with the Composer Link Service.

⚠ The repository is separate and not related to Salesforce Files and Notes and Attachments.


All documents generated with Composer Link Service are automatically uploaded to the repository and accessible with a link on the Document History object.

The repository works in the background and is not tangibly accessible to Salesforce admins or users.

⚠ All Composer documents stored in the repository expire by default after 547 days (a year and a half) from the initial document generation.

To enable the Conga File Repository, see [Enabling Composer Link Service and Conga File Repository](#).

## Enabling Composer Link Service and Conga File Repository

 Contact Conga Support to provision Composer Link Service and the Conga File Repository. To enable Composer Link Service and the Conga File Repository, you must sign an updated MSA

Enable Composer Link Service and the Conga File Repository to generate and send documents with Composer Link Service.

1. Click the Conga Composer Setup tab to access Conga Composer Setup.
2. Click the File Storage toggle and toggle it from *Disabled* to Enabled.
3. Click **Save**.

## Document History and Document History Detail Objects

The **Document History** object is a custom object that contains the link to a document (stored in the Conga File Repository) generated with Composer Link Service.

Composer automatically creates a unique Document History record specific to each document. Each Document History record is generated with a related Document History Detail child record.


The **Document History Detail** object is a child object of the Document History object. It contains key information such as the number of document views and the date/time that the document was last viewed.

 Document History records are accessed in the Document Histories tab.

## Document History Fields

- Document History Name
  - Auto-generated record name, such as *DH-001*.
- Conga Solution
  - Lookup field that displays a link to the related Conga Solution that generated the document
- URL
  - Displays the click-able hyperlink to the document
- Link Text

- Displays the hyperlink text used in the Notification Email Template to access the document
- By default, Link Text is set to *Click here to view document*.

 Current functionality does not allow users to change the default Link Text value.

- Last Viewed
  - Displays the date/time for when the document was last viewed
- Master Object ID
  - Displays the specific record ID that the document is generated from
- Number of Views
  - Displays the total number of views for the document
- Created By
  - Displays the user that created the Document History record and the date/time of the record creation
- Owner
  - Displays the current record owner
- Last Modified By
  - Displays the user that last modified the Document History record and the date/time of the record modification


## Document History Detail

- Document History
  - Lookup field to the related parent Document History record.
- Document Status
  - Lookup field that displays a link to the related Conga Solution that generated the document
- Email Address
  - Displays the email address the document is sent to.
- Number of Views
  - Displays the number of times the document is viewed
- Last Viewed
  - Displays the date/time for when the document was last viewed

## Conga PDF Viewer

View documents generated with Composer Link Service in the Conga PDF Viewer. The Conga PDF Viewer allows users and recipients to easily view documents generated with Composer Link Service.

Documents are automatically opened in the Conga PDF Viewer when clicking the document link in an email sent from Composer Link Service. Salesforce users can also click the hyperlink in the URL field on the [Document History object](#) to view online documents in the Conga PDF Viewer.


 The Conga PDF Viewer only supports PDF files. Users must use a PDF Template or the [DefaultPDF parameter](#) to generate a PDF file and view the file in the Conga PDF Viewer.

The Conga PDF Viewer allows users to easily zoom in and out by percentage or page fit, skip to specific pages within the document, and download the file locally.

## Composer Lightning Component

The Composer Lightning Component allows you to relate a [Solution Collection](#) to an object's page layout. All Conga Solutions in the selected Solution Collection become available to launch when the Composer Lightning Component is placed on an object's page layout.

The functionality of the Solution Collection and the Composer Lightning Component allow you to choose and launch different yet related Composer Solutions on the same Lightning page layout.

 **From March 1, 2022**, you must have Conga Composer version **8.200-Wilson Peak** or **later** installed in your Salesforce org in order to use the *Composer Lightning Component* to merge and generate the document.

The screenshot displays a Salesforce record page for 'Burlington Textiles Corp of America'. The page is divided into several sections: 'Related' (with tabs for Details and News), 'Contacts (1)' (listing Jack Rogers), and 'Opportunities (1)' (listing Burlington Textiles Weaving Plant Generator). On the right side, there is a 'Conga Document Generation' component highlighted with a red box, containing buttons for 'Composer SP Proposal', 'Quote Document', and 'Walkers Document Button'. Below this is an 'Activity' section with a 'New Task' button and a 'Next Steps' section.

## Add the Composer Lightning Component to an Object Record Page

Ensure the pre-requisites are fulfilled and then choose between the two methods to add the Composer Lightning Component onto an Object Record Page.

**⚠** You must enable My Domain to use custom Lightning components, including the Composer Lightning Component. For more information, see [Composer Lightning Component](#) and [Set Up a My Domain Name](#).

Before you begin, see the **Composer Lightning Component Pre-Requisites** below:

- Enable [Salesforce My Domain](#).
- Install Conga Composer 8.20.58 or higher.
- Create a [Solution Collection](#).
- Assign the necessary [profile permissions](#) to user profiles using the Composer Lightning Component.

**⚠** The Modify Metadata Through API Functions system permission is required for users that run and execute the Composer Lightning Component in Salesforce Mobile.

## Adding the Composer Lightning Component from an Object Record

Use this method if you already created the Conga Solution or Solution Collection that is launched from the Composer Lightning Component.

⚠ It is recommended to add one *Composer Lightning Component* per *Lightning Record Page* as best practice.

## To add the Composer Lightning Component From an Object Record

1. Navigate to a record of the object in Lightning.
2. Click the Setup gear wheel icon in the top right corner.
3. Select **Edit Page**.
4. On the left-hand side, scroll down to the Custom - Managed Lightning Components.
5. Drag the Conga Composer Lightning Component onto the Object Record Page.
6. Select the recently added Composer Lightning Component and choose a Solution Collection record from the Solution Collection picklist field.
7. Click **Save**.


## Adding the Composer Lightning Component Through the Document Automation Wizard

Use this method if you have not already created the Conga Solution or Solution Collection that is launched from the Composer Lightning Component.

## To Add the Composer Lightning Component Through the Document Automation Wizard

1. Click **Conga Quick Start**.
2. Click **Launch Wizard** under Document Automation Wizard or click the pencil icon next to an existing Conga Solution record.
3. Complete the initial four steps (Add Details, Gather Data, Add Templates, Assign Behaviors) or skip to the fifth step, Collection Step.
4. Choose an existing Solution Collection or click **Create Collection** to create a new Solution Collection.
  - If you create a new Solution Collection, click the + button to add Conga Solutions to the Solution Collection and then click **Save**.
5. Navigate to the sixth step in the Document Automation Wizard, Add Button to Page Layout.
6. Select **Go To Lightning App Builder**.

7. Click the Lightning Record Page where the Composer Lightning Component will live.
8. Click **Edit**.
9. On the left-hand side, scroll down to the Custom - Managed Lightning Components.
10. Drag the **Conga Composer Lightning Component** onto the Object Record Page.

 It is recommended to add one *Composer Lightning Component* per *Lightning Record Page* as best practice.

11. Select the recently added Composer Lightning Component and choose a Solution Collection record from the Solution Collection picklist field.
12. Click **Save**.

For information about specific features and parameter integration with the Composer Lightning Component, see [Composer Lightning Component FAQs](#).

## Supported Parameters for the Composer Lightning Component

The following parameters are available for use in the Composer Lightning Component.

### Background Mode Parameters

- [DS7](#)

#### **Note**

The BML and DS7Preview parameters are currently not supported for the Composer Lightning Component.

### Conga Appearance Parameters

- [DS1](#)

### Conga Email Template Parameters

- [CongaEmailTemplateGroup](#)
- [CongaEmailTemplateld](#)
- [EmailAdditionalTo](#)
- [EmailBCC](#)
- [EmailCC](#)
- [EmailLetterheadId](#)

- EmailRelatedTold
- EmailReplyTold
- EmailSubject
- EmailTemplateAttachments
- EmailTemplateld
- EmailTold
- EmailUseSignature

 **Note**

The DS10 parameter is currently not supported for the Composer Lightning Component.

## Data Gathering Parameters

- ContactId
- Id
- DV0 (Data Value)
- QueryId
- QVar0Id
- QVar1Id
- QVar2Id
- QVar0Format
- QVar1Format
- QVar2Format
- ReportId
- ReportVar
- XMD

## Template Selection Parameters

- DefaultLocal
- DS8
- SelectTemplates
- TemplateGroup
- Templateld

 **Note**



The DS4 and TNC parameters are currently not supported for the Composer Lightning Component.

## Output Parameters

- [AmazonS3Visible](#)
- [APDF](#)
- [APF](#)
- [AttachmentId](#)
- [AttachmentParentId](#)
- [AWD](#)
- [DefaultPDF](#)
- [FP0](#)
- [MTOFN](#)
- [OFN](#)
- [PDFA](#)
- [SelectAttachments](#)
- [ZipFiles](#)
- [ZipPDFs](#)

### **Note**

The LiveEdit parameter is not supported for the Composer Lightning Component.

## Activity Logging Parameters

- [AC0](#)
- [AC1](#)
- [DS3](#)
- [MFTS0](#)
- [MFTSValue0](#)
- [MFTS1](#)
- [MFTSValue1](#)
- [MFTS2](#)
- [MFTSValue2](#)
- [SC0](#)
- [SC1](#)

### **Note**

The AC2 and AC5 parameters are not supported for the Composer Lightning Component.

## Advanced Control Parameters

- [EC](#)

## Composer Link Service Parameters

- [LinkService](#)
- [NETId](#)

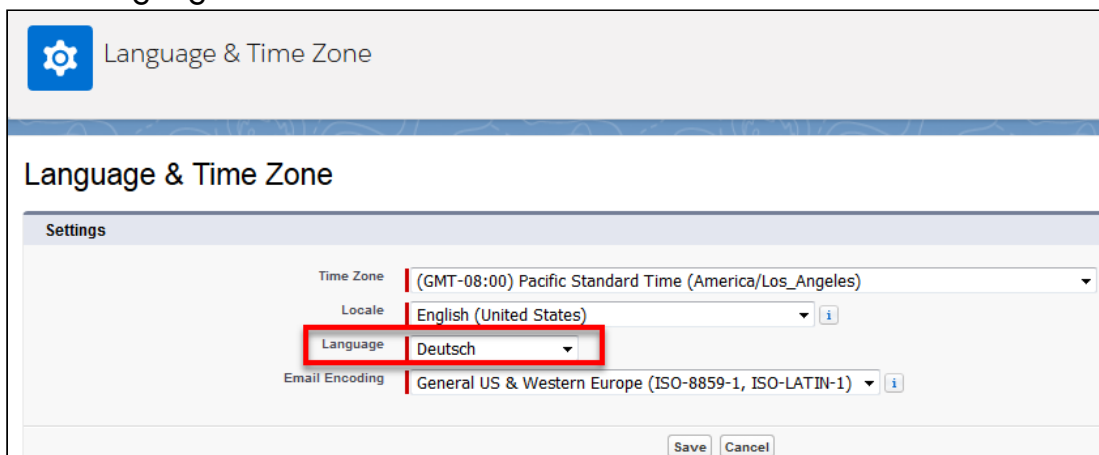
## Supported Languages for Composer Lightning Component UI

**Conga Composer Lightning Component** user interface supports dynamic translation for German, French, Dutch, Polish, and Spanish, in addition to English as a default language. It automatically translates to German, French, Dutch, Polish, or Spanish for the Salesforce Users based on their local language set in Salesforce.

⚠ You must have a minimum Conga Composer version 8.198-Longs installed in your Salesforce org to leverage German, French, Dutch, Polish, and Spanish dynamic translations.

To set dynamic translation


1. Under your Salesforce Profile, select **Settings**.
2. Click **Languages & Time Zone**.



3. Change the Language field to Deutsch (German), Français (French), Nederlands (Dutch), Polski (Polish), or Español (Spanish).
4. Click **Save**.

## Conga Keys

Use Conga Keys to streamline Composer solution migration from one Salesforce org to another.

 You must have Conga Composer version 8.186 (Quandary) or greater to use Conga Keys.

Conga Keys allows administrators to easily migrate Composer solutions between Salesforce orgs by leveraging the Key field on Conga Template, Conga Email Template, and Conga Query records.

The Key field automatically assigns a unique [external Id](#) to each Conga Template, Conga Email Template, and Conga Query record. The Key field allows administrators to reference Conga Templates, Conga Email Templates, and Conga Queries by an external Id that remains the same throughout different Salesforce orgs.

Conga Keys simplifies the solution migration process by eliminating the hassle of having to replace the record Id of each Conga Template, Conga Email Template, and Conga Query referenced in a Conga Solution. When migrating Conga Solution records, along with each Conga Template, Conga Email Template, and Conga Query record, Composer instantly recognizes each record by the Key field and the Conga Solutions are ready for use.

## Using Conga Keys

To reference Conga Templates, Conga Email Templates, and Conga Queries in a Conga Solution with Conga Keys, use a record's unique external Id in the Key field as a parameter value for [QueryId](#), [Templateld](#), [CongaEmailTemplateld](#) (CETId), or other additional Composer parameters.

Conga Template  
**CMT-00001**

Related **Details**

Template Name  
CMT-00001

Name  
Conga Proposal

Template Group ⓘ

Template Type  
Document

Key  
OT\_001UA0795984

⚠ Conga Keys is available for use in the Conga Solution Manager or Composer Solutions created manually in Salesforce Setup.

Add Report Data Add Query Data Add Templates Add Conga Email Template Customize with Parameters


Button URL in Progress /apex/APXTConga4\_\_Conga\_Composer?SolMgr=1  
&serverUrl={!API.Partner\_Server\_URL\_370}  
&Id={!Account.Id}  
&TemplateId=OT\_001UA0795984

In the example above, as a result of using a Conga Template record's Key field value (as opposed to the record Id), Composer and Salesforce successfully recognize the template in other Salesforce orgs when migrating the solution.

⚠ This example is applicable to any Conga Template, Conga Email Template, and Conga Query record referenced in a Conga Solution with its respective Key field's external Id.

Once all the Salesforce records for each Conga Solution, Conga Template, Conga Email Template, and Conga Query are migrated to an org, the Conga Solutions are ready for use. The administrator does not have to adjust the record Ids referenced in Conga Solutions.

## Conga Keys Setup

 Salesforce orgs with new installations of Conga Composer require no additional setup to use Conga Keys.


For information on setting up Conga Keys, see [Setting Up Conga Keys](#).

## Conga Keys Details

- The Key field is available these Composer objects: Conga Template, Conga Email Template, and Conga Query
- The Key field is automatically assigned to new records or when an existing record is updated or modified.
- Parameters in a Conga Solution can have a mix of both Conga Keys and record Ids referenced as parameter values.
- Conga Keys works across all related Sandbox and Production orgs.
- It is not required to use Conga Keys in Composer Solutions.

## Setting Up Conga Keys

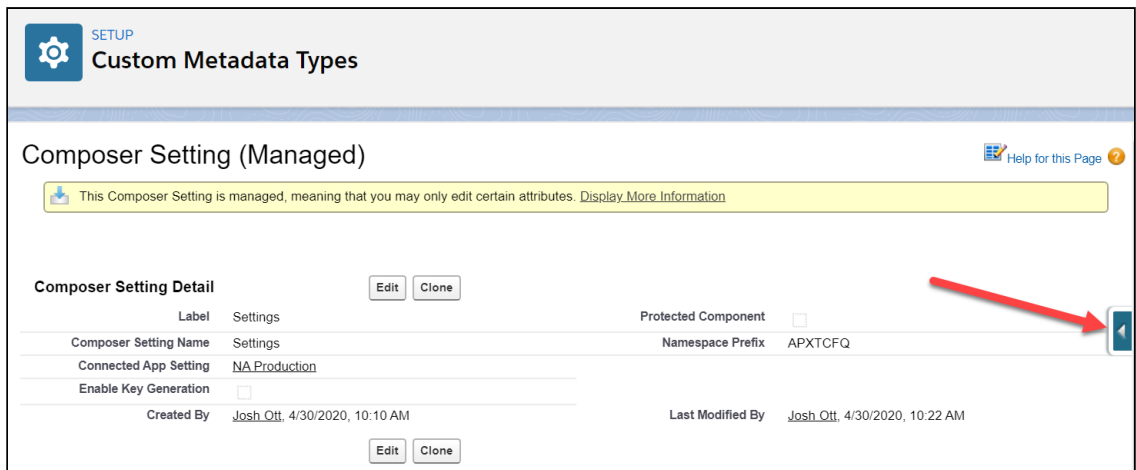
Setup Conga Keys in your Salesforce org.

 You must have Conga Composer version 8.186 (Quandary) or greater to use Conga Keys. The Conga Keys setup steps below are only required for Salesforce orgs with upgraded instances of Conga Composer. Salesforce orgs with new installations of Conga Composer do not need to complete any of the setup steps below.

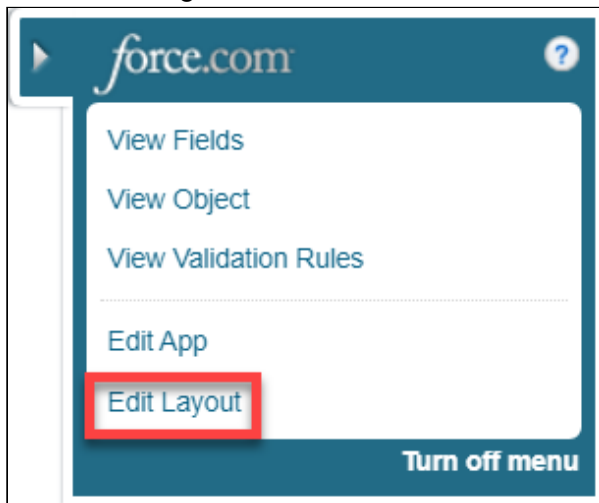
To setup Conga Keys:

1. Add the Key field to the Conga Template, Conga Email Template, and Conga Query object Page Layouts.
  - a. Navigate to Salesforce Setup.
  - b. Select **Object Manager** under the Object and Fields section.
    - In Salesforce Classic, select **Objects** under the Create section.
  - c. Click **Conga Template**.
  - d. Click **Page Layouts**.


- e. Click **Conga Template Layout**.
  - f. Select and drag the Key field onto the Information section of the Page Layout below.
  - g. Click **Save**.
  - h. Repeat steps **c - g** for the Conga Email Template and Conga Query objects.
2. Add the Enable Key Generation field to the Composer Setting Custom Metadata Type.
- a. Navigate to Salesforce Setup.
  - b. Type in Custom Metadata in the Quick Find search box and click **Custom Metadata Types**.
  - c. Click **Composer Setting**.
  - d. Click the Expand Quick Access Menu arrow to expand the menu.



- e. Click **Edit Layout**.




- f. Drag and Drop the Enable Key Generation field onto the Information section of the Page Layout below.
- g. Click **Save**.

 The Enable Key Generation checkbox field is checked by default.


3. Conduct record updates to existing Conga Template, Conga Email Template, and Conga Query records. Making edits to existing records automatically populates the Key field with a unique external Id. Use one of the following methods to trigger updates on existing Conga Template, Conga Email Template, and Conga Query records.
  - Edit any field on an existing record and save the changes. For example, double-click the Name field on a Conga Template to edit the field value and then click Save. No actual content editing of the field value is required. The save function triggers the Key field to automatically populate with a unique external Id. Repeat this step for each Conga Template, Conga Email Template, and Conga Query record you wish to use Conga Keys with.
  - Use and execute a SOQL Query in the Developer Console to make a mass edit on all Conga Template, Conga Email Template, and Conga Query records. No actual content edits to field values are made with this method, although it triggers the save function on each record and causes each record's Key field to automatically populate with an external Id.

4. Update Conga Solution records.

- Click the **Update Button** button on each Conga Solution record. This automatically updates each parameter value that references a Conga Template, Conga Email Template, and Conga Query record Id to use the new Key field's external Id instead. Each record Id is replaced with an external Id.

 You must complete Step 3 prior to this step in order to populate the Key field for each Conga Template, Conga Email Template, and Conga Query record.

- If you are not using the Solution Manager and created solutions manually in Salesforce Setup, manually swap each Conga Template, Conga Email Template, and Conga Query record Id referenced as a parameter value with the new corresponding external Id on the Key field.

 After completing the setup for Conga Keys, all new Conga Template, Conga Email Template, and Conga Query records will auto-populate in external Ids in the Key field.

## Create a Solution Collection

This documentation reviews Solution Collection functionality and how to create a Solution Collection.

### Conga Solution Collections

The ability to create Conga Solution Collections is a new feature for Conga Composer. You can group related Conga Solutions and easily launch them using the Composer Lightning Component, rather than selecting from multiple Lightning actions or components. For more information about the Composer Lightning Component, see [Composer Lightning Component](#).

When creating a new Conga Solution manually or with Conga Quick Start, you can add it to an existing collection of solutions or create a new Solution Collection.

### Create a Conga Solution Collection

To create a new Solution Collection with Conga Quick Start:

1. Navigate to the Lightning App Launcher and select **Conga Quick Start**.
2. Under the Document Automation Wizard, click **Launch Wizard**.
3. Navigate to the Collection Step.
4. Click **Create Collection**.
5. Enter a Collection Name.
6. Enter an optional brief Description.
7. Click the + sign next to the listed solution you want added to your new collection.
8. Click **Save**.

To create a new Solution Collection manually, take the following steps:

1. Navigate to the Lightning App Launcher and select **Conga Solution Collection**.
2. Click **Create Collection**.
3. Enter a Collection Name.
4. Enter an optional brief Description.
5. Click the + sign next to the listed solution you want added to your new collection.
6. Click **Save**.



## Update an Existing Solution Collection

To add a Conga Solution to an existing Solution Collection with Conga Quick Start:

1. Navigate to the Lightning App Launcher and select **Conga Quick Start**.
2. Under the Document Automation Wizard, click **Launch Wizard**.
3. Navigate to the Collection Step.
4. Under Available Solution Collection, click on the name of the collection where you want to add your new solution.
5. Click the + icon beside the solution you want to add to the existing collection.
6. Click **Save**.

To add a Conga Solution to an existing Solution Collection manually, take the following steps:

1. Navigate to the Lightning App Launcher and select **Conga Solution Collection**.
2. Click the name of the existing Solution Collection.
3. Click the + icon beside the solution you want to add to the existing collection.
4. Click **Save**.

## Gathering Data for Composer

Information on using Salesforce reports and Conga Queries to gather data for Conga Composer.

- [Data Gathering Basics](#)
- [Using Salesforce Reports with Conga Composer](#)
- [Using Conga Queries with Conga Composer](#)

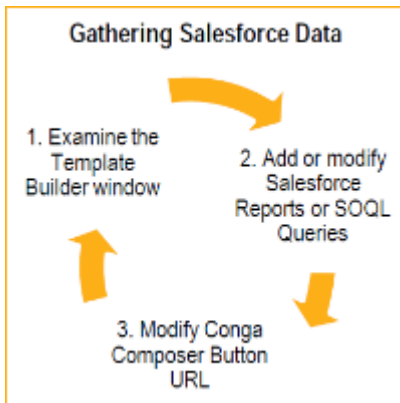
## Data Gathering Basics

Basic information about gathering data for Conga Composer.

- [Gather Salesforce Data](#)
- [Data Available for a Solution](#)
- [Adding an Alias](#)
- [Dynamically Filter Reports and Queries with pv Values](#)

## Gather Salesforce Data

The process of gathering the correct data from Salesforce is iterative. Generally, you start by examining the fields that Conga Composer® has automatically retrieved, which are displayed in the Master dataset. If those fields are sufficient for your solution, you're ready to build your template.



If not, you'll need to add additional Salesforce Reports or SOQL Queries as necessary to retrieve whatever data is required for your solution. For example, this additional data may include Related Lists or other data related to your Master Object, but not included in the Master Data (such as Contacts related to an Account).

For more information, see:

- [Using Salesforce Reports with Conga Composer](#)
- [Using SOQL Queries with Conga Composer](#)

After creating Salesforce reports and/or SOQL queries to retrieve your data, you add them to your solution using data gathering parameters.

**⚠ While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.**

## Data Available for a Solution

After you select the Master Object for your solution, the data from the master object is available within Template Builder. To gather information from Salesforce® that isn't contained in the Master Object (such as related lists), use:

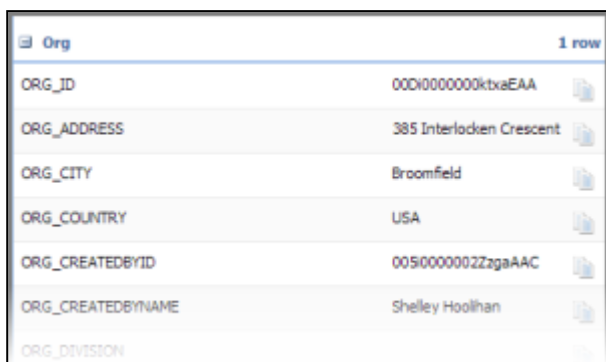
- Salesforce Reports
- SOQL Queries

The benefit of using Salesforce reports to gather additional data for Conga Composer is that creating reports and linking them to Conga Composer is quick and easy, but they are somewhat limited in the information you can access.

SOQL queries, on the other hand, are more powerful in that they can access most objects in Salesforce, but they're more complicated to set up.

## The Org Dataset

The Org dataset is displayed by default in the Conga® Template Builder to aid in template creation. This dataset contains fields with data about your company or organization, pulled from the Company Information section in Salesforce Setup.

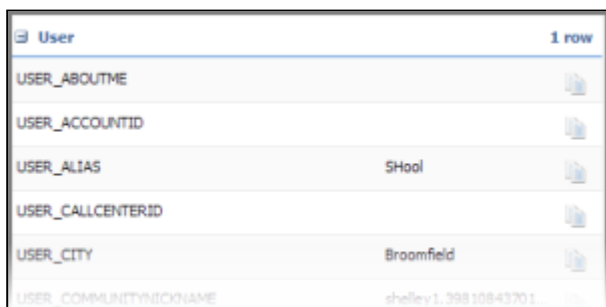


Org		1 row
ORG_ID	00D0000000ktxaEAA	
ORG_ADDRESS	385 Interlocken Crescent	
ORG_CITY	Broomfield	
ORG_COUNTRY	USA	
ORG_CREATEDBYID	0050000002ZzgaAAC	
ORG_CREATEDBYNAME	Shelley Hoolihan	
ORG_DIVISION		

Org dataset fields are used in templates that reference any company information. When used in templates as dynamic fields rather than static text, your template automatically collects the current company information as it generates.

## The User Dataset

The User dataset displays by default in the Conga Template Builder to assist in template creation. The User dataset displays fields from the User record for whoever is logged into Salesforce and has clicked the Composer button.

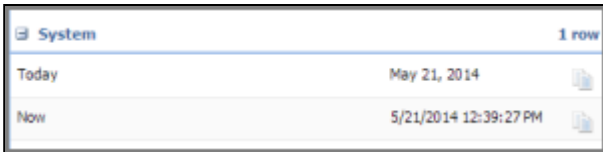


User		1 row
USER_ABOUTME		
USER_ACCOUNTID		
USER_ALIAS	SHool	
USER_CALLCENTERID		
USER_CITY	Broomfield	
USER_COMMUNITYCKNAME	shelley.1.39810843701	

These fields are particularly useful when referencing information about the user running the solution.

## The System Dataset

The System dataset is displayed by default in the Conga Template Builder to aid in template creation.



System		1 row
Today	May 21, 2014	
Now	5/21/2014 12:39:27 PM	

There are two fields available in the System dataset – Today and Now.

- Today will display today’s date.
- Now will display today’s date and time (at the time of merging).

Conga’s Today and Now will always display the date or time when the document was originally created whereas Microsoft Word® or Excel® date or time fields update to the current date/time each time the document is opened.

**Using 'System' Date fields in Excel requires the picture switch (numeric) in order to apply cell formatting.**

Example: `&=Date.Today(numeric)`

## Adding an Alias

The alias name becomes the dataset name in Template Builder. Alias names make it easy to identify data within the Template Builder as you create templates for your merged documents. The alias name becomes the dataset name in Template Builder and the name you reference in your template when creating detail regions to merge report or query data.

Report and query datasets are incorporated into the Composer button using the ReportId and QueryId parameters.

Report and query aliases must be enclosed in brackets, cannot contain spaces or special characters, and must be between 3 and 20 characters long. They precede the id of the report or query in the URL (see below). The dataset names of Master, Org, User, and ReportData cannot be used as aliases because they are reserved for the default datasets.

```
&Id={!Opportunity.Id}
&QueryId=[Account]a034600000ARPH5
```

Alias
ID

## Dynamically Filter Reports and Queries with pv Values

**⚠** While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

### What is a pv Value?

PV (or pv) stands for parameter value, which is a value that can be passed into a filter on a Salesforce report or Conga query. The first filter on a report or query is called pv0, the second pv1, the third pv2, and so on.

**i** A Composer solution can contain up to 6 pv values.

### What Does It Have to Do with Composer?

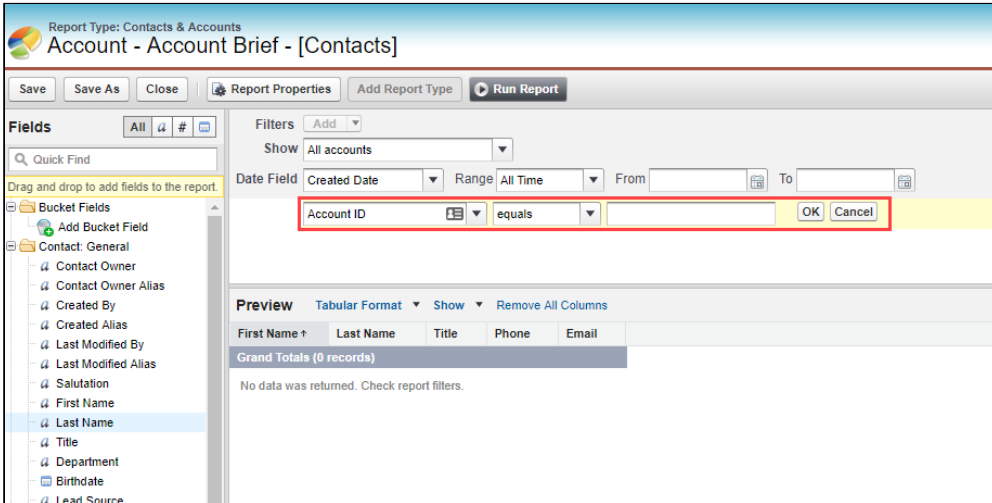
Conga Composer automatically passes the Id of the record from which you are launching Composer (known as the master object record Id) into the pv0 filter on your report or query (assuming you are using one). In this way, Composer is able to return data specific to the record you are working with.

For example, if you are gathering Contacts with a query, rather than returning ALL Contacts in your Salesforce org, Composer passes the Id of the master record into the pv0 filter on the query so that only Contacts related to the record you are working with are returned and merged into the document.

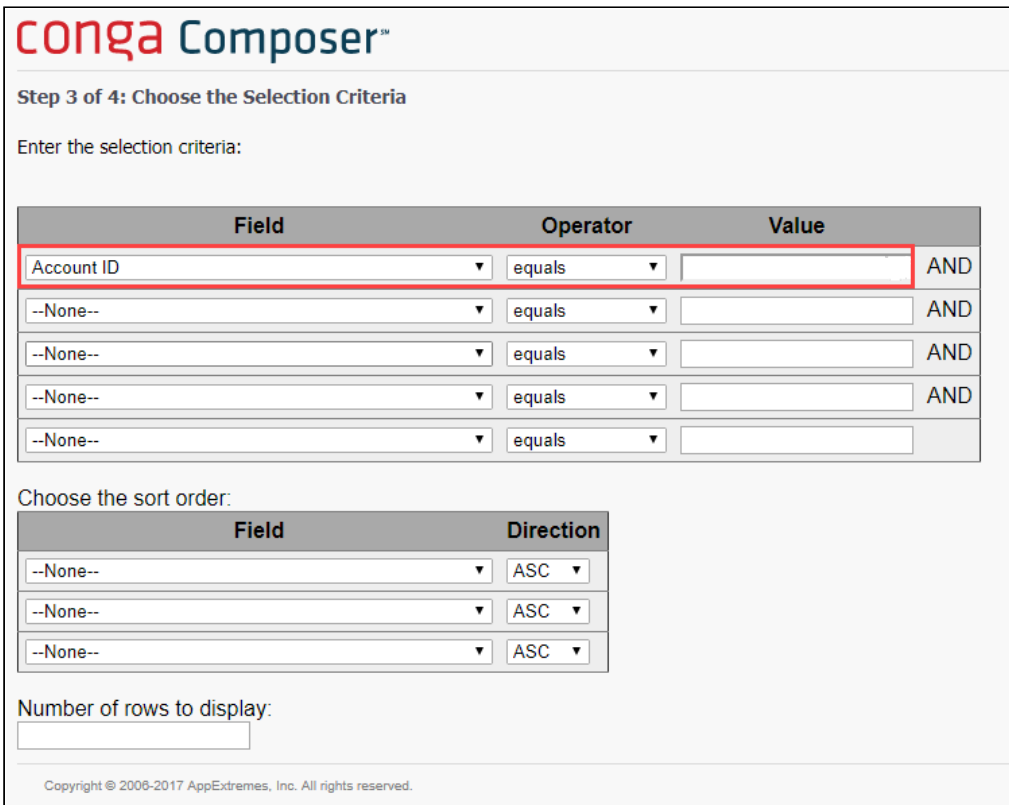
### How Does it Work?

Composer's default behavior is to pass the master object record Id into the pv0 filter. What you need to do is ensure your report or query has a blank filter to accept that value.

Using the example above, if you are retrieving Contacts with a report or query and your master object (the object on which you have built your Composer button) is an Account, you will need to add a filter for Account Id and leave it blank, like this:



If the master object is an Opportunity, you would add a filter for an Opportunity Id. If doing the same with a query, you will create this filter on Step 3 of the Conga Query Builder, and it will look like this:



The resulting SOQL statement looks like this (the WHERE clause filtering on pv0 is the equivalent of the filter in the report): `SELECT Name, Phone, Title FROM Contact WHERE AccountId = '{pv0}'`

## What Else Can I Do with pv Values?

In addition to the default behavior described above, you can use pv values to create advanced filter configurations to gather and merge data specific to your use case. These configurations include:

- Pass no value to pv0 (which is helpful when using Composer as a reporting tool to gather org-wide data)
- Pass a value other than the master object id into the pv0 filter
- Pass values to multiple filters (pv0, pv1, etc.)
- Pass multiple values to the same filter
- Pass a Null Value

The primary use case for passing a null value is when performing a Global Merge. In this case, you likely don't want to filter your reports on a specific record because you are conducting a record-independent merge.

To do so, append the report or query Id in your button URL with `?pv0=`

**i** If using the Conga Solution Manager, there is a field under Advanced Configuration on the Add Report or Add Query screen where you can add a "+" to the pv0 field, which will append the button URL for you and pass a null value.

## Pass a Value Other Than the Master Object Id

To filter pv0 on a value other than the master object Id, simply add the new field on which you want to filter to the URL. For example, if your master object is an Opportunity, but you want to filter your data source on the Account Id instead of Opportunity Id to get a list of Contacts from the Account, your URL would look like this: `?pv0={!Account.Id}or?pv0={!`

`Account.Id}`

**i** If using the Conga Solution Manager, there is a field under Advanced Configuration on the Add Report or Add Query screen where you can add the `{!Account.Id}` merge field as the pv0 value and it will update the URL for you.

**w** Do not pass through a field with special characters such as commas or ampersands, as that will cause issues with the URL.

## Pass Values to Multiple Filters

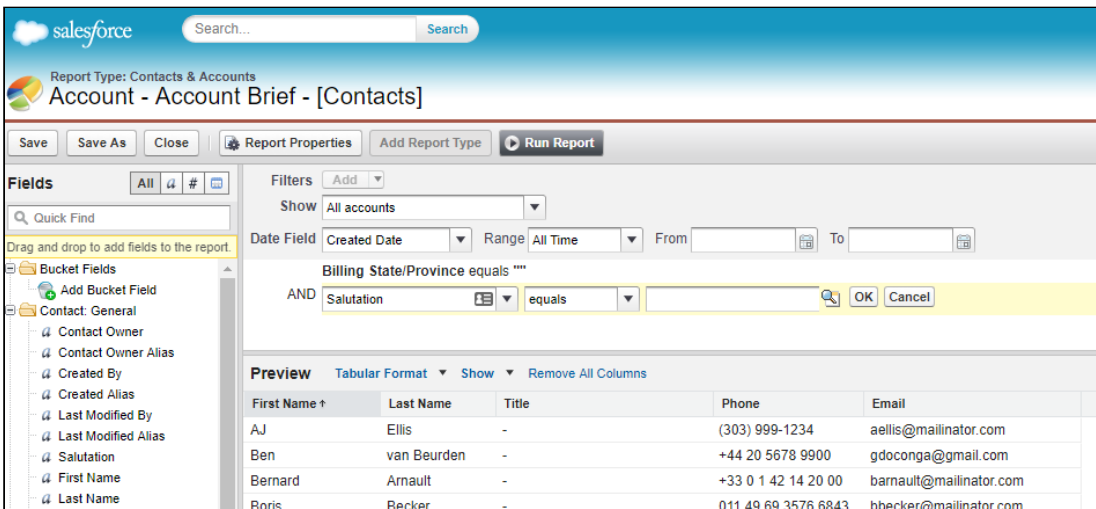
You may want to filter your data source on two or more criteria, such as filtering a list of Contacts on those in the same State/Province as the Account and who are also doctors. To

do so, you can pass multiple pv values by creating a report or query filter for each value you want to pass (pv0, pv1, and so on.) and appending the URL as follows (Note you must separate each pv value with a ~):

```
?pv0={!Account.BillingState}~pv1=Dr.or?pv0={!Account.BillingState}~pv1=Dr.
```

**i** If using the Conga Solution Manager, there is a field under Advanced Configuration on the Add Report or Add Query screen where you can add the pv1 value and it updates the URL.

For this scenario, your report would look like this (notice the second filter for the pv1 value of Dr.):



Your SOQL statement would look like this: `SELECT Name, Phone, Title FROM Contact WHERE Account.BillingState = '{pv0}' AND Salutation = '{pv1}'`

## Pass Multiple Values to the Same Filter

You can also pass multiple values into the same filter. Using the example above, if you want to return a list of Contacts who are in the same State or Province as the Account and who are either doctors or professors, you need to update the button URL by adding the additional value for the pv1 filter and separating it from the previous value with a "|" symbol, like this (note you don't need to change anything in the report or query because it already has a filter for pv1):

```
?pv0={!Account.BillingState}~pv1=Dr.|Prof.
```

**i** If using the Conga Solution Manager, there is a field under Advanced Configuration on the Add Report or Add Query screen where you can add the additional value and it updates the URL.



## Pass Multiple Values into a Query

You can pass multiple values into a query filter as well, however, this requires a slight change to the query syntax. Using the example Contacts query above we would have to modify the SOQL statement to look like this: `SELECT Name, Phone, Title FROM Contact IN{pv1})`

and the button url to: `-pv1='Dr.'|'Prof.'`


## Using Salesforce Reports with Conga Composer

Use Salesforce Reports to gather additional data for your Composer solution.

- [Use Salesforce Reports with Conga Composer](#)
- [Add Salesforce Reports to a Conga Composer Button URL Manually](#)
- [Best Practices: Report Naming Conventions](#)
- [Example: Use a Salesforce Report to Gather Information for an Account Plan](#)
- [Frequently Asked Questions About Using Salesforce Reports in Conga Composer](#)

## Use Salesforce Reports with Conga Composer

You can use a Salesforce report to retrieve data related to your Master Object. Conga Composer automatically filters the report by passing the Master Object ID into the report's Advanced Filters. (You may override the default behavior and instead use other information to filter the report as needed.)

 While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability. Please reference this [link](#) to the Conga Query Builder.

Conga Composer filters the report by passing the Master Object ID into the first cell of the value column in the report when the merge begins.

Ensure that the *High Assurance Session Required* setting is disabled, allowing Conga Composer to access exported report data. If this setting is enabled, the session security level required to export reports is increased, preventing Conga Composer to access the report data.

[Watch a video of this task.](#)

Let's say your Conga Composer button is located on an Account and you'd like to use a report to retrieve all Contacts related to that Account so you can print a phone directory.



1. Create a Conga Composer button and place it on your Account page layout.
2. Create a Salesforce report based on Contacts & Accounts.
3. Drag and drop the fields you're interested in from Fields to Preview.
4. In the Filters section, set the following:
  - a. Show: View drop-down list to All Accounts.
  - b. Date Field: Clear any values in Start Date and End Date.
  - c. **Required:** Add Field Filter as the first Advanced Filter on the report. Set field drop-down list to the ID of the Salesforce object on which you built your Composer button or link (i.e. Account ID), the operator remains as equals, and value remains blank.
  - d. Save the report into any publicly accessible folder, we recommend Conga Composer Reports.

This report is now ready to receive the Master Object ID (which is the Account ID) from Conga Composer.

Test your report by pasting sample data into the Advanced Filters and choose Run Report. Don't save the report with the sample data. (In the above example, you would paste an Account ID.)

Focus the scope of each Conga Composer report to pull only the objects and fields necessary. If you include an ID column in your report, position it as the last (right-most) column.


Conga Composer will retrieve a report that contains a maximum of 2,500 rows and contains less than 5 MB of total data. Additionally, each Conga Composer solution has a combined row limit of 25,000. You must use a Tabular format, not Summary or Matrix style reports.

## Add the Report to Your Composer Solution

After the report has been created, tested, and saved in the Conga Composer reports folder, you need to add the report to your solution. The best practice is to use a [Report Alias](#) in doing so.

You can [add the report to your solution manually](#), or if you created the button using Solution Manager, you want to add the report using Solution Manager.


After adding the report, update your button (if using Solution Manager) and then launch the button. When the screen loads open the Template Builder from the Tools & Settings drop-down menu. In the template builder, you should see a new dataset with the name of the alias you used (if no alias was used the dataset will be called ReportData).

 If you do not see the report dataset in the Template Builder that most likely means the report was not incorporated into the solution correctly. Review the button or link URL (specifically the &ReportID parameter) and launch the Template Builder again. If you see the dataset but it doesn't contain any data, then you need to check the report again. Likely culprits are accidentally saving a value in the Master Object ID filter (which is the first filter on the report - it should be left blank), not having a filter for the Master Object ID at all, or using the wrong Report Type.

## Add Salesforce Reports to a Conga Composer Button URL Manually

How to add Salesforce Reports to a Conga Composer Button URL. You can use existing Salesforce reports to gather the data for your merge document.

You can use existing Salesforce reports to gather the data for your merge document.

 While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

To add Salesforce reports to a button URL:

The button URL must exist and be open in edit mode. If not already open, click **Setup** → **Customize** → [Object] → **Buttons, Links, and Actions** and select the button you want to edit. In addition, all reports or queries must exist and have a Salesforce ID. (You can run the report or query to see the Salesforce ID.)

[Watch a video of this task.](#)

1. Click at the end of the button URL and press **ENTER** to begin a new line.
2. Type &ReportID=, the report alias (enclosed by brackets) and the report's Salesforce ID.
  - Type the parameter and value pair using the following format: `&[Parameter]=[value]`. Each parameter must include a valid value and

- may not include spaces. You can replace spaces in a value with the plus sign (+). Ensure that the report is visible to the end-user running the solution.
3. To add additional reports, separate each report alias and Salesforce ID with a comma.
  4. Click **Save**.

Test the button to ensure the correct data is retrieved.

## Additional Details


Alias names make it easy to identify data within the Template Builder as you create templates for your merged documents. The alias name becomes the dataset name in Template Builder.

Report aliases must be enclosed in brackets, cannot contain spaces or special characters and are limited to 20 total characters. The dataset names of Master, Org, User, and ReportData cannot be used as aliases because they are reserved for the default datasets.

Although Conga Composer supports up to 50 reports/queries, numerous or complicated reports can exceed the timeout of 60 seconds in Salesforce. There are other limitations that you should consider. For more information on limits, see [Product Data Limits](#).

## Best Practices: Report Naming Conventions

When naming reports created for use with Conga Composer, we suggest you adopt this naming convention.

 While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

When naming reports created for use with Conga Composer, we suggest you adopt a naming convention that reflects:

- The Master object for the Conga Composer solution the report is intended to support
- The name of the button or link to which the report belongs (Account Plan, Custom Invoice, etc.)
- The Report Alias - a description of the data being retrieved
- The Salesforce report type - to save space in the report name, this can be listed in the report's description field

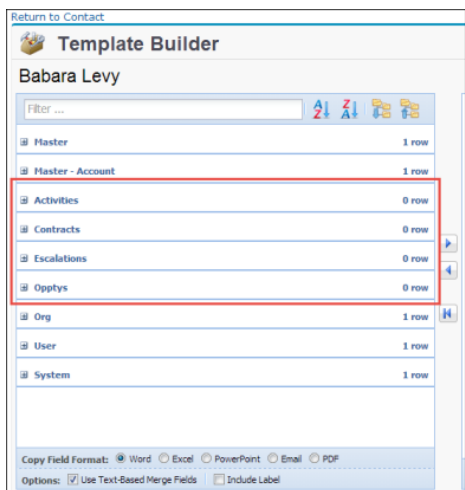
In the example below, four Salesforce reports called from a Conga Composer solution titled Executive Brief on a Contact page layout might be named as follows (note the Report Alias in brackets):

	<b>Contact - Exec Brief [Activities]</b> All activities for a given Contact.
	<b>Contact - Exec Brief [Contracts]</b> All Contracts for a given Contact.
	<b>Contact - Exec Brief [Escalations]</b> All Escalations for a given Contact.
	<b>Contact - Exec Brief [Opptys]</b> All Opportunities for a given Contact.

The Executive Brief button URL then looks as follows:


```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Contact Id}
&ReportId=[Activities]00OG0000005VR5B,
[Contracts]00OG0000005VR9T,
[Escalations]00OG0000005VQOg,
[Opptys]00OG0000005VR0y
```

By following this guidance, you will find it much easier to locate the Salesforce report responsible for the dataset revealed on the Template Builder window.



## Example: Use a Salesforce Report to Gather Information for an Account Plan

Conga Composer will automatically retrieve data from your master object, as well as specific related objects. To gather even more data for your Composer solution, you can use Salesforce Reports or SOQL queries.

 While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

### Example: Account Plan

Let's say you'd like to create an executive account plan that details all the open opportunities and their corresponding stage for any given account. You'll need a Salesforce Report to retrieve this data and make it available to Composer for merging. Since the Conga Composer button will be placed on the Account object, you'll need to create a report that dynamically retrieves data specific to whatever Account record a user might launch Composer from.

### Step 1: Create a Report with "Open" Filters

We'll start by building a report that returns all detail records - in our case, all Opportunities. This means there are not any filters applied to the report.

### Step 2: Add a Master Object ID Filter

Next, we'll add a filter for the master object ID. Set the field operator to "equals" but leave the first parameter value (pv0) blank. Conga Composer will automatically populate pv0 with the master record ID whenever the solution is launched from the custom button. This is the key to making this report dynamic - it will only return the detail data related to the master record from which Composer was launched.

### What are Parameter Values?

Parameter values represent the values of filter criteria on Salesforce Reports. Parameter value numbering begins at 0, so the first filter on a report is pv0, the second is pv1, and so forth.

## Create the Report:

Instructions	For the Account Plan Example, Use:
1. Create a Conga Composer Button and add it to your page layout.	<ul style="list-style-type: none"> <li>• Master Object = Account</li> <li>• Add button to the Account Page Layout</li> </ul>
2. Create a new report	Opportunity Report
3. Click Remove All Columns	
<p>4. Add back only the fields you would like Composer to retrieve to your report.</p> <p>Tip: Focus the scope of each report to pull only the object and fields necessary.</p>	Opportunity Name, Stage, Probability
5. Filter the report to retrieve all possible data that meets your desired criteria for the record.	<p>To Retrieve all open opportunities -</p> <ul style="list-style-type: none"> <li>• Show: All Opportunities</li> <li>• Opportunity Status: Open</li> <li>• Probability: All</li> <li>• Range: All Time</li> </ul>
<p>6. Add a Field Filter for:</p> <ul style="list-style-type: none"> <li>• Field: Master Object ID</li> <li>• Field Operator: equals</li> <li>• pv0: leave blank</li> </ul>	<ul style="list-style-type: none"> <li>• Field: Account ID</li> <li>• Field Operator: equals</li> <li>• pv0: leave blank</li> </ul>
<p>7. Save the report with:</p> <ul style="list-style-type: none"> <li>• Name: Recommend following Conga® naming conventions</li> <li>• Folder: Conga Composer Reports</li> </ul>	<ul style="list-style-type: none"> <li>• Name: Account - Account Plan - [Opportunities]</li> <li>• Folder: Conga Composer Reports</li> </ul>

## Test the Report

Test your report by inserting your master object ID into the pv0 field and run the report. In our example, we would insert an Account ID into the pv0.

**⚠ Make sure you delete your ID value in the pv0 after testing! If you save the Report with a hardcoded ID as a filter value, your Report will not be dynamic.**

**ℹ Conga Composer will retrieve a report that contains a maximum of 2,500 rows and contains less than 10 Mb of total data. Additionally, each Conga Composer solution has a combined row limit of 25,000 across all datasets.**

## Frequently Asked Questions About Using Salesforce Reports in Conga Composer

Answers to frequent questions about using Salesforce® reports with Conga Composer.

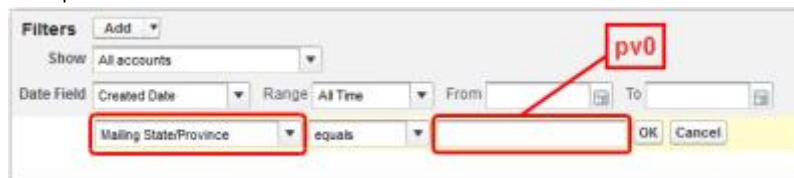
**⚠ While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.**

### How Can I filter the Report on Something Other Than the Master Object Id?

Although Conga Composer will use the Master Object ID as the default filter value in the report’s Advanced Filters, you may override this behavior and use any field from the master object. You can also use a field from a parent of the master object, like an Account field related to a Contact.

To use a different field, you’ll need to do the following:

- Configure your report to receive the desired value.
- Update your Conga Composer button to send the desired value.
- Configure your report to receive some other field value:
  - a. Edit the report.
  - b. In the Filters section, add a field filter and select the field you’d like to use in the field drop-down list.



- c. Leave the Value field empty (This cell is known as pv0).
- d. Save the report. Let’s extend our example from earlier. Instead of creating a directory of Contacts at a given Account, let’s say we want all Contacts who



are in the same state as the Account (but not necessarily affiliated with the Account). In the image above, the Account ID field has been replaced with Contact Mailing State/Province so that we can search for Contacts who are in the same State/Province as the Account.

Update your Conga Composer button to pass the desired value to the report:

1. Edit the Conga Composer button: **Setup** → **App Setup** → **Customize** → [Object] → **Buttons and Links** → **Edit**
2. At the end of the URL, add `?pv0=` to the ReportID parameter.

```
https://composer.congamerger.com
?SessionId={!API.Session_ID}
&ServerUrl={!$Api.Partner_Server_URL_290}
&Id={!Account.Id}
&ReportId=[ContactsAcctState]00OG0000006jajb?pv0={!Account.BillingState}
```

When using `?pv0=`, the pv must be lowercase. Do not use fields that contain commas or ampersands to filter reports, as these interfere with report results.

3. Select the Billing State/Province field from the Insert Merge Field drop-down list.

## How Can I Filter a Report with Criteria From Two or more Fields?

For even more demanding situations, you can pass a second, a third and even as many as five parameters to the same report. The values for those parameters are numbered sequentially: pv0, pv1, pv2, etc. Each pv parameter must be separated with a tilde symbol (~). The pv in the pv parameter must be lowercase letters.

Use ID fields in filters instead of name fields to ensure correct results. When passing numeric date values to a report, you must pass them in the US date format, mm/dd/yyyy. You can ensure the correct format by using a Salesforce formula field to parse a Date field. Extending the example from above, let's say we want a list of all contacts in the same State/Province as the account and who are also doctors.

The selection criteria of the report would need to be:

The ReportID parameter in the Conga Composer button URL would use pv1 to filter appropriately:

```
https://composer.congamerger.com
?SessionId={!API.Session_ID}
&ServerUrl={!$Api.Partner_Server_URL_290}
&Id={!Account.Id}
&ReportId=[ContactsAcctState]00OG0000006jajb?pv0={!Account.BillingState}~pv1=Dr.
```

## How Can I filter the Report with Two or More Values from One Field?

You may use multiple values in a single pv value by separating each value with a pipe symbol (|).

Again, extending the previous example, let's say we want a list of all contacts in the same State or Province who are either doctors or professors. In this case, the report is already ready to receive the filter criteria. We only need to change the ReportID parameter of the Conga Composer button by updating pv1.

```
https://composer.congamerger.com
?SessionId={!API.Session_ID}
&ServerUrl={!$Api.Partner_Server_URL_290}
&Id={!Account.Id}
&ReportId=[ContactsAcctState]00OG0000006jajb?pv0={!Account.BillingState}~pv1=Dr.|Prof.
```

## My Report Already has Filter Criteria Under Advanced Filters. Can Conga Composer Filter the Report Using Subsequent Rows in Advanced Filters?

Yes - If you want Conga Composer to filter your report with a value in, say, the pv1 position, you may define the ReportId parameter in your button to use any combination of pv positions.

Let's say your report has filter criteria in the first row of the Advanced Filters (i.e., the pv0 position) and you want Conga Composer to supply filter criteria for the second row (pv1). You could specify that as:

```
&ReportId=[ContactsWithAcctState]0005000000001LNBd?pv1={!Account.BillingState}
```

## What If I Don't Want to Dynamically Filter the Report? I Just Want the Report Contents As-Is.

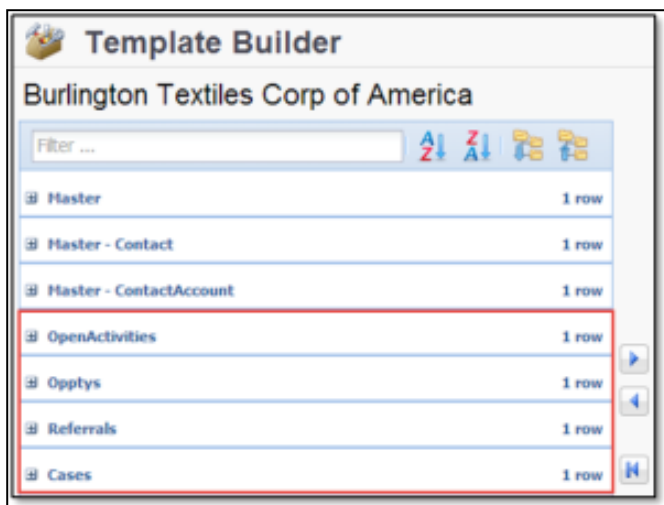
You can retrieve the contents of a report without applying any dynamic filtering by passing an empty value for pv0:

```
&ReportID=[StaticContactList]0005000000001LNBd?pv0=
```

## How Can I Gather Data from Multiple Reports?

You may include up to 50 ReportID parameters in a Conga Composer URL, thus making 50 reports accessible in a single merge operation. To do so, enter a comma at the end of the first report ID parameter, followed by the next report ID as shown below, followed by its comma, and so on. If a report alias is not included, Conga Composer will assign dataset names based on the sequence of report IDs included in the button URL (the first report ID has a dataset name of ReportData, the second is ReportData1, and so on).

Do not include spaces anywhere in your Conga Composer button URL. Here, we have Conga Composer configured to pull data from four reports:



```
https://composer.congamerge.com?Sess...PI.Session_ID}
&ServerURL={!API.Partner_Server_URL_290}
&Id={!Contact.Id}
&ReportID=[OpenActivities]00O50000001NGuW,
[Opptys]00O50000001O2gr,[Referrals]00O50000001NGv5?pv1={!Contact.MailingState},
[Cases]00O50000001NGvL
```

Each of these reports would be visible through the Template Builder window as a separate dataset. Conga Composer automatically assigns the worksheet names based on the report alias specified.

## How Can I control the Sort-Order of the Data Retrieved from a Report?

You can control the sort-order of the information Conga Composer retrieves from a report:

1. From the Reports Tab, run the Report by clicking its name.
2. Click the column-header label of the desired column to sort by.



3. Save the report.

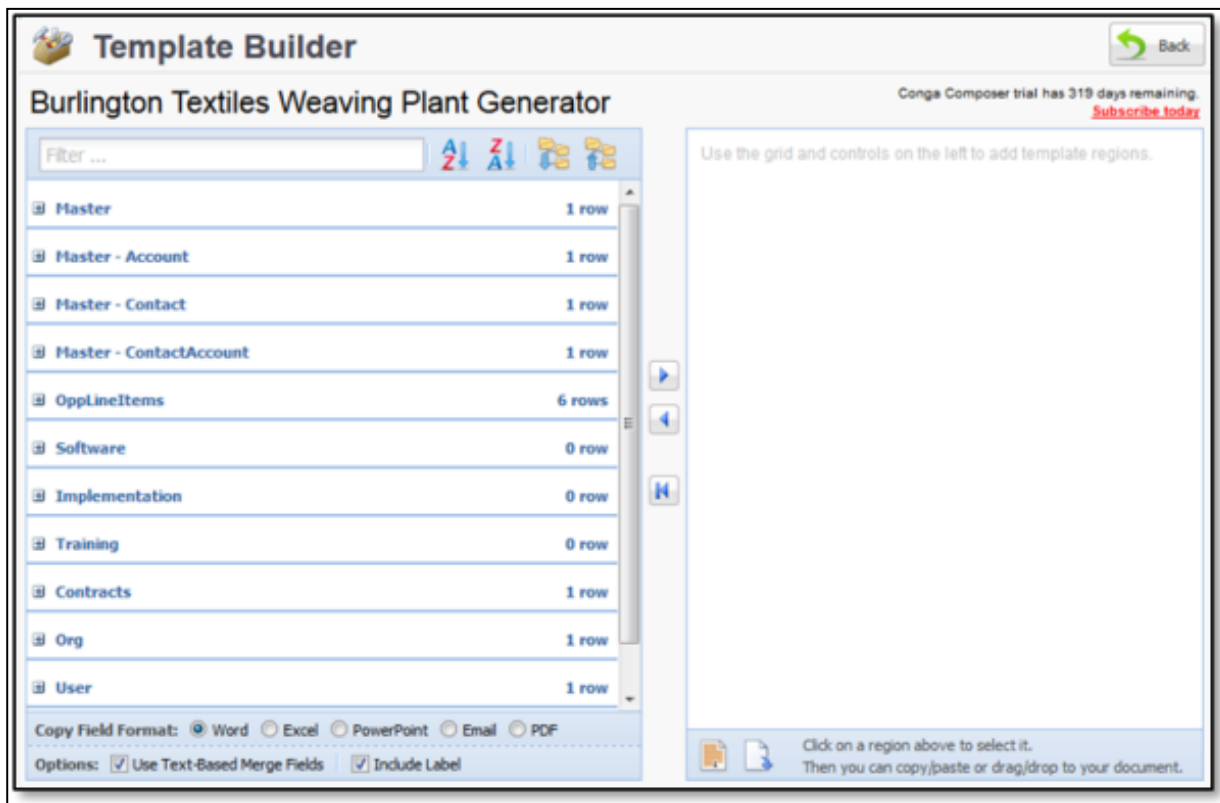
Say your report shows a list of Contacts related to an Account and you'd like the list of contacts sorted by Last Name. To set the sort order, you would run the report, click on the Last Name column (even though the report may have zero rows of information currently on it), then save the report.

When Conga Composer interacts with the report, the data it gathers is sorted according to your sort-order choice. You can sort on multiple columns by changing the report to a Summary Report and summarizing on another field. Conga Composer doesn't use the summary information but will adhere to the sort order of the report.

## I'm Creating Quotes (or Invoices) from an Opportunity and Its Related Opportunity Line Items. Do I Need a Report to Retrieve the Opportunity Line Items?

No, in this particular case, you don't. Conga Composer automatically makes Opportunity Line Items available to you without any reports because the master object is an Opportunity.

Opportunity Line Items appear as a separate dataset (OppLineItems) in the Template Builder window:



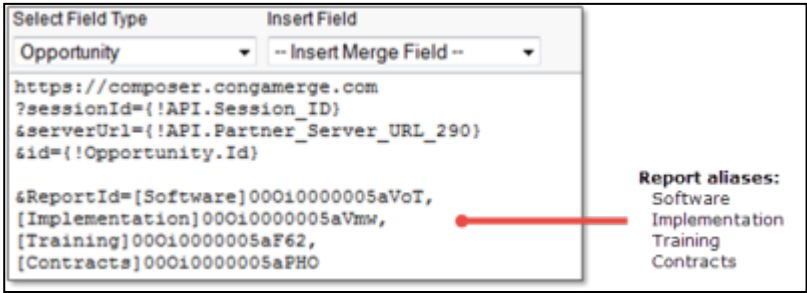
For more information, see [About Data in the Template Builder](#).

## How Can I Use My Own Label for a Dataset?

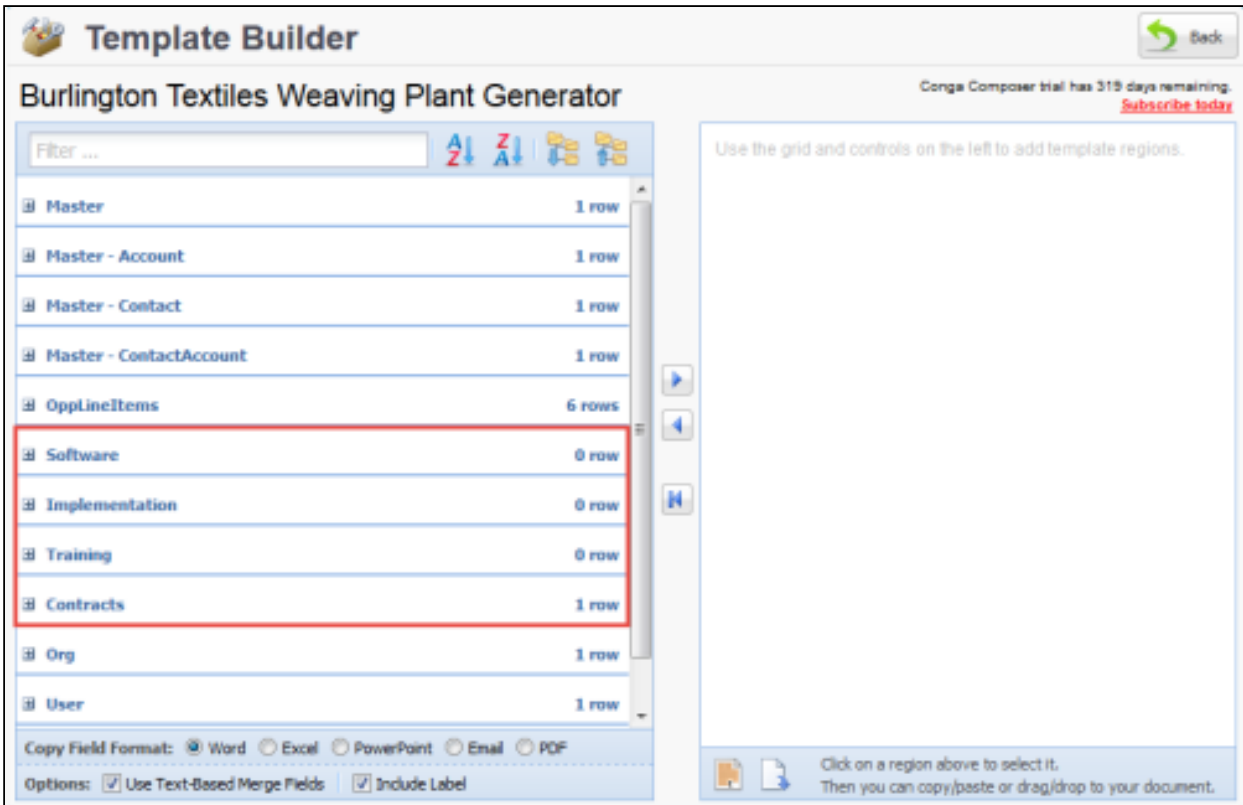
You may supply an alias for a Report (or Query) to override the default name. Please follow these guidelines:

- The alias must precede the Salesforce® Report (or Query) Id.
- It must be enclosed in square brackets ([ ]).
- It may contain only alphanumeric characters; punctuation marks and other special characters are not permitted.
- It must contain at least 2 characters and but not more than 20 characters.
- It must be unique within the entire Conga Composer URL.
- Spaces are not permitted within an alias, or between the brackets and the Report Id.
- Reserved sheet names (Master, Org, User, ReportData, etc.) may not be used as an alias.

In this example, an alias has been added to each Report Id.



The resulting datasets in the Template Builder window appear as (following the same order they are entered in the button URL):



The field names in your template must then refer to the alias name. For example, in Microsoft Word would you create a field for `{{TableStart:Contracts}}` and `{{TableEnd:Contracts}}`

## Using Conga Queries with Conga Composer

How to gather data for Composer solutions using Conga Queries.

- [About Conga Query Manager](#)
- [Advanced Conga Queries](#)
- [Conga Query Library](#)

- [Conga Query Manager Limits](#)
- [Create A Conga Query With Conga Query Builder](#)
- [FILE is the related List or in Attachments and you need an IMAGE](#)
- [Retrieving a document stored in a Salesforce file](#)
- [Using SOQL Queries with Conga Composer](#)

## About Conga Query Manager

Conga Query Manager supercharges your ability to gather Salesforce data by using SOQL (Salesforce Object Query Language) SELECT statements. Furthermore, Conga Composer extends SOQL's flexibility and power through the use of *pv* values. Conga Queries allow you to take your Conga Composer solutions to new heights.

## What Are Conga Queries?

Conga Queries are Salesforce records that contain a SOQL statement specifically tuned to work in conjunction with Conga Composer.

SOQL allows Composer to access your Salesforce data and merge it into your documents. Composer can access fields on your master object as well as records on parent records and related lists.

## Are Conga Queries Difficult to Write?

Crafting SOQL statements can be difficult depending on the complexity of your solution; however, in most cases, you can use the Conga Query Builder to create queries using a point and click interface that is as simple as building a Salesforce Report.

## (Legacy Functionality) Why Not Use Salesforce Reports?

In older versions of Conga Composer, solutions were built with Salesforce Reports instead of Conga Queries.

Going forward, Conga recommends that all solutions use Conga Queries, as they have several advantages over Salesforce Reports:

- Queries are slightly faster and more reliable than Salesforce Reports.
- Queries can access objects that cannot be displayed in Salesforce Reports (Notes, Approval History).
- Queries reduce clutter on the Reports Tab.
- Queries have tighter security and visibility controls than Salesforce Reports.

**⚠** Due to the almost unlimited customizability of SOQL statements, Technical Support Team will only support SOQL queries generated through Conga Query Builder or written by following one of our support articles. The technical support team does not support custom-written queries.

## Advanced Conga Queries

After creating a query with the Conga Query Builder, you may edit the SELECT statement (in the Salesforce record) to add SOQL features that aren't currently available from the Query Builder.

A popular reason to do so is to include one or more of the new SOQL aggregation functions such as SUM(), COUNT(), COUNT\_DISTINCT(), AVG() and GROUP BY which allow you to group and summarize a collection of records.

Once a query has been manually edited, do not use the Query Builder again on that record because the manual edits are overridden. Use Select Count(ID) from the Opportunity

Warning: Due to the complicated nature of SOQL statements, our support staff can only assist with the creation of SOQL statements generated through our tools, such as Conga Query Builder.

Please consult the [Salesforce SOQL documentation](#) for complete details about the aggregation functions.

## Product Family Example

The following example returns, for each unique Product Family value:

- The sum of Unit Prices
- A count of records
- The average of Unit Price values
- `SELECT PricebookEntry.Product2.Family, SUM(UnitPrice), COUNT(UnitPrice), AVG(UnitPrice) FROM OpportunityLineItem WHERE Opportunity.Id = '{pv0}' GROUP BY PricebookEntry.Product2.Family`

For a given Opportunity, this query results in the following values in the Conga ViewDataWorkbook.xls file:

A	B	C	D
OPPORTUNITYLINEITEM_FAMILY	OPPORTUNITYLINEITEM_EXPR0	OPPORTUNITYLINEITEM_EXPR1	OPPORTUNITYLINEITEM_EXPR2
Implementation Services	14500	1	14500
Training Services	1777.11	3	592.37
Software	13000	2	6500



## Use Aliases with Aggregate Functions

As displayed above in the Conga ViewDataWorkbook.xls file, the default field names do not clearly identify the source of the values displayed in columns B, C and D. To address this, add to the query a preferred "alias" that will replace the string "EXPRx" and render easily-identifiable field names.

```
SELECT PricebookEntry.Product2.Family SUM(UnitPrice) MyUnitPrice
COUNT(UnitPrice) MyCount AVG(UnitPrice) MyAverageFROM OpportunityLineItemWHERE
Opportunity.Id = '{pv0}'GROUP BY PricebookEntry.Product2.Family
```

For a given Opportunity, this query results in the following values in the Conga ViewDataWorkbook.xls file:

A	B	C	D
OPPORTUNITYLINEITEM_FAMILY	OPPORTUNITYLINEITEM_MYUNITPRICE	OPPORTUNITYLINEITEM_MYCOUNT	OPPORTUNITYLINEITEM_MYAVERAGE
Implementation Services	14508.0	1	14508.0
Training Services	1777.11	3	592.37
Software	13808.0	2	6904.0

## More Powerful Select Statements: Semi-Join and Anti-Join Support

Semi-Joins use the IN (...) operator within a WHERE clause to use the result of one SELECT statement as the search criteria for another SELECT.

Anti-Joins use the NOT IN (...) operator.

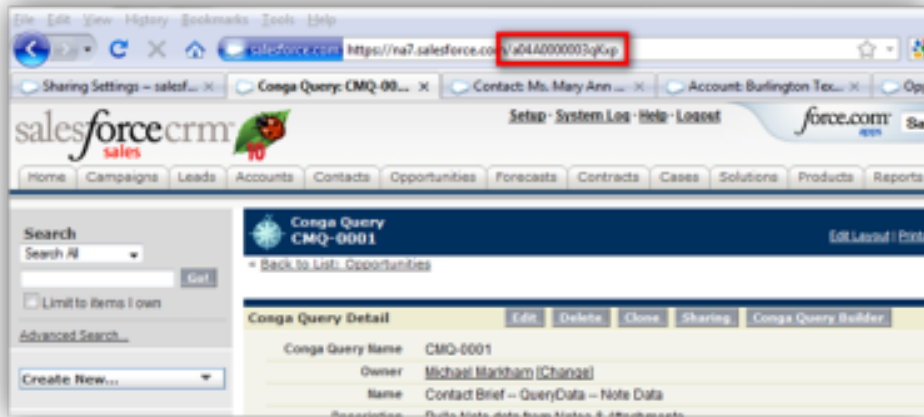
- Example 1:
  - SELECT Id, NameFROM AccountWHERE Id IN(SELECT AccountId FROM Opportunity WHERE StageName = 'Closed Lost')
- Example 2:
  - SELECT IdFROM AccountWHERE Id NOT IN (SELECT AccountId FROM Opportunity WHERE IsClosed = false)

## Include a Conga Query in a Conga Composer Button or Link

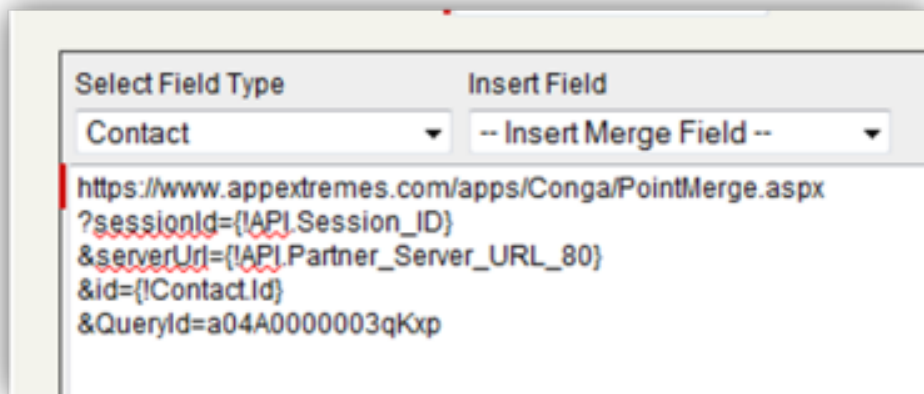
The final step of the process is to include the Id of the Conga Query record in your Conga Composer® button or link.

Here's how:

1. Copy the Salesforce Record Id of the Conga Query record.



2. Edit your Conga Composer button. Append the &queryId= parameter, and paste the record ID.



3. (Optional) Include a "Query Alias":  
You can supply an alias for a Report (or Query) to override the default name. Follow these guidelines when adding an alias:
  - The alias must precede the Salesforce Query ID.
  - It must be enclosed in square brackets ("[" and "]")
  - It may contain only alphanumeric characters; punctuation marks and other special characters are not permitted
  - It must contain at least 2 characters and but not more than 20 characters
  - It must be unique within the entire Conga Composer URL
  - Spaces are not permitted within an alias, or between the brackets and the Report Id
  - Reserved sheet names ("Master", "Org", "User", "ReportData", etc.) may not be used as an alias
  - Example: &QueryId=[MyContacts]a04A0000003qKxp

- The resulting sheet in the View Data Workbook appears labeled as “MyContacts” and would be referenced with this alias in TableStart and TableEnd fields.
4. (Optional) Specify the values to pass into the Conga Query. In the absence of a specific value, Conga will automatically pass the master object id into the pv0 field.
  5. Save the button. Navigate to a Salesforce record that displays the button and click the button. Click the View Data link to see the results.
  6. Add the newly available merge fields to your template.

## Conga Query Library

Below are sample queries you can copy and use for your solutions.

### Attachments

- Retrieve Attachments as Templates or Email Attachments
  - The query collects the attachments from the record to be used as templates
  - `SELECT Id FROM Attachment WHERE ParentId = '{pv0}' AND (Not(Name LIKE '%jpg%')) AND (Not(Name LIKE '%jpeg%')) AND (Not(Name LIKE '%png%'))`
- Get Attachments as Images
  - The query collects the image attachments from the selected record. The images can be merged into your template.
  - `SELECT Id FROM Attachment WHERE ParentId = '{pv0}' AND (Name LIKE '%png%' OR Name LIKE '%jpg%' OR Name LIKE '%jpeg%' OR Name LIKE '%tiff%')`
- Query Attachments for Images (Content Type)
  - `SELECT Id FROM Attachment WHERE ParentId = '{pv0}' AND ContentType IN ('image/bmp', 'image/gif', 'image/jpeg')`
- Select Attachment Images from Multiple Parent Records
  - The query collects image attachments from one or more parent IDs.
  - `SELECT Id FROM Attachment WHERE ParentId IN ({pv0}) AND (Name LIKE '%png%' OR Name LIKE '%jpg%' OR Name LIKE '%jpeg%')`
- Select Attachments Two Levels Down
  - The query collects attachments from contact records related to the selected Account.
  - `SELECT Id FROM Attachment WHERE ParentId IN (SELECT Id FROM Contact WHERE AccountId = '{pv0}')`
- Select Attachment from Contact in a Campaign

- Selects the ID from Contacts that are part of a specific campaign where the length of the attachment body exceeds a specific number of characters.
- `SELECT Id, BodyLength FROM Attachment  
WHERE parentid IN (SELECT contactid FROM campaignmember  
WHERE campaignid = '701o0000000JrJX') AND bodylength > 15000`
- Select Files Stored as Attachments in Salesforce
  - Select attachments stored as Files in Salesforce.
  - Requires the use of a QVAR (query variable) to pass the result of the query into the value of the TemplateID parameter. The below query and button URL will include the file in the output (along with whatever merged documents were generated).
  - Files Related List Query: `SELECT ContentDocument.id FROM ContentDocumentLink WHERE LinkedEntityID = '{pv0}'`
  - Button URL syntax: `&Qvar0ID=[id of Conga Query (above)]&TemplateID={QVAR0}`

## Files

- Retrieve an image from Salesforce Files
  - `SELECT ContentDocumentId, ContentDocument.FileExtension FROM ContentDocumentLink  
WHERE LinkedEntityId = '{pv0}'  
AND (ContentDocument.FileExtension LIKE '%jpg%'  
OR ContentDocument.FileExtension LIKE '%png%'  
OR ContentDocument.FileExtension LIKE '%gif%')`

## Opportunity Queries

- Retrieve all Opportunities for an Account
  - `SELECT Amount, CloseDate, Name, Type, Probability, StageName FROM Opportunity WHERE AccountId = '{pv0}'`
- Product Summary Query by Family
  - Retrieves all products from a Opportunity, groups by family, and sums the total price for each family.
  - `SELECT PricebookEntry.Product2.Family, SUM(Quantity) Qty, SUM(TotalPrice) Total  
FROM OpportunityLineItem WHERE OpportunityId = '{pv0}'`

- ```
GROUP BY PricebookEntry.Product2.Family
ORDER BY PricebookEntry.Product2.Family ASC
```
- Date Functions with Opportunity Query
    - The query demonstrates how to use a calendar year filter.
    - `SELECT Campaign.Name, SUM(Amount), MAX(Amount), AVG(Amount)`  
`FROM Opportunity WHERE Campaign.Name != ''`  
`AND Calendar_Year(CloseDate) = THIS_YEAR GROUP BY Campaign.Name`  
`ORDER BY Campaign.Name`
  - Date Query with Calendar Year Filters
    - The query retrieves data from last year and this year using calendar year filters.
    - `SELECT Campaign.Name FROM Opportunity WHERE Campaign.Name != ''`  
`AND (Calendar_Year(CloseDate) = LAST_YEAR`  
`OR Calendar_Year(CloseDate) = THIS_YEAR)`  
`GROUP BY Campaign.Name ORDER BY Campaign.Name`
  - Date Query with Calendar Year Filters
    - The query retrieves opportunity fields and groups based on calendar month of the close date.
    - `SELECT SUM(Amount), CALENDAR_MONTH(CloseDate)`  
`FROM Opportunity GROUP BY CALENDAR_MONTH(CloseDate)`
  - Primary Contact Role from Opportunity
    - `SELECT Id FROM OpportunityContactRole WHERE opportunityid = '{pv0}'`
  - Aggregated Opportunity Query (Sum)
    - The query retrieves an aggregated Amount field from Opportunity and groups the result set by Name.
    - `SELECT Name, Sum(Amount) FROM Opportunity WHERE id = '{pv0}' GROUP BY`  
`Name`
  - Aggregated Opportunity Query (COUNT)
    - The query retrieves all opportunities, counts the total records, and then groups by StageName.
    - `SELECT StageName, COUNT(Id) FROM Opportunity GROUP BY StageName`
  - Aggregated Opportunity Query
    - The query retrieves Opportunity Line Item info from the selected opportunity record and provides a count of unique record values in the results set.
    - `SELECT COUNT_DISTINCT(PricebookEntryId) FROM OpportunityLineItem`  
`WHERE OpportunityId = '{pv0}'`

## Contact Queries

- Query with Limit
  - The query retrieves the first fifty records of the results set.
  - `SELECT ID, FirstName, LastName FROM Contact ORDER BY Last Name ASC LIMIT 50`
- Query with Limit and Offset
  - The query retrieves the second fifty records of the results set.
  - `SELECT ID, FirstName, LastName FROM Contact ORDER BY Last Name ASC LIMIT 50 OFFSET 50`
- Query with Limit and Offset, Ascending Order
  - The query retrieves a specific number of records in the position specified by the values passed through the pv1 and pv2 filters.
  - `SELECT ID, FirstName, LastName FROM Contact WHERE AccountId = '{pv0}' ORDER BY Last Name ASC LIMIT {pv1} OFFSET {pv2}`
- Query to Retrieve Data from Chatter
  - The query retrieves data from Chatter linked to the selected record.
  - `SELECT ContentDocumentId FROM ContentDocumentLink WHERE LinkedEntityId = '{pv0}'`
- Semi-Join to Select Images from Chatter
  - `SELECT LatestPublishedVersion.Image_URL_Content_Formula__c FROM ContentDocument WHERE LatestPublishedVersionId IN (SELECT RelatedRecordId FROM AccountFeed WHERE ParentId = '{pv0}')`

## Logic Queries

- Using AND or OR Logic
  - The query retrieves fields from an Account record where the annual revenue is over \$1 million and the Name matches one of two values.
  - `SELECT ID, Name, AnnualRevenue FROM Account WHERE annualrevenue > 1000000 AND (Name Like 'B%' OR Name Like 'E%')`
- Multiple OR values using IN Clause
  - The query uses multiple OR values in query by using the IN clause.
  - `SELECT Id FROM Case WHERE use_case__c IN ('Quotes', 'Contracts', 'Proposals')`

## Nested Queries

- Nested Select Query
  - The query retrieves the primary Contact Role and all of the quotes from the Opportunity record.
  - ```
SELECT Id, (SELECT IsPrimary, Role, Contact.Name FROM OpportunityContactRoles WHERE Role = 'Technical Buyer'), (SELECT GrandTotal, LineItemCount, OpportunityId FROM Quotes) FROM Opportunity WHERE Id = '{pv0}'
```
- Nested Select with Semi-Join from Child
  - The query retrieves all of the contacts and custom object records based on filter criteria that are related to a particular contact record.
  - ```
SELECT Id, (SELECT AccountId, AssistantPhone, CleanStatus, Account.Name FROM Contacts WHERE Title LIKE '%facilities%'), (SELECT Account__c, Biceps__c FROM Custom_Objects__r WHERE Biceps__c = 'Jacked') FROM Account WHERE Id IN (SELECT AccountId FROM Contact where id IN ('{pv0}'))
```
- Semi-Join Query with Filter on Child
  - The query retrieves contacts that are related to the same account as the selected opportunity.
  - ```
SELECT Id, name FROM Contact WHERE AccountId IN (Select AccountId from Opportunity where stagename = 'closed won' )
```

## Work.com Queries

- Feedback Request Query
  - The query retrieves fields from the [work.com](#) feedback request object.
  - ```
SELECT Id FROM WorkFeedbackRequest WHERE PerformanceCycleId IN (SELECT Id FROM WorkPerformanceCycle WHERE Name = 'PSC 2014')
```
- Performance Cycle Query
  - The query retrieves fields from the [work.com](#) performance cycle object.
  - ```
SELECT Id,Name,CreateDate FROM WorkPerformanceCycle WHERE Name = 'PSC 2016'
```
- Approval Process Query
  - The query retrieves approval information from the selected record.

- `SELECT ActorId, Comments, CreatedById, CreatedDate, OriginalActorId, ProcessInstanceId, Id, StepStatus, SystemModstamp FROM ProcessInstanceStep WHERE ProcessInstanceId IN (SELECT Id FROM ProcessInstance WHERE TargetObjectId = '{pv0}')`

## Account Brief Queries

- Retrieve all Cases for an Account
  - `SELECT Reason, Type, IsClosed, ClosedDate, SuppliedCompany, Description, SuppliedName, Status FROM Case WHERE AccountId = '{pv0}'`
- Retrieve all Opportunities for an Account
  - `SELECT Amount, CloseDate, Name, Type, Probability, StageName FROM Opportunity WHERE AccountId = '{pv0}'`
- Retrieve all Contacts for an Account
  - `SELECT Birthdate, Phone, Email, Name, LastName, Salutation, Title FROM Contact WHERE AccountId = '{pv0}'`

## Mail Merge Queries vs. Quick Merge

These queries are to be used with Conga Mail Merge solutions.

- Example 1:
  - The Master query retrieves the fields selected as the master data set in the Mail Merge process. We are grouping by an account ID and name to give us a unique document per account.
    - Master: `SELECT Id, Name, BillingCity, BillingCountry, BillingState, BillingStreet, BillingPostalCode FROM Account GROUP BY Account.Id, Account.Name`
  - The Detail query gathers child data (Opportunity in this example) that will be related to the master records. The grouping identifies which parent records to relate to the child data.
    - Detail: `SELECT Account.Id, Account.Name, SUM(Amount) FROM Opportunity GROUP BY Account.Id, Account.Name`
- Example 2
  - The Master query retrieves the fields selected as the master data set in the Mail Merge process. This query groups by a contact to give us a unique document per Contact.




- Master: `SELECT Id, Email, Name, HomePhone, MailingCity, MailingCountry, MailingState, MailingStreet, MailingPostalCode FROM Contact GROUP BY BillingContact__c, Account.Name`
- The Detail query gathers child data (Opportunity in this example) that will be related to the master records. The grouping identifies which parent records to relate to the child data.
  - Detail: `SELECT BillingContact__c, SUM(Amount), Account.Name FROM Opportunity GROUP BY __c, Account.Name`

## Conga Query Manager Limits

The following limits apply to Conga Queries built and managed with the Conga Query Manager:

- Conga Composer supports queries that result in a maximum of 25,000 rows and no more than 10 Mb of total data.
- Each Conga Composer solution has a combined row limit of 25,000.
- The resulting dataset from a query that uses a GROUP BY clause may contain a maximum of 200 rows.
- Salesforce doesn't support queries that attempt to group on a formula field.
- Salesforce doesn't support queries of more than 100,000 characters.

 Querying objects with large numbers of records may result in timeouts. Refer to Salesforce documentation about [working with large datasets](#). Due to the complicated nature of SOQL statements, our support staff can only assist with the creation of SOQL statements generated through our tools, such as Conga Query Builder.

## Create A Conga Query With Conga Query Builder

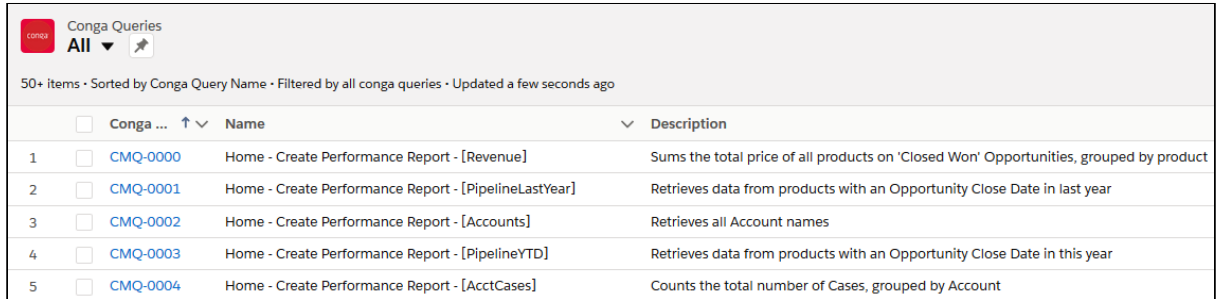
 Due to the almost unlimited customizability of SOQL statements, Technical Support Team will only support SOQL queries generated through Conga Query Builder or written by following one of our support articles. The technical Support Team does not support custom-written queries.

Use Conga Queries and SOQL statements to merge Salesforce data into your documents. Conga Queries are used any time you need to access child object data or need to access

data that is not on your master object. See [About Conga Query Manager](#) for more information.

## To create a Conga Query

1. Navigate to the **Conga Queries** tab.

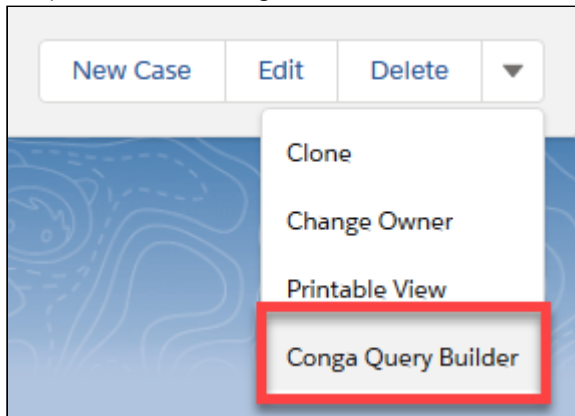


	Conga ...	Name	Description
1	<a href="#">CMQ-0000</a>	Home - Create Performance Report - [Revenue]	Sums the total price of all products on 'Closed Won' Opportunities, grouped by product
2	<a href="#">CMQ-0001</a>	Home - Create Performance Report - [PipelineLastYear]	Retrieves data from products with an Opportunity Close Date in last year
3	<a href="#">CMQ-0002</a>	Home - Create Performance Report - [Accounts]	Retrieves all Account names
4	<a href="#">CMQ-0003</a>	Home - Create Performance Report - [PipelineYTD]	Retrieves data from products with an Opportunity Close Date in this year
5	<a href="#">CMQ-0004</a>	Home - Create Performance Report - [AcctCases]	Counts the total number of Cases, grouped by Account

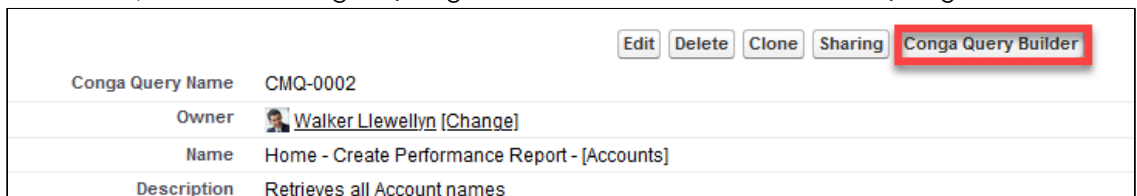
2. Click **New** to create a new Conga Query record.
3. Type a Name and Description.
4. Leave the SOQL Select Statement field blank.

*Info:* If you already have a SOQL query, you can enter it into the SOQL Select Statement field.

5. Click **Save**.
6. Click **Conga Query Builder**.
  - In Lightning, click the Conga Query Builder button. You may have to select the dropdown in the right-hand corner to see the button:

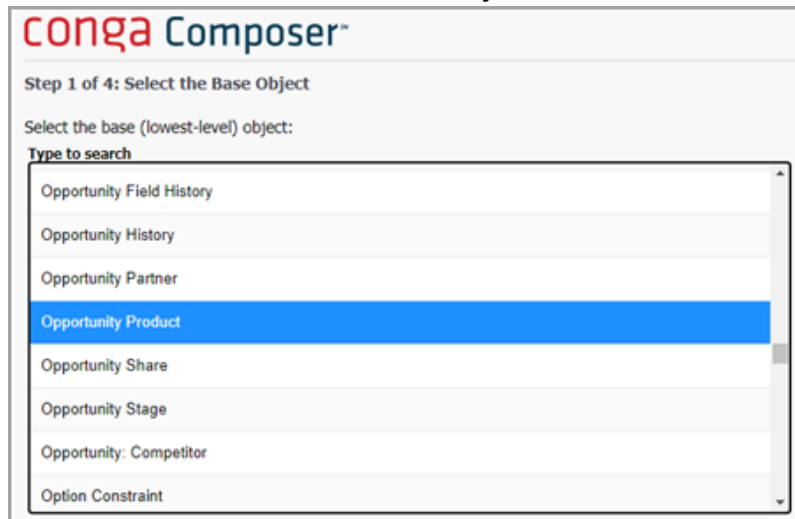


- In Classic, click the Conga Query Builder button to launch the Query Builder.




	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Clone</a> <a href="#">Sharing</a> <a href="#">Conga Query Builder</a>
Conga Query Name	CMQ-0002
Owner	Walker Llewellyn <a href="#">[Change]</a>
Name	Home - Create Performance Report - [Accounts]
Description	Retrieves all Account names

- Click an object in the list to select it as the base object of your query. The object should be the lowest level object in your data structure.  
For example: If you are querying for Opportunity Line Items but your button is on the Account, the lowest level object would be Opportunity Line Items (Opportunity Product). You would select that object on this screen.




- Click **Next**.
- Click on the fields you want to display and click >> to move them to the Selected Fields area. Click the headers to access additional fields.
- Click **Next**.
- Set the **Field** dropdown to the ID of the object where your button resides.
- Set the **Operator** dropdown to equals and leave the **Value** field blank.

 You can specify additional selection criteria if needed.

- (Optional) Choose a sort order and number of rows to display.
- Click **Next**.
- On the resulting page, you see your SOQL statement.

 You can also test your query from this page.

- (Optional) To test your query, enter the Salesforce ID of an appropriate record into the pv0, pv1, pv2, pv3, or pv4 fields and click **Preview**.  
The information is displayed in a pop-up window so make sure you have any pop-up blockers disabled or have set your browser to allow pop-ups from Salesforce.
- When you have completed testing, click **Save**.

 If you manually edit the SOQL query generated by the Query Builder, do not use the Query Builder again for that record because your edits will be overwritten.

## Example

Let's say you want to display a list of related contacts from an Account in your generated document. To create that query you would:

1. Open the **Conga Queries** tab.
2. Click **New**.
3. Fill in Query record information:
  - a. Name: "Account Contacts"
  - b. Description: "Gather contact information for accounts"
4. Leave both the SOQL Select Statement fields and the Key field blank.
5. Click **Save**.
6. Click **Conga Query Builder**.
7. Select **Contact**.
8. Click **Next**.
9. Select the following fields and click >> to move them to the Selected Fields area.
  - Business Phone
  - First Name
  - Last Name
  - Title.
10. Click **Next**.
11. Set the **Field** dropdown to Account ID.
12. Set the **Operator** dropdown to "equals".
13. Leave the **Value** field blank.
14. Click **Next**. The SOQL statement should display: `SELECT Phone, FirstName, LastName, Title FROM Contact WHERE AccountId = '{pv0}'`
15. (Optional) To test the query, enter an Account ID in the pv0 field and click **Preview**. In the pop-up window, you should see up to 20 rows of contacts associated with the specific Account ID.
16. Click **Save**.

## FILE is the related List or in Attachments and you need an IMAGE

When FILE is the related List or there is a file in Attachments and you need to grab an IMAGE. The image must be in the document, not an attachment.

1. Navigate to **Salesforce Setup** → **Salesforce Files** → **Fields** to create a custom field on the Salesforce File Object.

2. Create a new formula text field and name the field. For this example, the field name is File Image URL.
  3. Enter the following URL: `LEFT( $Api.Enterprise_Server_URL_100, FIND( ".com", $Api.Enterprise_Server_URL_100 ) + 4)& "sfc/servlet.shepherd/version/download/" & Id`
  4. Create a Conga Query with this syntax: `SELECT ContentDocument.LatestPublishedVersion.Image_URL_Formula__c FROM ContentDocumentLink WHERE LinkedEntityId = '{pv0}' AND ContentDocument.FileType IN ('PNG','JPG','JPEG')`
- i If you want to filter with the title instead of the file type, then use this field: `ContentDocument.Latestpublishedversion.Title`
5. In the button URL, add the Query: `&QueryID=[FileImages]IDofQuery`
  6. In the template, the following should appear: `{{TableStart:FileImages}} {{IMAGE:LATESTPUBLISHEDVERSION_FILE_IMAGE_URL}} {{TableEnd:FileImages}}`
  7. In Solution Manager, click **Add Query Data**.

## Retrieving a document stored in a Salesforce file

In Composer, you can use a query to retrieve documents and files stored in Salesforce. Very helpful for Files instead of Attachments

Use a SOQL query to retrieve files stored in Salesforce when launching a Composer button. To leverage a SOQL query that retrieves files stored in Salesforce, create a [Conga Query](#) record with the syntax below.

- `SELECT ContentDocument.id  
FROM ContentDocumentLink  
WHERE LinkedEntityId = '{pv0}'`

i To show the most recent File from the related list, add the following to the query:  
`ORDER BY ContentDocument.LastModifiedDate DESC LIMIT 1`

To show File(s) that contain a key phrase, add the following to the query: `AND ContentDocument.Title LIKE '%KeyPhrase%'`

Add the new Conga Query record to a Composer button URL with the `Qvar0ID`, `Qvar0Format`, and `TemplateID` parameters. Reference the sample syntax below.

- `&Qvar0ID=QueryIDfromAbove  
&Qvar0Format=10000  
&TemplateId={qvar0}`

Limited to 20 templates or document files.

## Using SOQL Queries with Conga Composer

In addition to using Salesforce reports to gather information from Salesforce, you can also use Salesforce Object Query Language (SOQL) queries stored in the [Conga Query Manager](#). This custom object stores SOQL statements that will retrieve data related to the Master Object.

See [Create a Conga Query](#) for step by step instructions on creating a query with the Conga Query Builder.


## Using Queries Instead of Reports

SOQL Queries have the following advantages over Salesforce Reports for gathering data:

- Queries are slightly faster and more reliable than Salesforce Reports.
- Queries can access objects that cannot be displayed in Salesforce Reports; for example, Notes and Approval History.
- Queries reduce clutter on the Salesforce Reports tab.
- Queries have tighter security and visibility controls than Salesforce Reports.
- Queries are more reliable when multiple queries (over 10) are included in a Conga Composer solution.
- Using Salesforce reports remains a viable option for basic requirements.
- Queries use the API name of the fields, so can be used in multi-language orgs. Salesforce reports use the field name only.

Here are some important items to note:

- Conga Composer will retrieve a report (or query) that contains a maximum of 2,500 rows and less than 5 MB of total data. Also, each Conga Composer solution has a combined row limit of 25,000.
- Use Queries instead of Salesforce Reports for related lists when using a portal / community.
- Conga Queries are recommended for Salesforce Lightning.

 Due to the complicated nature of SOQL statements, our support staff can only assist with the creation of SOQL statements generated through our tools, such as Conga Query Builder.

# Creating Composer Templates

Information for developing Word, Excel, PowerPoint, HTML Email, and PDF templates for Composer.

Templates determine the final look and feel of your merged documents. Conga Composer works with templates created in the leading Microsoft Office products such as Word, Excel, and PowerPoint. Composer also works with Adobe Portable Document Format (PDF) documents, and you can create templates for HTML email. You can use up to 10 templates in each solution.

Select one of the following topics for more information:

- [Composer Template Basics](#)
- [Word Templates](#)
- [PowerPoint Templates](#)
- [Excel Templates](#)
- [HTML Email Templates](#)
- [PDF Templates](#)

## Composer Template Basics

Select one of the following topics for more information:

- [About Conga Templates](#)
- [About Templates and Conga Applications](#)
- [The Template Builder](#)
- [About Data in the Template Builder](#)
- [Create and Add Templates to Conga Solutions \(Process\)](#)
- [Use a Template from the Template Library](#)
- [Create a Template with Template Builder](#)
- [Add a Template to Conga Template Manager](#)
- [Conga Email Template Manager](#)
- [Pre-selecting a Merge Template for a Conga Composer Solution](#)
- [Conga Template Library for Composer Release 7](#)
- [Conga Template Library for Composer Release 8](#)
- [Merge Fields](#)
- [Use the Template Builder to Construct your Template](#)
- [Enabling Conga Composer Templates Saved as Salesforce Files for Community Users](#)

## About Conga Templates

Conga Templates can be created by using the Template Builder or directly in the source application. You can also take advantage of the [Template Library](#) to select and customize a pre-formatted template. To create a template, you should have the source application installed on your computer.

## The Template Builder

As you use the [Template Builder](#) to identify the merge fields you want to add to a template, the Template Builder can create a new template document in the intended application (Microsoft Word, Excel, or PowerPoint, or Adobe Acrobat Pro).

- ✔ HTML email templates can be created in the Conga Advanced Email Editor in the Conga Email Templates custom object. Often, fields are cut and pasted from Microsoft Word® or another text editor.

## Source Application

You can also drag and drop merge fields directly from the Template Builder, copy and paste fields from the View Data Workbook, or type the merge field names directly into the document. Using the Template Builder or the View Data Workbook reduces the chance of errors in field naming.

When working in the source application, you should have a working knowledge and familiarity with the software; that level of detail is not covered here.

## Template Library

The Conga Template Library contains several templates for common solutions set up using standard Salesforce® objects and fields. While most work as-is, at the least, you will likely want to change the company information and logo. Templates include sample output files so that you can view a merged template, and if the setup is required to use a template, it includes usage notes.

We include two template libraries. Although the libraries are listed for [Composer Release 7](#) and [Release 8](#), if you are using Release 8 or higher, you can also use templates developed for Release 7. The templates for Release 8 use text-based merge fields, which are easier to use but were not supported for Release 7.



## The Conga Template Manager

After you build your document template, you can store it in the Conga Template Manager to make it available for Conga applications.

The Conga Template Manager is a custom object that's included with the Conga Composer application. It's designed to house document templates for use in Conga Composer and Conga Mail Merge processes. The Conga Template Manager is the preferred storage location for Conga templates.

The benefits of using the Conga Template Manager include security, source control, ability to use the Master Field To Set operation on a per-template basis, and reduced need to update IDs in button URLs.

## The Conga Email Template Manager

After you build your email template, you can store it in the Conga Email Template Manager to make it available for Conga applications.

The Conga Template Manager is a custom object, included with the Conga Composer application, and designed to house document templates for use in Conga Composer and Conga Mail Merge processes.

## Template Groups

You can add group templates during run-time. To do so, you add a Template Group name in the Template Group field on the Conga Template Record for each template you want to add. Then use the TemplateGroup parameter to select which group of templates to show to your users in the Composer user interface.

**i** The name you enter in the Template Group field must be identical for each template in the group.

## Merge Template Storage Locations

After designing a template you may store your template in any of these locations:

- The Conga Template Manager (for document templates)
- The Conga Email Template Manager (for email templates)
- The Salesforce template repository (Setup → Communication Templates → Mail Merge Templates)

- Your local hard drive or network drive

✓ Recommended for System Administrators during the building and testing phase. Do not use a colon in document file names.

## About Templates and Conga Applications

A template determines how a merged document will appear and what information it will contain.

Creating a template requires some familiarity with the source software application. To create a working template in Microsoft Word, for example, you should be familiar with how to create documents in Word. The more familiarity and mastery you have in the source software application, the easier you'll find it and perhaps the more you'll be able to do with the styling of the final output.

These topics are intended to help you understand what features are available within the templates and provide instruction on how to accomplish what you want to do.

## What Types of Documents Do You Need

Conga Composer solutions can output to a variety of document types. Most solutions output one type of document, but more advanced solutions may include multiple templates, sometimes using different document types. For example, a quote solution might generate the quote using a Microsoft Word template, use a parameter to save the resulting document as a PDF, attach the PDF to a record in Salesforce®, and then email the PDF using an HTML email template.

You can create templates to output to:

- Microsoft Word
- Microsoft PowerPoint
- Microsoft Excel
- Adobe Portable Document Format (PDF)
- HTML Email

Once you know the types of documents you want your solution to use, you can look at what data you need to have for each template document.

## Data Needed for Your Template Determines the Reports and Queries to Include

The information you want to display on the merged documents must first be identified and gathered. For example, if your final document will be an invoice, you will want data about the account, order number, inventory levels, shipping information, price, and the sales representative. Data contained in the Salesforce Master Object where the button is placed will be automatically available to Conga Composer, but all other data must be gathered by using reports or queries.

## Existing Documents as Sources

Most likely you'll have source documents to work from. These can be electronic or printed documents that you are tasked to reproduce as templates. Those source documents can quickly give you a list of data fields that you'll need to collect.

If you don't have an existing document to identify the fields, you can mock up a template and list the fields, their object, and any special formatting required. (For example, you might need the item price, formatted in the currency specified in the record.)

## Identify Repeating Data and Conditional Text

Your template may use tables of data. In a quote, for instance, you might want to separate items by product family. Conga Composer templates easily handle tables of repeating data, and can show or hide data based on conditions, such as whether a specific table contains data—it can hide tables without data—or whether an order total that exceeds a threshold number receives different payment terms.

The template type you choose may well depend on the repeating data and conditional text features you need. Microsoft Word is the most full-featured application for repeating data and the singular choice for conditional statements; PowerPoint and Excel have limited ability to work with repeating data.

The Template Builder can help you create tables of repeating data, so that adding one is as simple as drag and drop.

## Consider Security

Solutions are run by your end users and so must be accessible to those with the proper security credentials. And not by those who should not have access.

By storing templates in Salesforce, Conga Composer ensures that all authorized users can access the same templates.

If you need to password-protect a merged document, you can use parameters to convert a Word template into a PDF document and specify the password needed to open the document.

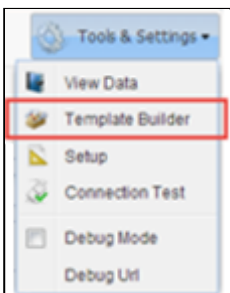
## Internal or External Audience

In addition to who can run the solution, you should also consider who will receive and view the merged documents. Is the audience internal and the information company confidential? If so, you may want to add distribution warnings and copyright information.

## The Template Builder

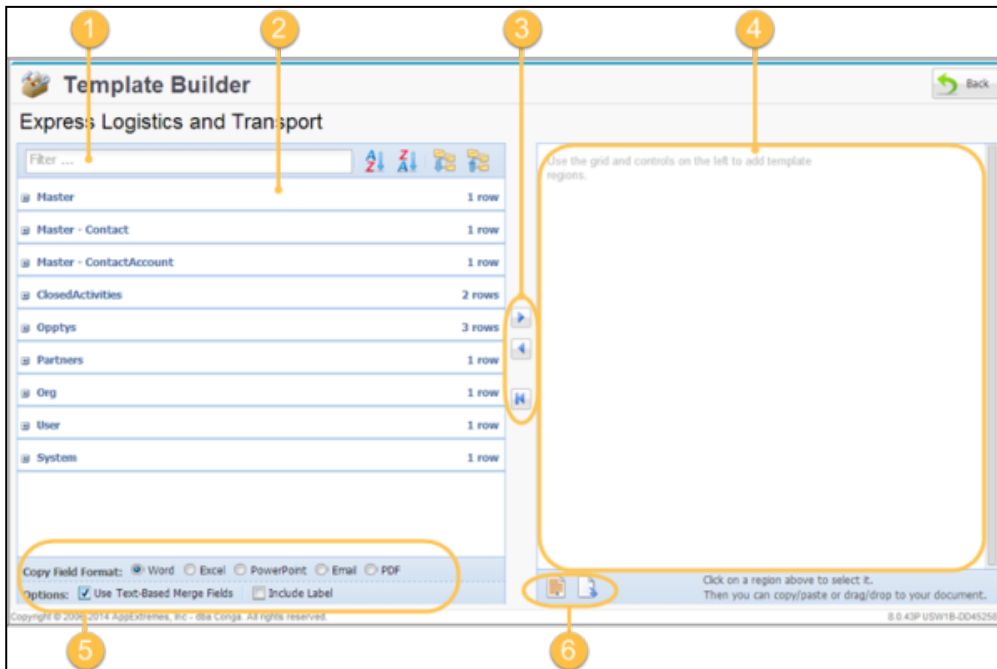
An essential tool in the development of Conga Composer solutions is the Template Builder.

The Template Builder is an essential tool in the development of Conga Composer solutions is the Template Builder. This tool is located in the *Tools & Settings* drop-down menu in the upper-right corner of the Conga Composer page.



The Template Builder exposes the data retrieved by Conga Composer from the Master Object and from any Salesforce Reports or Conga Queries referenced in the Conga Composer URL.

The field names listed in the left pane indicate the field names you must use in your templates. The field names displayed here will be used to create merge fields on your Word, Excel, PowerPoint, Email, or PDF template which will populate the corresponding data from Salesforce.



1	Use the Filter box to quickly locate data fields that match your search terms.
2	Click the plus sign next to a data set to expand it to see and select data fields.
3	Use arrows to move selected data to and from the template work area.
4	Stage data fields and tables of repeating data in the template work area before adding the fields to the template.
5	Choose the type of template you are building, as well as whether to use text-based merge fields and labels as data fields are moved to the template work area.
6	Select fields in the template work area and then use the buttons to copy and paste fields to your template or let Template Builder create and populate a new template. You can also copy and paste or drag and drop fields from the template work area to an open template document.

**i** View Data (available from the Tools & Settings menu on the Conga Composer window) will download the available data fields into an Excel workbook.

## About Data in the Template Builder

Data is displayed in the Template Builder to be accessed in the Conga Composer merge process.

The Master Object, identified with the ID parameter in the solution URL, determines what data is displayed in the Template Builder. Conga Composer retrieves all fields (both standard and custom) from the Master Object and displays them in the Template Builder so you have them available for merging into your template.

Conga Composer can also be run from the home page without a Master Object selected. This allows you to report organization-wide as you are not limited to a Master Object record. For more information, see Conga Global Merge for Solution Manager.

Solution URL with Opportunity as Master Object:

```
https://composer.congamerge.com  
?sessionId={!API.Session_ID}  
&serverUrl={!API.Partner_Server_URL_290}  
&id={!Opportunity.Id}
```

For example, if your solution URL contains `&Id={!Opportunity.Id}`, the Opportunity object is your Master Object, and you will place the Conga Composer button on the Opportunity page layout. Accordingly, the fields presented in the Master dataset on the Template Builder (or in the View Data workbook) will be primarily Opportunity fields.

The User and Org datasets appear in the Template Builder only if Track Activities is enabled for the Master Object. When using certain standard Salesforce objects (Account, Contact, Opportunity, or Quote) as the Master Object, Conga Composer also displays fields from related objects in the Master dataset.

The following fields are available when standard Salesforce objects are selected as the Master Object for Conga Composer:

Conga Composer Master Object (Dataset Name)	Additional Object Data Retrieved (Dataset Name)
Account (Master)	<ul style="list-style-type: none"> <li>• Account Owner–User object (Master)</li> <li>• Primary Contact from Contact Roles (Master - Contact)</li> <li>• Primary Contact’s Account–may differ from the Master Account (Master - ContactAccount)</li> <li>• Primary Contact’s Account Owner–User object (Master - ContactAccount)</li> <li>• Primary Contact Owner–User object (Master - Contact)</li> </ul>
Case (Master)	<ul style="list-style-type: none"> <li>• Account –look-up from Case (Master - Account)</li> <li>• Account Owner (Master - Account)</li> <li>• Case Owner–User object (Master)</li> <li>• Contact–lookup from Case (Master - Contact)</li> <li>• Contact Owner (Master - Contact)</li> </ul>
Contact (Master)	<ul style="list-style-type: none"> <li>• Account–lookup from Contact (Master - Account)</li> <li>• Account Owner–User object (Master - Account)</li> <li>• Contact Owner–User object (Master)</li> </ul>
Opportunity (Master)	<ul style="list-style-type: none"> <li>• Account (Master - Account)</li> <li>• Account Owner–User object (Master - Account)</li> <li>• Opportunity Line Items (OppLineItems)</li> <li>• Opportunity Line Items’ Products (OppLineItems)</li> <li>• Opportunity Owner –User object (Master)</li> <li>• Primary Contact from Contact Roles (Master - Contact)</li> <li>• Primary Contact’s Account (Master - ContactAccount)</li> <li>• Primary Contact’s Account Owner–User object (Master - ContactAccount)</li> <li>• Primary Contact Owner –User object (Master - Contact)</li> </ul>
Quote (Master)	<ul style="list-style-type: none"> <li>• Account–lookup from Quote (Master - Account)</li> <li>• Account Owner–User object (Master - Account)</li> <li>• Contact–lookup from Quote (Master - Contact)</li> <li>• Contact Owner–User object (Master - Contact)</li> <li>• Opportunity –look-up from Quote (Master - Opportunity)</li> <li>• Opportunity Owner–User object (Master - Opportunity)</li> <li>• Quote Line Items (QuoteLineItems)</li> <li>• Quote Line Items’ Products (QuoteLineItems)</li> </ul>

## Create and Add Templates to Conga Solutions (Process)

You can create a Conga Template at any point in the process of designing a solution. The Conga Trigger shown below has been proven to work well and reduce the number of iterations and refinements.

Gather materials & requirements to identify fields and repeating data.

- Gather requirements from stakeholders.
- Use existing documents as source material.
- Identify the objects and data fields you need from Salesforce.
- Identify repeating data, such as line items, and calculations.
- Mockup a template with static text and placeholders for merge fields.

This step sets the stage for a successful template. You have a list of all the fields you need and the objects where they are stored.

Create a report or SOQL query to retrieve only the information you've identified as essential to the template. Targeted reports or queries are faster and more efficient than broader ones. At the end of this stage, you'll have a good idea of how you want to format and arrange the information on the template document. You'll also know what software you'll use to create the template. If the solution requires more than one template, you will know that, too.

Create or identify existing reports and queries, if needed.

- Create or select one or more existing reports or queries to gather the information for your solution template.
- Optimized reports and queries will consume less resources and load more quickly.
- You can create reports and queries manually, or by using Solution Manager.
- For solutions that require data contained in the Master Object only, no report or query may be needed.

After you've identified or created your reports and queries, build your solution and include the reports or queries in the button URL. You may find it more helpful to build your solution to view the field data that is already available through the Master Object. Then, circle back to create a report or query to gather the data you need.

Build your solution, either manually or by using Solution Manager.

- Build your solution using Solution Manager, a Solution Pack, or manually.
- Include the appropriate reports or queries.



- If using Solution Manager, select a sample record to load data into the Template Builder.

Use Template Builder to create templates.

- The Template Builder makes it easier to add merge fields to a template document.
- You can add lists or tables of repeating data.
- Add appearance parameters (picture switches) to format dates, numbers, and currency field data.
- Use the document software to style and format the template document.
- More information: [The Template Builder](#). [Display data in Template Builder](#).

You can also use a template from one of the Template Libraries. Use the Conga Template Manager to store templates.

- The Conga Template Manager is a custom object that makes storing templates easy.
- Store document templates in the Conga Template Manager.
- Store HTML email templates in the Conga Email Template Manager.
- (Optional) Add the templates to your solution or let users select templates at runtime.
- You can add template IDs to a button URL using Solution Manager or manually.
- Parameters let you select one or more templates automatically whenever the solution runs.
- You can specify a group of templates from which users can select during the merge process.
- You can also let users select a template during the merge process.
- For advanced solutions, you can add logic to automatically select templates based on conditions.
- More information: [Streamlining Template Selection](#).

## Use a Template from the Template Library

The Conga template libraries offer pre-built Composer templates you can use in your solutions. View available templates using the links below.

- [Template Library \(Release 7\)](#)
- [Template Library \(Release 8\)](#)

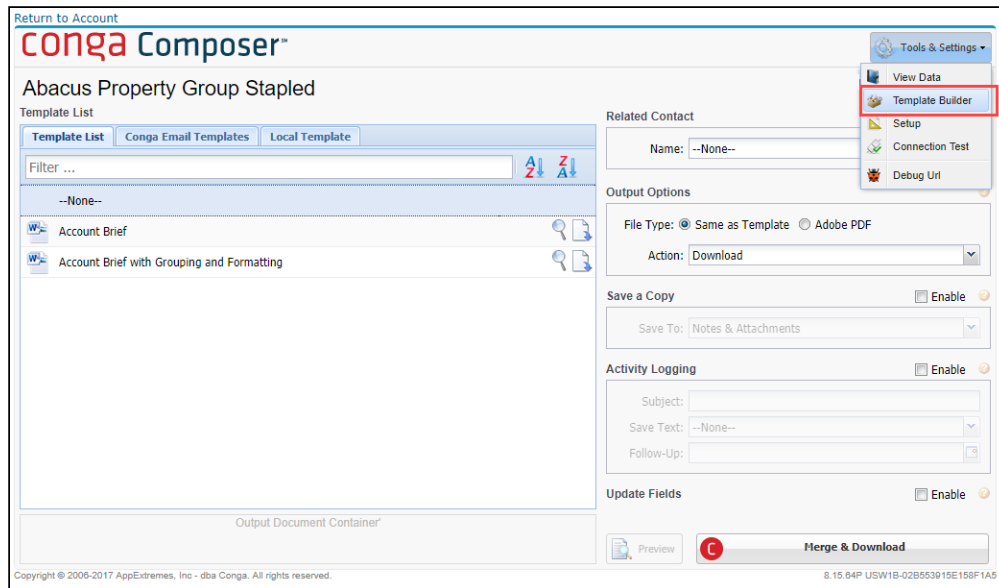
## Create a Template with Template Builder

You can use the Conga Template Builder to quickly build a template for a solution. For more information, see [The Template Builder](#).

✓ You may want to start with a template from our [Template Library](#).

## To access the Template Builder

1. Launch Conga Composer from a Composer button (or from a Solution Record).
2. In the Conga Composer window that opens, click **Tools & Settings** → **Template Builder**.



Conga Composer loads the datasets from the master object, including any data you created using reports and queries.

## Template Builder

The left side of the Template Builder lists mail merge field names from Salesforce as well as sample data. To use these fields to create a template:

1. In the Copy Field Format area, select the document type for the template (i.e. Word, PPTX, Excel, etc.).
2. In the Options area, select or clear the Include Label box to include or ignore the field label as merge fields are copied into the template region. For example, with Include Label selected, you will get: `Account Name: {{ACCOUNT_NAME}}`. Without Include Label selected, you will just get the merge field: `{{ACCOUNT_NAME}}`.
3. In the left pane of the Template Builder, locate and double-click a field to add it to the right pane. Optionally, you can click the field name and then click the right arrow to add it.
4. Repeat the previous step to add additional fields.

5. When all the merge fields you want to add to your template are in the right pane of the Template Builder, you can either copy and paste or drag and drop those merge fields into your template.
6. Save the template locally and perform a test merge with it.

✔ You can also select a field, copy it, and paste it directly into your document template rather than adding merge fields to the right pane of the Template Builder. With this method, you can only add merge fields one at a time.

After a test merge, you will likely need to format date and number fields. See [Formatting merge fields with picture switches](#) for more information.

## Add a Template to Conga Template Manager

Conga Templates, also known as Conga Template Manager, are records that are created from the Conga Template custom object. Conga Template Manager stores document templates for Conga Composer.

### To add a template

1. Navigate to the **Conga Templates** Salesforce tab.
2. Click **New**.
3. Give your template record a descriptive name. This is the display name that will show in the Composer UI.
4. (Optional) Add a description to your template as necessary.
5. (Optional) In the Template Group field, type a template group name (if using the [TemplateGroup](#) parameter in your Composer solution).
6. (Optional) In the Master Field to Set fields, enter up to three master object fields to update whenever the template is merged. Follow the syntax as specified in the Help Text on the Salesforce field.
7. Click **Save**.
8. On the Conga Template page, find the *Files* related list and follow the steps to upload your template file to the record.

⚠ It is possible to use the Notes & Attachments related list, but it is recommended to update the page layout to use Files. Salesforce is [deprecating Notes & Attachments](#) and it should no longer be used.

✔ Do not use a colon in document file names.

## Conga Email Template Manager

The Conga Email Template Manager custom object enables you to create Conga Email Templates that work a lot like Conga Composer templates built in Microsoft Word.

The Conga Email Template Manager custom object enables you to create and store Conga Email Templates (CET) in your Salesforce instance. These templates can be used in addition to - or in place of - a file template (Microsoft Word, PowerPoint, Excel, or a PDF) in your Composer solution.

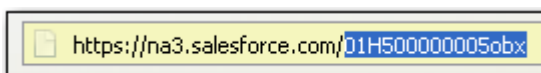
## Pre-selecting a Merge Template for a Conga Composer Solution

For most Conga Composer solutions, it's often desirable to pre-select the template that you've intended to use with this particular merge operation. You may do so by including the `TemplateId` parameter in the URL that calls Conga Composer.

```
https://composer.congamerge.com/?SessionID={!API.Session_ID}
&ServerURL={!API.Partner_Server_URL_290}
&Id={!Contact.Id}
&TemplateId=01H500000005tMS
```

The ID of the desired template is found by navigating to the template, stored in either the Salesforce template repository or the Conga Template Manager.

Click the name of the template and select the ID from the template's URL.



The ID required is from the Template Manager record, not the attachment stored with the record.

## Conga Template Library for Composer Release 7

This template library is a collection of sample templates that offers examples of the types of documents you can create with Conga Composer release 7. You can use these sample templates as-is or modify them to meet your specific needs. Most of these templates will work "out of the box" with a standard Salesforce instance. However, some templates

require a special setup before they will work. Templates requiring setup have an associated Usage Notes PDF file.


For a collection of templates that use Microsoft Word merge fields intended for use with Composer release 8, see [Conga Template Library for Composer Release 8](#).


### Contact Templates

Description	Template
Simple Thank You Letter (Business)	<a href="#">Simple Thank You Letter (Business) Template</a>
Simple Thank You Letter with watermark (Business)	<a href="#">Simple Thank You Letter with watermark (Business) Template</a>
Simple Thank You letter with logo (Donation)	<a href="#">Simple Thank You letter with logo (Donation) Template</a>
Simple Thank You letter (Donation)	<a href="#">Simple Thank You letter (Donation) Template</a>
Gift Certificate	<a href="#">Gift Certificate Template</a>
Labels for Contact Object	<a href="#">Labels for Contact Object Template</a>
Payment Verification Request Template - DocuSign Writeback Anchor Tags Note: You must update the tags.	<a href="#">Payment Verification Request Template (with DocuSign Writeback Anchor Tags) Template</a>
W9 - DocuSign Standard Anchor Tags	<a href="#">W9 with DocuSign Standard Anchor Tags Template</a>

## Conga Template Library for Composer Release 8

The following sample templates show the types of documents you can create with Conga Composer release 8. You can use them as-is or modify them to meet your specific needs.

 Because the queries are specific to demo.org for your reference, not all templates will work "out of the box" for you. To pull the required data from your org, [create queries](#) based on your business use case.

 The Word templates use text-based merge fields and are compatible only with Composer release 8. For a collection of templates that use Microsoft Word merge

fields intended for use with Composer release 7, see [Conga Template Library for Composer Release 7](#).

Description	Template
Account Summary	<a href="#">Account Summary Template</a>
Case Work Order	<a href="#">Case Work Order Template</a>
Contract Agreement	<a href="#">Contract Agreement NDA Template</a>
Donation Letter	<a href="#">Donation Letter Template</a>
Donation Welcome Letter	<a href="#">Donation Letter for multiple contacts Template</a>
EOY Tax Receipt	<a href="#">EOY Tax Receipt Template</a>
Order	<a href="#">Order Template</a>
Proposal (Email)	<a href="#">Proposal Email Template</a>
Proposal (Word)	<a href="#">Proposal Presentation Template</a>
Quarterly Business Review	<a href="#">QBR Template</a>
Quote	<a href="#">Quote Template</a>
Sales Activity Report	<a href="#">Sales Activity Report Template</a>
Statement of Work (SOW)	<a href="#">SOW Template</a>
Standard Service Agreement with Watermark	<a href="#">Standard Service Agreement Template</a>
Simple Thank You letter with logo (Donation)	<a href="#">Thank You Letter Template</a>
Labels for Global Merge	<a href="#">TextBased Labels for Global Merge Template</a>
Envelope for Contact Object	<a href="#">Envelope for Contact Object Template</a>

## Merge Fields

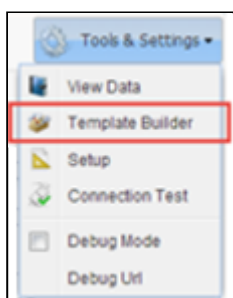
Conga Composer lets you customize and format template documents with special merge fields.

**What is a merge field?** You can insert merge fields and modify text by selecting a word or text area on your document. The syntax for what we call a text-based merge field is as follows: `{{ACCOUNT_NAME}}`.

In Word document templates, you can use special merge fields. To view a list, see [Special Merge Fields in Word Templates](#).

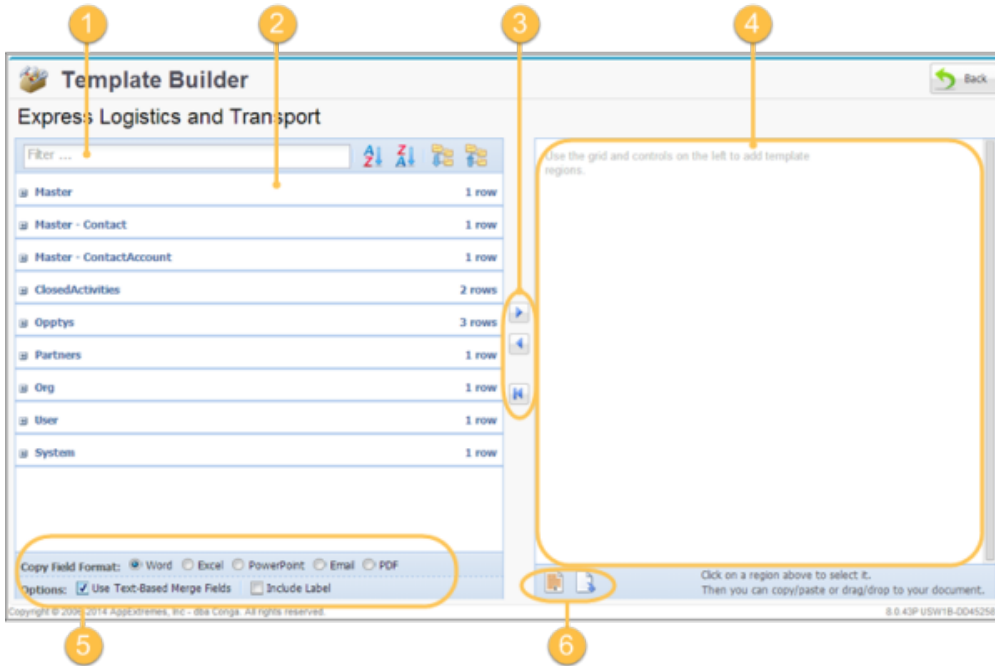
## Use the Template Builder to Construct your Template

An essential tool in the development of Conga Composer solutions is the Template Builder.



The Template Builder is located in the Tools & Settings drop-down menu in the upper-right corner of the Conga Composer page. The Template Builder exposes the data retrieved by Conga Composer from the Master Object and from any Salesforce Reports or Conga Queries referenced in the Conga Composer URL. The field names listed in the left pane indicate the field names you must use in your templates. The field names displayed here will be used to create merge fields on your Word, Excel, PowerPoint, Email, or PDF template

which will populate the corresponding data from Salesforce.



1	Use the Filter box to quickly locate data fields that match your search terms.
2	Click the plus sign next to a data set to expand it to see and select data fields.
3	Use arrows to move selected data to and from the template work area.
4	Stage data fields and tables of repeating data in the template work area before adding the fields to the template.
5	Choose the type of template you are building, as well as whether to use text-based merge fields and labels as data fields are moved to the template work area.
6	Select fields in the template work area and then use the buttons to copy and paste fields to your template or let Template Builder create and populate a new template. You can also copy and paste or drag and drop fields from the template work area to an open template document.

View Data (available from the Tools & Settings menu on the Conga Composer window) will download the available data fields into an Excel workbook. The Template Builder supports



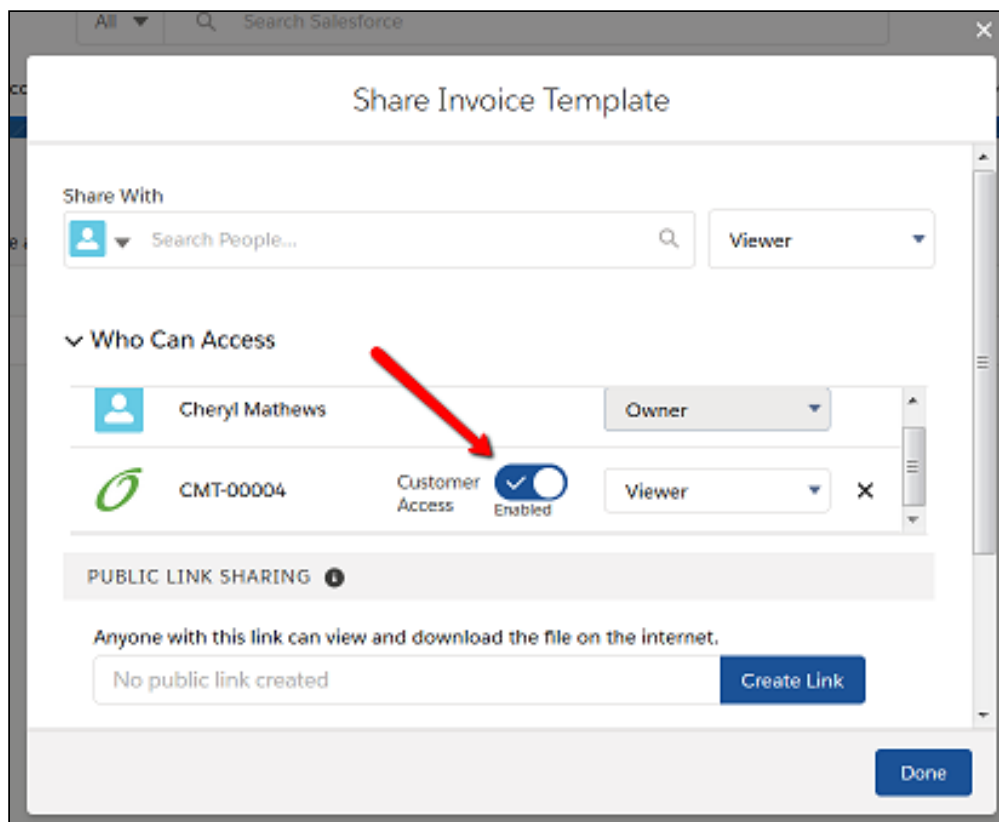
multiple template types. By default, it is set to create text-based merge fields for Microsoft Word.

## Enabling Conga Composer Templates Saved as Salesforce Files for Community Users

If your Community users cannot access Conga Templates saved as Salesforce Files, please follow these instructions:

**⚠ You must be logged into Salesforce Lightning Experience to follow these instructions**

1. Open your Conga Template record.
2. Click **Related** to view the Notes & Attachments related list where your template file is attached.
3. Click **View All**.
4. Click the Dropdown Arrow to the right of your template file and select **Share**.
5. In the Share window expand the Who Can Access section.
6. Make sure the Customer Access Enabled switch is switched on:



# Word Templates

You can merge data from Salesforce into formatted Word documents that can contain images, text, and tables.

- [Word Template Basics](#)
- [Detail Data in Word](#)
- [Special Merge Fields in Word](#)
- [Advanced Word Templates](#)

## Word Template Basics

Get started with Word templates for Composer.

## Word Template Guide

Overview of Composer Word templates.

## Learn Templates and Conga Applications

- [About Templates and Conga Applications](#)
- [Create and Add Templates to Conga Solutions \(Process\)](#)
- [Conga Template Library for Composer Release 8](#)
- [About Conga Templates](#)
- [About Data in the Template Builder](#)

## Data Available for a Solution

- [The Template Builder](#)
- [Create a Template with Template Builder](#)

## Merge Template Storage Locations

- [Add a Template to Conga Template Manager](#)
- [Create a Conga Email Template](#)

## Streamlining Template Selection in Conga Composer

- [Pre-selecting a Merge Template for a Conga Composer Solution](#)

## Create Word Templates

- [Template Example: Microsoft Word](#)
- [About Formatting Word Merge Fields](#)
- [Format Date Values in Word](#)
- [Dynamic Date Formats in Word](#)
- [Format Number Values in Word](#)
- [Format Numbers as Text in Word](#)
- [Dynamic Currency Formats in Word](#)

## Use Detail Data in Word Templates

- [Using Detail Data in Word](#)
- [Repeating Data in Word](#)
- [Tables of Repeating Data in Word](#)
- [Repeating Data within a Paragraph in Word](#)
- [Organize Repeating Data Tables into Groups in Word](#)
- [Grouping Repeating Data with Additional Information in Word](#)
- [Add Page Breaks to Repeating Data Tables in Word](#)
- [Add Footers to Repeating Data Tables in Word](#)
- [TableHide in Word](#)
- [Master Data Conditional TableHide in Word](#)
- [Conditional TableHide for Detail Data in Word](#)
- [Conditional TableHide using an Operator in Word](#)
- [Conditional TableHide Comparing Two Field Values in Word](#)

## Use Conditional Text in Word Templates

- [IF Fields: Comparison Statements in Word](#)

## Use Special Merge Fields in Word Templates

- [Special Merge Fields in Word Templates](#)
- [CHECKBOX in Word](#)
- [Checkboxes for Multi-Select Picklist Fields in Word](#)
- [HTML in Word](#)
- [HYPERLINK in Word](#)
- [IMAGE in Word](#)
- [IncludePicture in Word](#)
- [REPLACE in Word](#)
- [RowCount in Word](#)

- [RTL in Word](#)
- [SUM\(ABOVE\) in Word](#)
- [TableList in Word](#)
- [TEXTBOX in Word](#)
- [WATERMARK in Word](#)

## About Microsoft Word Merge Templates

### Overview

Microsoft Word templates are the most dynamic template type available for Conga Composer. With Word templates, you can merge master Salesforce data and repeating rows of detail or related data, format dates, numbers, and currencies. You can apply Word formatting and styles and use special merge fields for additional functionality like logic, formulas, and hyperlinks.

- ✓ We recommend starting with Word templates to see if they can meet your needs before moving on to another template type like Excel, PowerPoint, HTML Email, and PDF.

### Merge Fields

Microsoft Word templates use merge fields to map data from Salesforce into your Word document. Composer Release 8 supports two types of merge fields:

- Traditional Word merge fields are inserted into the Word document using Word merge field functionality.
- Text-based merge fields are inserted into the Word document by typing the merge field name surrounded by two curly brackets.

Traditional merge field: <<ACCOUNT\_NAME>>

Text-based merge field: {{ACCOUNT\_NAME}}

Merge fields are added to Word templates using the [Conga Composer Template Builder](#).

### Formatting Merge Fields

Merge fields containing dates, numbers, or currencies can be formatted using picture switches. Picture switches are added after the field name using \@ for dates and \# for numbers and currencies.

- Example: {{TODAY \@ "MMMM dd, yyyy"}}

- Example: `{{ANNUAL_REVENUE \# Currency}}`

For more information on formatting merge fields, see: [Format Date Values](#) and [Format Number Values](#)

## Detail Data

You can merge detailed data into a Word template. For example, for a Composer solution running on an Opportunity, you can merge Opportunity Product data into the template. The detail region of your template grows dynamically with rows of repeating data. For example:

Detail Region:

<code>{{TableStart:OppLineItems}}</code> <code>{{PRODUCT_NAME}}</code>	<code>{{OPPORTUNITY_LINEITEM_SALESPRICE \# CURRENCY}}</code>	<code>{{OPPORTUNITY_LINEITEM_QUANTITY}}</code>	<code>{{OPPORTUNITY_LINEITEM_TOTALPRICE}}</code> <code>{{TableStart:OppLineItems}}</code>
---	--	--	--

The output, if there were three products on the Opportunity record, would look like this;

Basic Widget	\$50	35	\$1,750
Standard Widget	\$100	15	\$1,500
Advanced Widget	\$250	10	\$2,500

For more information about detailed regions, see: [Using Detail Data in Word](#) and [Repeating Data in Word](#)

## Special Merge Fields

Special merge fields allow you to use advanced functionality in Word templates. Below is a list of some of the common special merge fields:

- [Adjust](#) - Adjust a date into the future or past (useful in automatically creating due dates 30 days into the future).
- [Checkbox](#) - Create a check box that is either selected or deselected based on the value of the merge field.
- [HTML](#) - Retain HTML rich text formatting in the output document.
- [Hyperlink](#) - Render a clickable hyperlink and format the text of the hyperlink.
- [Image](#) - Merge images into Word templates.


- [Replace](#) - Replace a character with a different character (replace a ; with a bullet).
- [TableHide](#) - Hide a table if it does not contain data.
- [TableGroup](#) - Group data in a table based on a value in the table (useful for grouping products by Product Family).
- [SUM\(Above\)](#) - Calculate values in a table and merge the result as static text.
- [IF Statements](#) - Conditionally show or hide data based on the result of a comparison.
- [Watermark](#) - Merge a watermark into your Word document.


## Template Library


Visit the [Composer Release 8 Template Library](#) to see examples of Word templates.

## Format Date Values in Word

The date formatting picture switch begins with `\@`, followed by any combination of the following parameters.

Parameter	Description		
Month (M)	Displays the month.		
	<table border="1"> <tr> <td>M</td> <td>Displays the month as a number without a leading zero for single-digit months. February is shown as 2.</td> </tr> </table>	M	Displays the month as a number without a leading zero for single-digit months. February is shown as 2.
	M	Displays the month as a number without a leading zero for single-digit months. February is shown as 2.	
	<table border="1"> <tr> <td>M M</td> <td>Displays the month as a number with a leading zero for single-digit months. February is shown as 02.</td> </tr> </table>	M M	Displays the month as a number with a leading zero for single-digit months. February is shown as 02.
	M M	Displays the month as a number with a leading zero for single-digit months. February is shown as 02.	
<table border="1"> <tr> <td>M M M</td> <td>Displays the month as a three-letter abbreviation. February is shown as Feb.</td> </tr> </table>	M M M	Displays the month as a three-letter abbreviation. February is shown as Feb.	
M M M	Displays the month as a three-letter abbreviation. February is shown as Feb.		
<table border="1"> <tr> <td>M M M M</td> <td>Displays the full name of the month. February.</td> </tr> </table>	M M M M	Displays the full name of the month. February.	
M M M M	Displays the full name of the month. February.		
<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The letter M must be uppercase to distinguish months from minutes.                 </div>			

Parameter	Description								
Day (d)	<p data-bbox="437 331 1126 367">Displays the day of the month or the day of the week.</p> <table border="1" data-bbox="437 385 1426 1146"> <tr> <td data-bbox="437 385 491 584">d</td> <td data-bbox="491 385 1426 584"> <p data-bbox="504 405 1407 479">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="504 501 1034 533">The sixth day of the month is shown as 6.</p> </td> </tr> <tr> <td data-bbox="437 584 491 784">d d</td> <td data-bbox="491 584 1426 784"> <p data-bbox="504 604 1366 678">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="504 701 1050 732">The sixth day of the month is shown as 06.</p> </td> </tr> <tr> <td data-bbox="437 784 491 943">d d d</td> <td data-bbox="491 784 1426 943"> <p data-bbox="504 797 1401 833">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="504 855 833 887">Tuesday is shown as Tue.</p> </td> </tr> <tr> <td data-bbox="437 943 491 1146">d d d d</td> <td data-bbox="491 943 1426 1146"> <p data-bbox="504 960 1098 996">Displays the full name of the day of the week.</p> <p data-bbox="504 1019 625 1050">Tuesday.</p> </td> </tr> </table> <div data-bbox="437 1160 1426 1245" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p data-bbox="459 1182 1155 1218"> The letter d can be either uppercase or lowercase.</p> </div>	d	<p data-bbox="504 405 1407 479">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="504 501 1034 533">The sixth day of the month is shown as 6.</p>	d d	<p data-bbox="504 604 1366 678">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="504 701 1050 732">The sixth day of the month is shown as 06.</p>	d d d	<p data-bbox="504 797 1401 833">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="504 855 833 887">Tuesday is shown as Tue.</p>	d d d d	<p data-bbox="504 960 1098 996">Displays the full name of the day of the week.</p> <p data-bbox="504 1019 625 1050">Tuesday.</p>
d	<p data-bbox="504 405 1407 479">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="504 501 1034 533">The sixth day of the month is shown as 6.</p>								
d d	<p data-bbox="504 604 1366 678">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="504 701 1050 732">The sixth day of the month is shown as 06.</p>								
d d d	<p data-bbox="504 797 1401 833">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="504 855 833 887">Tuesday is shown as Tue.</p>								
d d d d	<p data-bbox="504 960 1098 996">Displays the full name of the day of the week.</p> <p data-bbox="504 1019 625 1050">Tuesday.</p>								
Year (y)	<p data-bbox="437 1283 932 1319">Displays the year as two or four digits.</p> <table border="1" data-bbox="437 1337 1426 1736"> <tr> <td data-bbox="437 1337 491 1536">y y</td> <td data-bbox="491 1337 1426 1536"> <p data-bbox="504 1357 1311 1431">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="504 1453 1082 1485">1999 is shown as 99; 2006 is displayed as 06.</p> </td> </tr> <tr> <td data-bbox="437 1536 491 1736">y y y y</td> <td data-bbox="491 1536 1426 1736"> <p data-bbox="504 1556 912 1592">Displays the year as four digits.</p> <p data-bbox="504 1615 673 1646">1999 or 2006</p> </td> </tr> </table>	y y	<p data-bbox="504 1357 1311 1431">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="504 1453 1082 1485">1999 is shown as 99; 2006 is displayed as 06.</p>	y y y y	<p data-bbox="504 1556 912 1592">Displays the year as four digits.</p> <p data-bbox="504 1615 673 1646">1999 or 2006</p>				
y y	<p data-bbox="504 1357 1311 1431">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="504 1453 1082 1485">1999 is shown as 99; 2006 is displayed as 06.</p>								
y y y y	<p data-bbox="504 1556 912 1592">Displays the year as four digits.</p> <p data-bbox="504 1615 673 1646">1999 or 2006</p>								

Parameter	Description								
Hours (h)	<p>A lowercase h bases time on the 12-hour clock.</p> <p>An uppercase H bases time on the 24-hour clock.</p> <table border="1" data-bbox="437 443 1426 1146"> <tbody> <tr> <td data-bbox="437 443 488 600">h</td> <td data-bbox="488 443 1426 600">           Displays the hour without a leading zero for single-digit hours.            9 A.M. is shown as 9.         </td> </tr> <tr> <td data-bbox="437 600 488 792">H</td> <td data-bbox="488 600 1426 792">           Displays hours without a leading zero for single-digit hours on the 24-hour clock.            9AM is shown as 9. 9PM is shown as 21.         </td> </tr> <tr> <td data-bbox="437 792 488 949">h h</td> <td data-bbox="488 792 1426 949">           Displays the hour with a leading zero for single-digit hours.            9 PM. is shown as 09.         </td> </tr> <tr> <td data-bbox="437 949 488 1146">H H</td> <td data-bbox="488 949 1426 1146">           Displays the hour with a leading zero for single-digit hours on the 24-hour clock.            9 AM. is shown as 09. 9 PM is shown as 21.         </td> </tr> </tbody> </table>	h	Displays the hour without a leading zero for single-digit hours. 9 A.M. is shown as 9.	H	Displays hours without a leading zero for single-digit hours on the 24-hour clock. 9AM is shown as 9. 9PM is shown as 21.	h h	Displays the hour with a leading zero for single-digit hours. 9 PM. is shown as 09.	H H	Displays the hour with a leading zero for single-digit hours on the 24-hour clock. 9 AM. is shown as 09. 9 PM is shown as 21.
h	Displays the hour without a leading zero for single-digit hours. 9 A.M. is shown as 9.								
H	Displays hours without a leading zero for single-digit hours on the 24-hour clock. 9AM is shown as 9. 9PM is shown as 21.								
h h	Displays the hour with a leading zero for single-digit hours. 9 PM. is shown as 09.								
H H	Displays the hour with a leading zero for single-digit hours on the 24-hour clock. 9 AM. is shown as 09. 9 PM is shown as 21.								
Minutes (m)	<p>Displays minutes.</p> <table border="1" data-bbox="437 1249 1362 1572"> <tbody> <tr> <td data-bbox="437 1249 517 1411">m</td> <td data-bbox="517 1249 1362 1411">           Displays minutes without a leading zero for single-digit minutes.  <code>{{TIME \@ "m"}}</code> is shown as 2.         </td> </tr> <tr> <td data-bbox="437 1411 517 1572">mm</td> <td data-bbox="517 1411 1362 1572">           Displays minutes with a leading zero for single-digit minutes.  <code>{{TIME \@ "mm"}}</code> is shown as 02.         </td> </tr> </tbody> </table> <div data-bbox="437 1585 1426 1666" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> The letter m must be lowercase to distinguish minutes from months.</p> </div>	m	Displays minutes without a leading zero for single-digit minutes. <code>{{TIME \@ "m"}}</code> is shown as 2.	mm	Displays minutes with a leading zero for single-digit minutes. <code>{{TIME \@ "mm"}}</code> is shown as 02.				
m	Displays minutes without a leading zero for single-digit minutes. <code>{{TIME \@ "m"}}</code> is shown as 2.								
mm	Displays minutes with a leading zero for single-digit minutes. <code>{{TIME \@ "mm"}}</code> is shown as 02.								



Parameter	Description
A.M. and P.M. (AM/PM)	am/pm Displays A.M. and P.M.
	AM/PM Displays A.M. and P.M. as uppercase. <pre> {{NOW \@ "h AM/PM"}} </pre> and <pre> {{NOW \@ "h am/pm"}} </pre> are both shown as 9 AM.

## To format long text (date) in another language

1. Ensure language is enabled in Word.
2. Click **Insert Tab**.
3. Select **Date/Time**.
4. Select **language**.
5. Select **date/time format**.
6. Select **update automatically**.

## Date Format Examples

Picture Switch Syntax	Result
<pre> {{OPPORTUNITY_CLOSEDATE \@ MM-dd-yyyy}} </pre>	03-01-2014
<pre> {{OPPORTUNITY_CLOSEDATE \@ dd/MM/yyyy}} </pre>	01/03/2014
<pre> {{OPPORTUNITY_CLOSEDATE \@ "MMM dd, yyyy"}} </pre>	Mar 01, 2014
<pre> {{OPPORTUNITY_CLOSEDATE \@ "MMMM d, yyyy"}} </pre>	March 1, 2014

 If the result contains spaces, enclose the formatting code in straight quotes.

## Date and Time Format Examples

Picture Switch Syntax	Result
<pre> {{OPPORTUNITY_DATE_AND_TIME \@ "MM-dd-yyyy, at h:mm"}} </pre>	08-22-2016, at 12:03

Picture Switch Syntax	Result
<code>{{OPPORTUNITY_DATE_AND_TIME \@ "MMM dd, yyyy, h:mm"}}</code>	Aug 22, 2016, 12:03

## What Is the Difference Between a Field Label and a Field Name (Word)?

Explains field label vs field name in Word® templates.

Traditional merge fields (ex: «ACCOUNT\_NAME») in Word templates are comprised of three components:

- Field label
- Field name
- Appearance parameters (picture switch)

You may find that some field names in the Template Builder are quite long (ex: PROPERTY\_ALTERNATE\_TENANT\_FIRST\_NAME), which can make formatting your Word template a bit tricky, especially when working in tables. Editing the field label while keeping the field name in tact is a handy technique that offers a bit more flexibility for those lengthy field names.

The field label is what you see when viewing the template. It is the text portion of the field that appears between the double chevrons (« »). In the example below, the field label highlighted is TableStart:Products.

Product	Quantity	Price
«TableStart:Products» »«Product_Name»	«Quantity»	«Total_Price» »«TableEnd:Products»

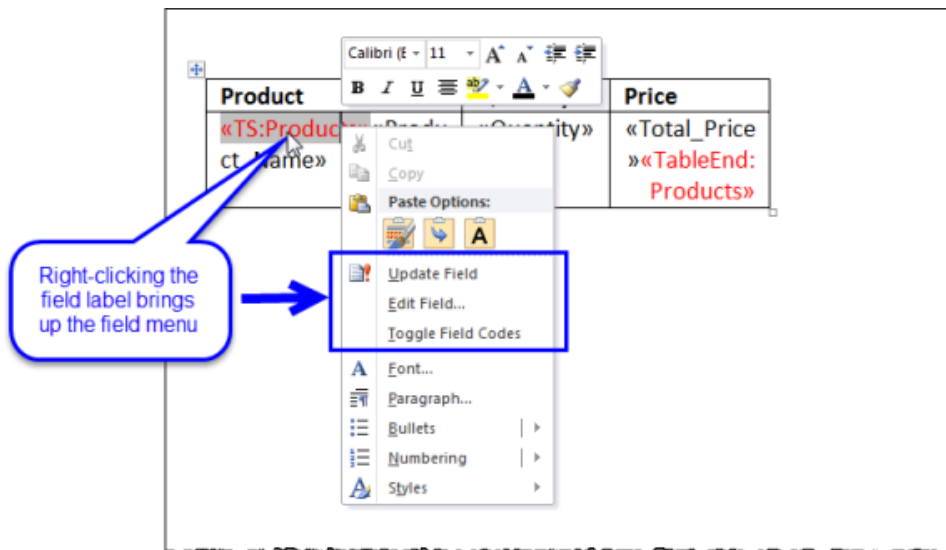
The field label can be changed if needed to better suit the template. To edit a field label, simply place your cursor within the field and alter the field label as desired. DO NOT delete the double chevrons on either side of the field. For example, below TableStart:Products has been shorted to TS:Products to better fit within the table cell.

If using TableGroup or TableGroupStart, you cannot shorten or alter the field label.

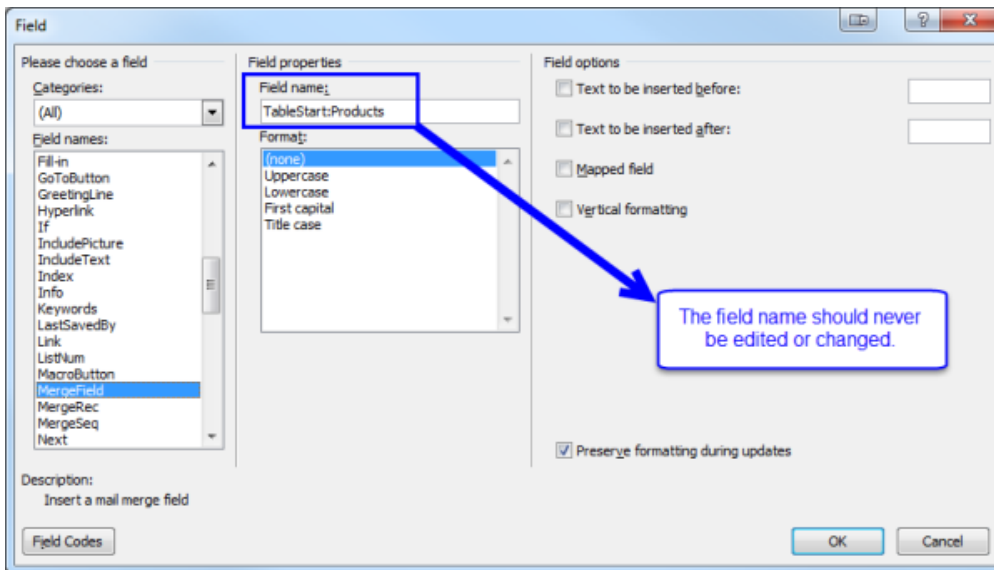
Product	Quantity	Price
«TS:Products»«Product_Name»	«Quantity»	«Total_Price»«TableEnd:Products»

Click within the field and delete or type text as desired to alter the field label.

The field name is the underlying field name, only visible when Edit Field or Toggle Field Codes is selected from the field menu.



Select **Edit Field** from the field menu to bring up the Field dialog box. The field name cannot be changed, shortened, or altered under any circumstances. It must match exactly to the field name listed in the Template Builder.



If you click **OK** from the Field dialog box, Word will revert your field to the full field label; instead choose **Cancel** to maintain the changes you previously made to your field label. Your field label will now remain as the altered version while the underlying field name remains the full field name as referenced in the Template Builder.

## Template Example: Microsoft Word

This invoice template is an example taken from the Template Library.


The invoice template below is an example taken from the Template Library. It illustrates how a template document can use several elements, such as tables (with and without borders and coloring), groups of repeating data, calculations, conditional statements, and other field formatting to produce a seamless and customized invoice at merge time.

The screenshot shows an invoice template for 'WORLDS AWAY INCORPORATED'. It includes a header with the company logo, a table for account and payment information, a table for line items, a subtotal row, an invoice total due box, a conditional message, and payment details. Callouts 1-8 highlight these elements.

1	A page header. Each page can have its own header or you can repeat headers across pages.
2	<p>A borderless table containing the account information. You can apply text formatting, such as colors, font sizes, and so on, to any template text.</p> <p>The Payment Due Date has a date field adjustment of 30 days from today [details], and is formatted to show the full month name, the day of the month, and the four digit year. [details]</p> <p>The Total Amount Due also has a Word field formatting picture switch to display the amount as currency. [details]</p>

3	The TableGroup field will group like data. In this case, a table of repeating data will appear for each product family listed in the invoice. <a href="#">[details]</a>
4	<p>This table row will repeat to display each returned row of data that the solution report or query gathered. Repeating data sections begin with a TableStart field, and end with a TableEnd field. <a href="#">[details]</a></p> <p>Repeating information is not confined to a table. You can place detail items directly on a page, to create a list, for example. <a href="#">[details]</a></p>
5	Calculation fields that subtotal the quantity and total price line items in the table above. <a href="#">[details]</a>
6	Although this is a summary field of the entire invoice amount, the content comes directly from a Salesforce data field and is not calculated on the template.
7	A conditional statement (IF statement) that checks whether the invoice total is over a specified amount and, if so, to add text related to free shipping. This text does not appear for invoices where the total is below the specified amount. <a href="#">[details]</a> This is an example only. Do NOT use double brackets with IF statements.
8	The page footer repeats across pages.

When merged, the placeholder merge fields are replaced with data from the Salesforce record. This merged template appears as below:

			<b>INVOICE</b>																																																																																				
																																																																																							
Violet Duke Director of Procurement (92) 841-8042 mgdo.nov.14@gmail.com	ABC Distribution Company 123 Main St. Denver, CO 80204 USA	Payment Due Date: January 29, 2015  Total Amount Due: \$25,000.00																																																																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008080; color: white;"> <th colspan="4">Widgets</th> </tr> <tr style="background-color: #cccccc;"> <th>PRODUCT</th> <th>SALES PRICE</th> <th>QTY</th> <th>TOTAL PRICE</th> </tr> </thead> <tbody> <tr> <td>Basic Widget</td> <td>\$50.00</td> <td>25</td> <td>\$1,250.00</td> </tr> <tr> <td>Standard Widget</td> <td>\$100.00</td> <td>15</td> <td>\$1,500.00</td> </tr> <tr> <td>Advanced Widget</td> <td>\$250.00</td> <td>10</td> <td>\$2,500.00</td> </tr> <tr> <td>Super Awesome Widget</td> <td>\$1,000.00</td> <td>3</td> <td>\$3,000.00</td> </tr> <tr> <td style="text-align: right;"><b>Subtotal</b></td> <td></td> <td><b>53</b></td> <td><b>\$8,250.00</b></td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008080; color: white;"> <th colspan="4">Software</th> </tr> <tr style="background-color: #cccccc;"> <th>PRODUCT</th> <th>SALES PRICE</th> <th>QTY</th> <th>TOTAL PRICE</th> </tr> </thead> <tbody> <tr> <td>Sales Enablement Suite License</td> <td>\$100.00</td> <td>10</td> <td>\$1,000.00</td> </tr> <tr> <td>Customer Success Suite License</td> <td>\$75.00</td> <td>10</td> <td>\$750.00</td> </tr> <tr> <td>Inventory Management Suite License</td> <td>\$50.00</td> <td>10</td> <td>\$500.00</td> </tr> <tr> <td style="text-align: right;"><b>Subtotal</b></td> <td></td> <td><b>30</b></td> <td><b>\$2,250.00</b></td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008080; color: white;"> <th colspan="4">Professional Services</th> </tr> <tr style="background-color: #cccccc;"> <th>PRODUCT</th> <th>SALES PRICE</th> <th>QTY</th> <th>TOTAL PRICE</th> </tr> </thead> <tbody> <tr> <td>Gold Consulting Plan</td> <td>\$5,000.00</td> <td>1</td> <td>\$5,000.00</td> </tr> <tr> <td style="text-align: right;"><b>Subtotal</b></td> <td></td> <td><b>1</b></td> <td><b>\$5,000.00</b></td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008080; color: white;"> <th colspan="4">Training</th> </tr> <tr style="background-color: #cccccc;"> <th>PRODUCT</th> <th>SALES PRICE</th> <th>QTY</th> <th>TOTAL PRICE</th> </tr> </thead> <tbody> <tr> <td>25+ User Training</td> <td>\$8,000.00</td> <td>1</td> <td>\$8,000.00</td> </tr> <tr> <td style="text-align: right;"><b>Subtotal</b></td> <td></td> <td><b>1</b></td> <td><b>\$8,000.00</b></td> </tr> </tbody> </table>				Widgets				PRODUCT	SALES PRICE	QTY	TOTAL PRICE	Basic Widget	\$50.00	25	\$1,250.00	Standard Widget	\$100.00	15	\$1,500.00	Advanced Widget	\$250.00	10	\$2,500.00	Super Awesome Widget	\$1,000.00	3	\$3,000.00	<b>Subtotal</b>		<b>53</b>	<b>\$8,250.00</b>	Software				PRODUCT	SALES PRICE	QTY	TOTAL PRICE	Sales Enablement Suite License	\$100.00	10	\$1,000.00	Customer Success Suite License	\$75.00	10	\$750.00	Inventory Management Suite License	\$50.00	10	\$500.00	<b>Subtotal</b>		<b>30</b>	<b>\$2,250.00</b>	Professional Services				PRODUCT	SALES PRICE	QTY	TOTAL PRICE	Gold Consulting Plan	\$5,000.00	1	\$5,000.00	<b>Subtotal</b>		<b>1</b>	<b>\$5,000.00</b>	Training				PRODUCT	SALES PRICE	QTY	TOTAL PRICE	25+ User Training	\$8,000.00	1	\$8,000.00	<b>Subtotal</b>		<b>1</b>	<b>\$8,000.00</b>
Widgets																																																																																							
PRODUCT	SALES PRICE	QTY	TOTAL PRICE																																																																																				
Basic Widget	\$50.00	25	\$1,250.00																																																																																				
Standard Widget	\$100.00	15	\$1,500.00																																																																																				
Advanced Widget	\$250.00	10	\$2,500.00																																																																																				
Super Awesome Widget	\$1,000.00	3	\$3,000.00																																																																																				
<b>Subtotal</b>		<b>53</b>	<b>\$8,250.00</b>																																																																																				
Software																																																																																							
PRODUCT	SALES PRICE	QTY	TOTAL PRICE																																																																																				
Sales Enablement Suite License	\$100.00	10	\$1,000.00																																																																																				
Customer Success Suite License	\$75.00	10	\$750.00																																																																																				
Inventory Management Suite License	\$50.00	10	\$500.00																																																																																				
<b>Subtotal</b>		<b>30</b>	<b>\$2,250.00</b>																																																																																				
Professional Services																																																																																							
PRODUCT	SALES PRICE	QTY	TOTAL PRICE																																																																																				
Gold Consulting Plan	\$5,000.00	1	\$5,000.00																																																																																				
<b>Subtotal</b>		<b>1</b>	<b>\$5,000.00</b>																																																																																				
Training																																																																																							
PRODUCT	SALES PRICE	QTY	TOTAL PRICE																																																																																				
25+ User Training	\$8,000.00	1	\$8,000.00																																																																																				
<b>Subtotal</b>		<b>1</b>	<b>\$8,000.00</b>																																																																																				
Conga Invoice for ABC Distribution Company - Dec 30, 2014 Prepared by: MGet / Owner: MGet			Page 1 of 2																																																																																				

			<b>INVOICE</b>																
																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #008080; color: white;"> <th colspan="4">Implementation Services</th> </tr> <tr style="background-color: #cccccc;"> <th>PRODUCT</th> <th>SALES PRICE</th> <th>QTY</th> <th>TOTAL PRICE</th> </tr> </thead> <tbody> <tr> <td>Remote Implementation Support</td> <td>\$1,500.00</td> <td>1</td> <td>\$1,500.00</td> </tr> <tr> <td style="text-align: right;"><b>Subtotal</b></td> <td></td> <td><b>1</b></td> <td><b>\$1,500.00</b></td> </tr> </tbody> </table>				Implementation Services				PRODUCT	SALES PRICE	QTY	TOTAL PRICE	Remote Implementation Support	\$1,500.00	1	\$1,500.00	<b>Subtotal</b>		<b>1</b>	<b>\$1,500.00</b>
Implementation Services																			
PRODUCT	SALES PRICE	QTY	TOTAL PRICE																
Remote Implementation Support	\$1,500.00	1	\$1,500.00																
<b>Subtotal</b>		<b>1</b>	<b>\$1,500.00</b>																
<b>INVOICE TOTAL DUE:</b>			<b>\$25,000.00</b>																
<p><b>Congratulations - your deal is so big that it qualifies for free shipping! Please use offer code BIGDEAL when submitting your payment.</b></p> <p><b>Payment Details</b></p> <p>Please remit payment to:</p> <p style="margin-left: 40px;">                     World's Away Inc.                      Ben Allen                      385 Interlocken Crescent, Suite 600                      Broomfield, CO 80021                 </p>																			

## Dynamic Date Formats in Word

This article describes how to adjust date merge fields in Conga Composer Word Templates. You can adjust a date merge field by including the ADJUST: prefix. For example, you could display the date 30 days past an Opportunity Close Date.

The ADJUST: prefix works with any Salesforce date field or the Conga TODAY or NOW fields from the System dataset, which is provided by default in the Template Builder.

Table 1. ADJUST

<b>Syntax</b>	<code>{{ADJUST:MyDateField:Year:Month:Day:Hour:Minute:Second}}</code>
<b>Values</b>	<ul style="list-style-type: none"> <li>• MyDateField is TODAY, NOW, or the name of your date field; for example, Close_Date.</li> <li>• Year is the number of years you want to adjust the date.</li> <li>• Month is the number of months you want to adjust the date.</li> <li>• Day is the number of days you want to adjust the date.</li> <li>• Hour is the number of hours you want to adjust the date.</li> <li>• Minute is the number of minutes you want to adjust the date.</li> <li>• Second is the number of seconds you want to adjust the date.</li> </ul>
<b>Notes</b>	<p>You must use all of the data elements (years, months, days, hours, minutes, seconds). For any element that you do not want to adjust, insert a zero (0).</p> <p>Dates can be adjusted for the future or past with a positive or negative integer, respectively.</p> <p>You can also use an appearance parameter to edit how the date field will appear. However, appearance parameters work only with Master fields.</p>
<b>Examples</b>	<ul style="list-style-type: none"> <li>• <code>{{ADJUST:Close_Date:0:0:30:0:0:0}}</code> - The value of the Close_Date field is adjusted to 30 days in the future.</li> <li>• <code>{{ADJUST:Created_Date:0:-6:0:0:0:0}}</code> - The value of the Created_Date field is adjusted to 6 months ago.</li> <li>• <code>{{ADJUST:Close_Date:0:0:30:0:0:0 \@ "MMMM d, yyyy"}}</code> - The value of the Close_Date field is adjusted to 30 days in the future, and an appearance parameter formats the result.</li> </ul>



## Format Numbers as Text in Word

Conga Composer can apply formatting options to write out numbers within fields as ordinal text.

These features return English values only but are helpful for currencies and numbers that need to be written out, such as on a check or in a contract.

Format Code	Description
\* CardText	<p>Formats a number as English words.</p> <p><code>{{Opportunity_Amount \* CardText}}</code> Returns five thousand two hundred forty five.</p> <p>To capitalize each word, also use \* FirstCap.</p>
\*DollarText	<p>Formats a number as English words in a bank-check style.</p> <p><code>{{Opportunity_Amount \*DollarText}}</code> Returns five thousand two hundred eighty and 00/100.</p> <p>To capitalize each word, also use \*FirstCap.</p> <p>Dollar amount must be less than \$999,999.99.</p>
\* Ordinal	<p>Formats an integer as an ordinal number, such as 1st or 2nd.</p> <p>Can also be used with number fields or the "day" or "month" portion of a Date field.</p> <p><code>{{Opportunity_CloseDate \@ "MMMM"}}</code> the <code>{{Opportunity_CloseDate \@ "d" \* Ordinal}}</code>, <code>{{Opportunity_CloseDate \@ "yyyy"}}</code> returns March the 14th, 2014.</p>

Format Code	Description
\* FirstCap	<p>Capitalizes the first letter of the first word.</p> <p><code>{{ACCOUNT_NUMBEROFEMPLOYEES \* CardText \* FirstCap}}</code> returns Five thousand two hundred forty five.</p> <p>Multiple picture switches may be applied to one merge field.</p> <p>The example above shows both the CardText and FirstCap picture switches. When using FirstCap in conjunction with CardText, there cannot be a space between \* and CardText. Example: <code>\*FirstCap \*CardText</code></p>

## Capitalization and Number Format Field Switches for Word Templates

The following is a list of some of the available format field switches for use in Word@ templates. These are helpful when text or numbers need to be displayed and in a specific manner; for example, when a currency amount needs to be spelled out as on a check (ex: Five hundred twenty-one instead of \$521.00).

For more information on Microsoft Office switches and how to display field results please see Microsoft Office's help article on [Format \(\\*\\) Field Switches](#).

### Capitalization Formats

\* Caps	<p>Capitalizes the first letter of each word.</p> <p>For example, <code>{ MERGEFIELD CONTACT_FULLNAME \* Caps }</code> displays "Julie Tanner" even if the value is stored in Salesforce as lowercase letters (julie tanner).</p>
\* FirstCap	<p>Capitalizes the first letter of the first word.</p> <p>For example, <code>{ MERGEFIELD CONTACT_DESCRIPTION \* FirstCap }</code> displays "Loves chocolate chip cookies."</p>

\* Upper	<p>Displays all letters in uppercase even if the value is stored in Salesforce as lowercase letters (julie tanner) or proper case (Julie Tanner).</p> <p>For example, <code>{ MERGEFIELD CONTACT_FULLNAME \* Upper }</code> displays "JULIE TANNER."</p>
\* Lower	<p>Displays all letters in lowercase.</p> <p>For example, <code>{ MERGEFIELD CONTACT_DESCRIPTION \* Lower }</code> displays "loves chocolate chip cookies."</p>

## Number Formats

\* CardText	<p>Displays results as cardinal text. The result is formatted in lowercase letters unless you add a format switch to specify a different capitalization.</p> <p>For example, <code>{ MERGEFIELD OPPORTUNITY_AMOUNT \* CardText }</code> displays "seven hundred ninety", and <code>{ MERGEFIELD OPPORTUNITY_AMOUNT \* CardText \* Caps }</code> displays "Seven Hundred Ninety".</p>
\*DollarText	<p>Displays results as cardinal text. Microsoft Word inserts "and" at the decimal place and displays the first two decimals (rounded) as Arabic numerators over 100. The result is formatted in lowercase letters unless you add a format switch to specify a different capitalization.</p> <p>For example, an Opportunity Amount of \$14.55 <code>{ MERGEFIELD OPPORTUNITY_AMOUNT \*DollarText }</code> displays "fourteen and 55/100".</p>
\* OrdText	<p>Displays results as ordinal text. The result is formatted in lowercase letters unless you add a format switch to specify a different capitalization.</p> <p>For example, an Opportunity Close Date of August 21, 2014 <code>{ MERGEFIELD OPPORTUNITY_CLOSEDATE \@ "d" \* OrdText }</code> displays "twenty-first".</p>

\* Ordinal	<p>Displays results as ordinal Arabic numerals.</p> <p>For example, { MERGEFIELD OPPORTUNITY_CLOSEDATE \@ "d" \* Ordinal } displays "21st".</p>
------------	---

## Embedding Graphic Files in Merge Templates in Word

You can embed graphic files (such as your company's logo) in your merge templates to improve its appearance.

Please follow these rules when embedding graphics in your templates:

- Graphic files must be JPG or PNG files - certain other file types may prevent proper merging of data or may not merge at all
- Use the smallest possible file size that will provide adequate resolution.
- In Conga Mail Merge situations, remember that graphics are included on every page, dramatically increasing the file size:
- Running a Conga Mail Merge solution to produce 1,000 documents for a template file of 250 KB (including graphics) would result in a total merge output file size of 250 MB.
- You can reduce the file size of your graphic images by using a graphics utility program, such as Adobe Photoshop, Paint.NET, or Image Resizer (part of the Microsoft PowerToys collection).

## Native Word Field Types

You can create a native Word field using two methods:

Create a native Word field by inserting a field and choosing the relevant type. To add a native field, on the Insert tab in Word, click **Quick Parts** → **Field**.

### Create a Text-Based Field with the @ Prefix:

Conga Composer Release 8 uses text-based merge fields (ex: `{{Field_Name}}` ), so when using native Word field types you will need to include the at sign (@) at the beginning of the field name. This indicates to Conga Composer (from within a text-based merge field) that you want a native Word field rather than the usual merge field.

`{{@Title}}` results in a native Word "Title" field.

`{{Title}}` results in a merge field called "Title" (this is the field name displayed in Template Builder).

Below is the list of supported native Word fields. Each requires the at sign (@) at the beginning of the text-based merge field (ex: `{{@ASK}}` )

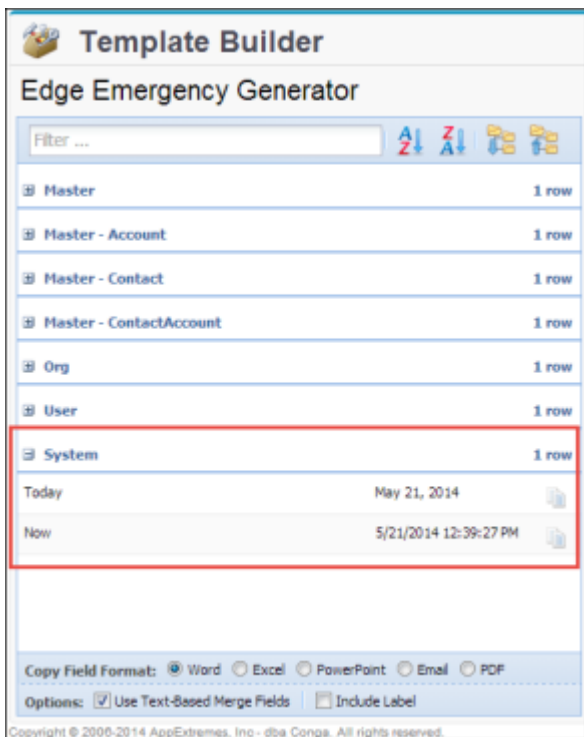
ADDRESSBLOCK	EDITTIME	MERGESEQ	SEQ
ADVANCE	EQ	NEXT	SET
ASK	FILESIZE	NEXTIF	SKIPIF
AUTHOR	FILLIN	NOTEREF	STYLEREF
AUTONUM	GOTOBUTTON	NUMCHARS	SUBJECT
AUTONUMLG	GREETINGLINE	NUMPAGES	SYMBOL
LAUTONUMOUT	HYPERLINK	NUMWORDS	TA
AUTOTEXT	IF	PAGE	TC
AUTOTEXTLIST	INCLUDEPICTURE	PAGEREF	TEMPLATE
BARCODE	INCLUDETEXT	PRINT	TIME
BIBLIOGRAPHY	INDEX	PRINTDATE	TITLE
BIDIOUTLINE	INFO	PRIVATE	TOA
CITATION	KEYWORDS	QUOTE	TOC
COMMENTS	LASTSAVEDBY	RD	USERADDRESS
COMPARE	LINK	REF	USERINITIALS
CREATEDATE	LISTNUM	REVNUM	USERNAME
DATE	MACROBUTTON	SAVEDATE	XE
DOCPROPERTY	MERGEFIELD	SECTION	
DOCVARIABLE	MERGEC	SECTIONPAGES	

**Examples:**

- `{{@SEQ OppLineItems}}`
- `{{@REF BookmarkName}}`

## System Fields in Word

There are two fields in the System dataset, available in the Conga Template Builder. These fields are provided by default (just like the Master, Org, and User datasets) to aid in template creation.



## Today

The System.Today field will display today's date and will not automatically update to the then-current today's date any time the merged Word output file is opened in the future. If you do have that requirement, we suggest using Word's date feature instead, which does automatically update each time the file is opened. Just like any other date field, this field can be formatted using date picture switches.

- `&=System.TODAY`
- `{{System.TODAY}}`

## Now

The System.Now field displays date and time. This field will not automatically update to the then-current today's date and time any time the merged Word output file is opened in the future. If you do have that requirement, we suggest using Word's date feature instead, which does automatically update each time the file is opened. Just like any other date field, this field can be formatted using date picture switches.

- `{{System.NOW}}`
- `&=System.NOW`

## Support for Text Boxes and Shapes in Word

Word's Text Box feature (not to be confused with a Composer TextBox merge field) and shapes are supported in Conga Composer release 8. You can merge both merge fields and static text into text boxes and shapes. Text boxes are great to use when you have data that needs to be in a very specific place in your template that is difficult to format as such with a table.

Complex functionality, such as multi-layered text boxes or combining text boxes and tables, is not suitable for use in text boxes.

Congratulations, `{{Account_Name}}`!

Unlike a table, a text box or shape will not dynamically grow to accommodate merged data. You'll need to appropriately size the text box or shape in the template.

## Unsupported Word Features

Unfortunately, Conga Composer currently won't work with every feature available in a Microsoft Word document. We encourage you to test your specific requirements with Conga. A few known limitations are listed in this article.

### Known Limitations

- FORMTEXT fields
- Office 2007/2010/2013 Digital Signatures
- Password protection ("Read Only Enforced" or "Read Only Recommended"). The Word file must be editable in order for Composer to merge data to it. To utilize password protection, please see [TextBox fields](#) and the `&WPD` (Word protected document) and `&WDPW` (Word protected document password) parameters.

- Track Revisions Enabled with unaccepted revisions. To resolve this, go to **Review** → **Changes** → **Accept All Changes**.
- The use of gradient backgrounds in Word templates.
- Nested IF statements, where the else value is another conditional IF statement work with Conga Composer

❗ Conga does not support nested IF statements in Microsoft Word. The recommended alternative to nested IF statements in Microsoft Word is to use the native IF and CASE Logical Functions in Salesforce formulas.

## Formatting Word Merge Fields with Picture Switches

Format merge data Word fields with picture switches or appearance parameters. Word provides several formatting switches (also called picture switches or appearance parameters) that affect the appearance of your merged data. This is invaluable, particularly for numbers and dates. You can use the [Template Builder](#) to add these picture switches to your merge fields so the merged data is correctly formatted in your output file.

You can use picture switches to format:

- [Date Values](#)
- [Number Values](#)
- [Numbers as Text](#)

You can manually add a picture switch to your field by simply typing it into your merge field in Word.

1. Click after the merge field name, but inside of the double curly braces.
2. Type a space and then type your picture switch.

Example: `{{ACCOUNT_CREATEDDATE \@ MM/dd/yy}}`

✔ To ensure that your data looks consistent in merged output, manually add a picture switch to all date and number fields.

## Format Number Values in Word

The number formatting picture switch begins with `\#`, followed by any combination of the following parameters. Please see the examples following this table. When Conga Composer populates a numeric value, the digit grouping symbol (the thousands separator)



is defined according to your Locale settings (as defined in Salesforce's **Setup** → **Personal Information**) or according to the Culture or CurrencyCulture Conga Composer parameters.

Despite the cultural settings that govern the output from Conga Composer, within an appearance parameter, you must use a comma (",") for the digit grouping symbol (for thousands) and a period (".") as the decimal symbol. You cannot swap the comma and period (American style) with a picture switch. If your goal is international currency behavior, please see [Dynamic Currency Formats in Word](#).

Parameter	Description
0 (zero)	<p>Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a 0 (zero).</p> <p><code>{{Points \# 00.00}}</code> appears as 09.00.</p>
#	<p>Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, nothing appears in that place.</p> <p><code>{{Cost \# \$### }}</code> appears as \$15.</p> <p>You can format phone number as well using this merge field <code>{{merge_field \# ###-###-####}}</code></p>
, (digit grouping symbol)	<p>Separates a series of three digits.</p> <p>The digit grouping symbol is always entered as a comma into the formatting code, regardless of the country's cultural preference. Conga Composer will format the output according to the cultural preference.</p> <p><code>{{= NetProfit \# \$#,###,###}}</code> appears as \$2,456,800.</p>
%, \$, *, and so on	<p>Includes the specified character in the result.</p> <p><code>{{= NetProfit \# "###%"}}</code> appears as 33%.</p>

Parameter	Description
[positive value]; [negative value]	Specifies different number formats for results with positive or negative values.  <pre> {{Sales95 \# "\$#,##0.00;-\$#,##0.00"}} </pre> Positive values appear as \$1,245.65. Negative values appear as -\$1,245.65.  Negative numbers <pre> {{Widgets \# "0#,##0.00;-0#,##0.00"}} </pre>
Currency	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol, and two decimal places  <pre> {{OPPORTUNITY_AMOUNT \# Currency}} </pre> Positive values appear as \$1,245.65. Negative values appear as (\$1,245.65).
Currency0  Also available:  Currency1, 2, 3, 4	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol, and zero decimal places (or 1 decimal place, 2 decimal places, 3, or 4 if using Currency1, Currency2, Currency3, etc.) Currency decimal only works up to 4***  <pre> {{OPPORTUNITY_AMOUNT \# Currency0}} </pre>
Number	Formats the value according to the Salesforce Locale, excluding a currency symbol, but including a digit grouping symbol, and two decimal places.  <pre> {{OPPORTUNITY_AMOUNT \# Number}} </pre>
Number0, 1, 2, 3, 4	Formats the value according to the Salesforce Locale, including a digit grouping symbol, but excluding two decimal places and a currency symbol. (or with 1 decimal place, 2, 3 or 4)  <pre> {{OPPORTUNITY_AMOUNT \# Number0}} </pre> Can also be used with SUM fields.

Parameter	Description
Percent	<p>Formats the value according to the Salesforce Locale, by multiplying the value by 100 and including a digit grouping symbol, two decimal places, and a percent sign.</p> <pre>{{OPPORTUNITY_PROBABILITY \# Percent}}</pre> <p>Cannot be used with SUM fields.</p>

## Examples

Here are some common examples of the number appearance parameter. Assume that the field we are formatting is OPPORTUNITY\_AMOUNT and that it contains 5280.25 and the Salesforce Locale is “English (United States)”.

Example	Output
<pre>{{OPPORTUNITY_AMOUNT \# #,##0}}</pre>	5,280
<pre>{{OPPORTUNITY_AMOUNT \# \$#,##0}}</pre>	\$5,280
<pre>{{OPPORTUNITY_AMOUNT \# \$#,##0.00}}</pre>	\$5,280.25
<pre>{{OPPORTUNITY_AMOUNT \# Currency}}</pre>	\$5,280.25

Now assume we are formatting the same field value as above, however, the Salesforce Locale is “German (Germany)”.

Example	Output
<pre>{{OPPORTUNITY_AMOUNT \# #,##0}}</pre>	5.280
<pre>{{OPPORTUNITY_AMOUNT \# “#,##0 €”}}</pre>	5.280 €
The formatting code is enclosed in quotes because it contains spaces.	
<pre>{{OPPORTUNITY_AMOUNT \# #,##0.00}}</pre>	5.280,25

Example	Output
<code>{{OPPORTUNITY_AMOUNT \# Currency}}</code>	5.280,25 €

## Dynamic Currency Formats in Word

You can format currency values based on a value stored in a Salesforce field.

To format currency values based on a Salesforce field:

1. On a Salesforce object, create a picklist field and populate it with culture code values. For a list of available culture codes, see [Supported Culture Codes](#).
2. In the Conga Composer button URL, insert the CurrencyCulture parameter and the related object and field in the following format: `&CurrencyCulture={!Object.Field}`

**Example:** For an Opportunity object containing a field named

Culture: `&CurrencyCulture={!Opportunity.Culture_c}`

3. Format the currency fields in your template using the “\# Currency” picture switch.

**Example Merge Field:** `{ MERGEFIELD Opportunity_LineItem_SalesPrice \# Currency }`

Disable Wrap Text in cells that use Currency formatting to keep the value and the symbol on the same line. Navigate to **Table** → **Table Properties** → **Cell** → **Options** and clear the Wrap Text check box.

## Detail Data in Word

Conga Composer can dynamically display repeating data from a Salesforce report, SOQL query, or Related List inside a merge template.

### Using Detail Data in Word

Conga Composer can merge data from child objects of your master object by using Conga Queries. This data is called detail data.

You can retrieve multiple records from a child object (such as quote line items) and then designate an area on your template that will dynamically accommodate the number of line items (or other child records) that are present. Merge fields and any other text placed within a detail region will automatically repeat for each row of detail data.

Detail regions can be used in almost any place in a template and can include all kinds of elements. Some examples:

- A single table row – the row repeats for each record

- An entire table – the table repeats for each record
- A paragraph – the paragraph repeats for each record
- Across pages – the page break repeats for each record

To see syntax examples, reference [Repeating Data in Word](#).

When creating a detail region, make sure to follow these rules:

- `TableStart` and `TableEnd` tags (TS/TE tags) must be inside the same section in the document. If you put a section break inside these tags, the template will not merge.
- TS/TE tags must refer to the same dataset. Examples:
  - `TableStart:data` `ACCOUNT_NAME` `TableEnd:data` will work.
  - `TableStart:data` `ACCOUNT_NAME` `TableEnd:otherdata` will not work, the template will not merge.
- Detail regions must contain a pair of matching TS/TE tags. If TS/TE tags do not match, the template will not merge.
- If used within a table, TS/TE must be on the same row or within the same cell. They cannot be used across multiple rows in a table. Wrap the whole table in TS/TE tags if you need to repeat multiple rows at a time.
- Nested detail regions are only supported with the [use of nested queries](#).

## Repeating Data in Word

Composer can display repeating data (aka child object data or detail data) in a variety of ways. You can also group detail data and apply conditional logic to hide tables or sections that do not contain data.

### Repeating Rows in a Table

Using `TableStart` and `TableEnd` tags (TS/TE tags), you can set up a table with a row that will dynamically repeat for each child record returned in the query. Once a query has been added to your Composer Solution, Template Builder will give you the syntax needed to add to your table. As an example, imagine you have a query to pull Quote Line Items and you want to display those in a pricing table. Your Word template would look like this:

Product	Quantity	List Price	Total Price
<code>TableStart:QLI</code> <code>PRODUCT_NAME</code>	<code>QUANTITY</code>	<code>UNIT_PRICE</code>	<code>TOTAL_PRICE</code> <code>TableEnd:QLI</code>

`{{TableStart}}` initiates the set of repeating data. This field is placed in the first cell in the table. `{{TableEnd}}` closes the set of repeating data. This field is placed at the end of the first row of data in the table. The TS/TE tags must reference the alias of the query as shown in the Template Builder.

For more information, see [Using Detail Data in Word](#).

## Groups of Repeating Data Tables

Conga Composer includes two special tags to enable grouping related data into tables:

- TableGroup
- TableGroupStart

### TableGroup

Using TableGroup, you can group repeating data on a specific field in your dataset.

Examples:

- Group Quote Line Items by Product Category
- Group Opportunities by Owner
- Group Revenue by Customer

This tag is placed in the first row of the table and can show or hide the value being grouped upon. For more information, see [Organize Repeating Data Tables into Groups](#).

### TableGroupStart

Similarly to TableGroup, you can use TableGroupStart to group on a value in a dataset. The difference is that TableGroupStart allows you to place merge fields, static text, or tables between the TableGroupStart and TableGroupEnd tags.

For more information, see [Grouping Repeating Data with TableGroupStart](#).

## Hiding Tables with No Information

You can hide a table using TableHide or hide a table row using Row Burner. For example, you might have a pricing table to display One-Time products, but you only want the table to show if there are One-Time products included in your Quote. Or you might have a table row to show the Quote overall discount percentage, but you only want that row to show if the discount is greater than zero.

For more information, reference these topics:


- [TableHide](#) or [Conditional TableHide](#)

- [Row Burner](#)

## Grouping Repeating Data with Additional Information in Word using TableGroupStart and TableGroupEnd

TableGroupStart adds the benefit of displaying multiple field values within the heading. In addition to TableGroup, you can also group on a value in a table by including a pair of special merge fields, `{{TableGroupStart:DatasetName:FieldName}}` and `{{TableGroupEnd:DatasetName:FieldName}}`.

TableGroupStart is very similar to TableGroup in that it lets you group on a field in your detail data; however, TableGroupStart adds the benefit of displaying multiple field values within the heading.

 Do not mix and match the Tablegroupstart and End method with the Tablegroup method. You must pick one method for your template. Tablegroup does not work in Mail Merge.

 See [Add Page Breaks to Repeating Tables of Data in Word](#).

<b>Syntax</b>	<code>{{TableGroupStart:DatasetName:FieldName}}</code> <code>{{TableGroupEnd:DatasetName:FieldName}}</code>
<b>Values</b>	<p>DatasetName corresponds to the name of the dataset from the Template Builder window.</p> <p>FieldName corresponds to a field name displayed on the Template Builder window and is the field on which you want to group the data.</p>
<b>Notes</b>	<p>These fields must exist in a single table cell.</p> <p>The FieldName in the <code>{{TableGroupStart}}</code> and <code>{{TableGroupEnd}}</code> fields must not contain hyphens (-) or colons (:). If your FieldName does contain these, use the SCHP parameter.</p> <p>The field names for <code>{{TableStart}}</code> and <code>{{TableEnd}}</code> cannot be abbreviated and both must be placed within the same row of the detail table.</p>

### Example Usage:

<b>Price Book</b> Prices valid for 30 days from quote date.		
<b>Product</b>	<b>Quantity</b>	<b>Total Price</b>
GenWatt Diesel 1000kW (Products)	2	200000
GenWatt Gasoline 2000W (Products)	1	150000

returns upon merge:

Products Price Book		
Prices valid for 30 days from quote date.		
Product	Quantity	Total Price
GenWatt Diesel 1000kW (Products)	2	200000
GenWatt Gasoline 2000W (Products)	1	150000

Products Price Book		
Prices valid for 30 days from quote date.		
Product	Quantity	Total Price
Installation: Industrial - Medium (Services)	1	50000
SLA: Silver (Services)	1	20000

## Disable Row Numbers in Word

To disable the `ROW_NUMBER` field in all detail datasets use the parameter `&DisableRowNums=1`.

This parameter prevents the `ROW_NUMBER` field from displaying in report, query, Opportunity Line Items or Quote Line Items datasets.

## Numbering Table Rows in Word

Microsoft Word can automatically number rows in a table. This is particularly useful in detail regions, where you may have large number of rows of detail data.

Row Numbers	Favorite Color
1	Red



Row Numbers	Favorite Color
2	Blue
3	Green

There are two ways to accomplish row numbering:

- [Use the Row Number field from the report or query dataset](#)
- [Create a Sequence field](#)

## Format a Table with Alternate Row Colors (Word 2007/2010 only)

Create a table for use as a detail region.

On **Table Tools** → **Design**, select a Table Style.

## Improved Support for Tables Using Text Wrapping in Word

Conga Composer can merge data to Word tables that utilize “Around” for text wrapping (Table Properties → Table → Text Wrapping → Around).

## Repeating Data within a Paragraph in Word

You may place a detail region within a paragraph. You may use a detail region only once within a paragraph.

**Example:**

{{ACCOUNT\_NAME}}

Thank you for your interest, {{TableStart:Contacts}}{{First\_Name}}{{TableEnd:Contacts}}.

If the source report or query returns multiple results, the entire paragraph will repeat for however many results are returned in the report/query. When using a Detail Region in this way, limit the report’s row count to one, to avoid repeating of data within the paragraph.

## Using Detail Regions with IF Statements

Using Detail Regions with If Statements.

There are two scenarios to consider when using Detail Regions with IF statements:

- **Scenario 1:** All merge fields in the IF statement come from detail data
- **Scenario 2:** Only some merge fields in the IF statement come from detail data

In both scenarios, you can use Word merge fields or text-based merge fields, with one exception: the IF statement itself must be a Word merge field. This means that all the fields inside or around the IF statement can be either Word or text-based merge fields, but the IF statement itself *must* be a Word merge field. Examples using both types of syntax are displayed below.

## Scenario 1: All Merge Fields in the IF Statement Come from Detail Data

In Scenario 1, where all merge fields in the IF statement come from detail data, enclose the entire IF statement with TableStart and TableEnd fields and leave them outside the IF statement. Structuring the IF statement this way prevents the IF statement from breaking if your query returns no data.

The example below uses a query with an alias of "Case" and the TableStart and TableEnd Word merge fields are toggled open.

```
{ MERGEFIELD TableStart:Case \* MERGEFORMAT }{ IF "<<CASE_STATUS>>" = "Closed"
"<<CLOSE_DATE>>" "Open" }{ MERGEFIELD TableEnd:Case \* MERGEFORMAT }
```

The same example IF statement is displayed below with text-based merge fields

```
{{TableStart:Case}}{ IF "{{CASE_STATUS}}" = "Closed" "{{CLOSE_DATE}}" "Open" }{{TableEnd:Case}}
```

The entire IF statement is enclosed in between the TableStart and TableEnd merge fields because all of the fields (Case Status and Close Date) are retrieved from the Case query.

## Scenario 2: Only Some Merge Fields in the IF Statement Come from Detail Data

In Scenario 2, where one merge field in the IF statements is retrieved from detail data, enclose only that field in between the TableStart and TableEnd fields. Additionally, ensure that the field has its own line.

```
{ IF "<<OPPORTUNITY_AMOUNT>>" > "100000" "This is a big deal Opportunity! Here are the
product prices:<<TableStart:OppLineItems>>
<<OPPORTUNITY_LINEITEM_LISTPRICE>><<TableEnd:OppLineItems>>" "" }
```

The example below displays the same IF statement, but with text-based merge fields.

```
{ IF "{{OPPORTUNITY_AMOUNT}}" > "100000" "This is a big deal Opportunity! Here are the
product prices:{{TableStart:OppLineItems}}
{{OPPORTUNITY_LINEITEM_LISTPRICE}}{{TableEnd:OppLineItems}}" "" }
```

Consider an example where there are two different detail datasets utilized in a single IF statement. The syntax for this use case is an extension of the example above, where each dataset requires its own line.

```
{ IF "<<OPPORTUNITY_CLOSEDATE>>" = "<<TODAY>>" "<<TableStart:Case>>
<<Close_Date>><<TableEnd:Case>>"
"<<TableStart:Opportunities>>
<<Close_Date>><<TableEnd:Opportunities>>" }
```

**⚠ Nested IF statements in Word, where the *else* value is another IF statement, work with Conga Composer but are not recommended. Conga Technical Support does not support nested IF statements. It is highly recommended to evaluate nested IF statement logic using Salesforce formula fields instead of IF statements, and then merging the formula field values into the document.**

There are other methods to conditionally hide content based on Salesforce data. For more information, see [Conditionally Remove a Row in a Table that Has No Data](#).

## Repeating Data in Word - TableStart and TableEnd

Use TableStart and TableEnd (TS/TE) in Composer to merge any data from queries you've added to your solution. Anything placed between the TS/TE tags, including tables, graphics, text, and merge fields, is automatically repeated for each row of data in the corresponding dataset. TS/TE can be placed anywhere in the document and can be used outside of a table.

<b>Syntax</b>	<code>{{TableStart:Dataset}}{{TableEnd:Dataset}}</code>
<b>Values</b>	<code>Dataset</code> is the alias you gave the query when you added it to your Composer solution. This name will also be displayed in Template Builder

<b>Notes</b>	<ul style="list-style-type: none"> <li>• Use the same alias in a pair of TS/TE tags. Mismatched aliases will cause the merge to error out.</li> <li>• TS/TE tags can be nested:                     <pre style="margin-left: 20px;">{{TableStart:Account}}{{ACCOUNT_NAME}}                     {{TableStart:Oppty}}{{OPPORTUNITY_NAME}}{{TableEnd:Oppty}}                     {{ACCOUNT_BILLING_ADDRESS}}{{TableEnd:Account}}</pre> </li> <li>• Insert a blank line before TableEnd to display the next record on a new line.</li> <li>• Insert a page break before TableEnd to display the next record on a new page.</li> <li>• If you place a TableStart or TableEnd on a line by itself, it will leave some blank space on the merged document. You can minimize the amount of vertical white space that gets left by reducing the font size.</li> <li>• (Optional) Use red text for TS/TE so that they stand out and are easy to recognize. Text color does not change output functionality.</li> </ul>																								
<b>Usage</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Title</th> <th style="text-align: left;">Phone</th> <th style="text-align: left;">Email</th> </tr> </thead> <tbody> <tr> <td> <pre style="font-family: monospace; font-size: 0.9em;">{{TableStart:Contacts}} {{First_Name}} {{Last_Name}}</pre> </td> <td> <pre style="font-family: monospace; font-size: 0.9em;">{{Title}}</pre> </td> <td> <pre style="font-family: monospace; font-size: 0.9em;">{{Phone}}</pre> </td> <td> <pre style="font-family: monospace; font-size: 0.9em;">{{Email}} {{TableEnd:Contacts}}</pre> </td> </tr> </tbody> </table> <p>Returns on merge:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Title</th> <th style="text-align: left;">Phone</th> <th style="text-align: left;">Email</th> </tr> </thead> <tbody> <tr> <td>Robert Blalock</td> <td>Account Manager</td> <td>(555) 555-1212</td> <td><a href="mailto:rblalock@supremetech.org">rblalock@supremetech.org</a></td> </tr> <tr> <td>Cassandra Wilson</td> <td>Sales Representative</td> <td>(555) 555-1414</td> <td><a href="mailto:cwilson@supremetech.org">cwilson@supremetech.org</a></td> </tr> <tr> <td>Fred Smith</td> <td>Mail Clerk</td> <td>(555) 555-0006</td> <td><a href="mailto:fsmith@supremetech.org">fsmith@supremetech.org</a></td> </tr> </tbody> </table>	Name	Title	Phone	Email	<pre style="font-family: monospace; font-size: 0.9em;">{{TableStart:Contacts}} {{First_Name}} {{Last_Name}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Title}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Phone}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Email}} {{TableEnd:Contacts}}</pre>	Name	Title	Phone	Email	Robert Blalock	Account Manager	(555) 555-1212	<a href="mailto:rblalock@supremetech.org">rblalock@supremetech.org</a>	Cassandra Wilson	Sales Representative	(555) 555-1414	<a href="mailto:cwilson@supremetech.org">cwilson@supremetech.org</a>	Fred Smith	Mail Clerk	(555) 555-0006	<a href="mailto:fsmith@supremetech.org">fsmith@supremetech.org</a>
Name	Title	Phone	Email																						
<pre style="font-family: monospace; font-size: 0.9em;">{{TableStart:Contacts}} {{First_Name}} {{Last_Name}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Title}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Phone}}</pre>	<pre style="font-family: monospace; font-size: 0.9em;">{{Email}} {{TableEnd:Contacts}}</pre>																						
Name	Title	Phone	Email																						
Robert Blalock	Account Manager	(555) 555-1212	<a href="mailto:rblalock@supremetech.org">rblalock@supremetech.org</a>																						
Cassandra Wilson	Sales Representative	(555) 555-1414	<a href="mailto:cwilson@supremetech.org">cwilson@supremetech.org</a>																						
Fred Smith	Mail Clerk	(555) 555-0006	<a href="mailto:fsmith@supremetech.org">fsmith@supremetech.org</a>																						

## Organize Repeating Data Tables into Groups in Word

You can group data on a column from a detail dataset by including the merge field `{{TableGroup:DatasetName:FieldName:Show}}` in the first row of a table.

Do not mix and match the Tablegroupstart or End method with the Tablegroup method. You must pick one method for your template.

Syntax	<pre style="font-family: monospace; font-size: 0.9em;">{{TableGroup:DatasetName:FieldName:Show}}</pre> <p style="text-align: center; margin: 5px 0;">OR</p> <pre style="font-family: monospace; font-size: 0.9em;">{{TableGroup:DatasetName:FieldName:Hide}}</pre>
--------	--

<p>Values</p>	<p>DatasetName is the name of the dataset from the Template Builder window.</p> <p>FieldName is the name of the field from the Template Builder window, and is the field that creates the groups.</p> <p>Show indicates to display the value of the field for each grouping. OrHide hides the values in the merged output file.</p> <p>The Hide value is useful when grouping on a field with a value you don't want to display, such as a record ID.</p>																																	
<p>Notes</p>	<p>The FieldName in the TableGroup field must not contain hyphens (-) or colons (:). If your field name contains these, use the SCHP parameter in your button URL to strip out punctuation from the field name.</p> <p>The field names for TableStart and TableEnd must not be abbreviated and must be on the same row of the detail table.</p>																																	
<p>Usage</p>	<table border="1" data-bbox="507 1003 1407 1193"> <thead> <tr> <th colspan="3" style="text-align: center;">{{TABLEGROUP:OPPLINEITEMS:OPPORTUNITY_LINEITEM_PROD_FAMILY:SHOW}}</th> </tr> <tr> <th style="text-align: center;">Product</th> <th style="text-align: center;">Quantity</th> <th style="text-align: center;">Total Price</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM PRODUCTNAME}}</td> <td style="text-align: center;">{{OPPORTUNITY_LINEITEM_QUA NTITY}}</td> <td style="text-align: center;">{{OPPORTUNITY LINEITEM TOTALPRICE}} {{TABLEEND:OPPLINEITEMS}}</td> </tr> </tbody> </table> <p>Returns on merge</p> <table border="1" data-bbox="517 1317 1401 1597"> <thead> <tr> <th colspan="3" style="text-align: center;">Hardware</th> </tr> <tr> <th style="text-align: left;">Product</th> <th style="text-align: center;">Quantity</th> <th style="text-align: center;">Total Price</th> </tr> </thead> <tbody> <tr> <td>GenWatt Diesel 1000kW (Products)</td> <td style="text-align: center;">2</td> <td style="text-align: center;">200000</td> </tr> <tr> <td>GenWatt Gasoline 2000W (Products)</td> <td style="text-align: center;">1</td> <td style="text-align: center;">150000</td> </tr> </tbody> </table> <table border="1" data-bbox="517 1641 1401 1843"> <thead> <tr> <th colspan="3" style="text-align: center;">Hardware</th> </tr> <tr> <th style="text-align: left;">Product</th> <th style="text-align: center;">Quantity</th> <th style="text-align: center;">Total Price</th> </tr> </thead> <tbody> <tr> <td>Installation: Industrial - Medium (Services)</td> <td style="text-align: center;">1</td> <td style="text-align: center;">50000</td> </tr> <tr> <td>SLA: Silver (Services)</td> <td style="text-align: center;">1</td> <td style="text-align: center;">20000</td> </tr> </tbody> </table>	{{TABLEGROUP:OPPLINEITEMS:OPPORTUNITY_LINEITEM_PROD_FAMILY:SHOW}}			Product	Quantity	Total Price	{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUA NTITY}}	{{OPPORTUNITY LINEITEM TOTALPRICE}} {{TABLEEND:OPPLINEITEMS}}	Hardware			Product	Quantity	Total Price	GenWatt Diesel 1000kW (Products)	2	200000	GenWatt Gasoline 2000W (Products)	1	150000	Hardware			Product	Quantity	Total Price	Installation: Industrial - Medium (Services)	1	50000	SLA: Silver (Services)	1	20000
{{TABLEGROUP:OPPLINEITEMS:OPPORTUNITY_LINEITEM_PROD_FAMILY:SHOW}}																																		
Product	Quantity	Total Price																																
{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUA NTITY}}	{{OPPORTUNITY LINEITEM TOTALPRICE}} {{TABLEEND:OPPLINEITEMS}}																																
Hardware																																		
Product	Quantity	Total Price																																
GenWatt Diesel 1000kW (Products)	2	200000																																
GenWatt Gasoline 2000W (Products)	1	150000																																
Hardware																																		
Product	Quantity	Total Price																																
Installation: Industrial - Medium (Services)	1	50000																																
SLA: Silver (Services)	1	20000																																

## Add Footers to Repeating Data Tables in Word

The TableGroupFooter fields allow you to display a row at the bottom of the grouped table using TableGroupStart. This row may be used to reiterate data that's already displayed elsewhere in the table or to display new information. TableGroupFooter requires TableGroupStart and must reference the same dataset.

Syntax	<pre> {{TableGroupFooterStart:DatasetName}} {{TableGroupFooterEnd:DatasetName}} </pre>
Values	DatasetName is the name of the dataset from the Template Builder window.
Notes	<p>After creating a grouped table using TableGroupStart, insert a row at the bottom of the table to accommodate the footer.</p> <p>You may insert one merge field into the TableGroupFooter.</p> <p>The TableGroupFooterStart and TableGroupFooterEnd fields must be placed within the same table row or cell.</p> <p>The dataset referenced in the TableGroupFooterStart field must match:</p> <ul style="list-style-type: none"> <li>• The dataset referenced in the TableGroupFooterEnd field.</li> <li>• The dataset as referenced in the TableGroupStart and the detail region.</li> </ul>

Usage

Case Review: {{ACCOUNT\_NAME}}

---

**{{TABLEGROUPSTART:CASES:TYPE}}**

TYPE: {{Type}}                      Status: {{Status}}                      Opened: {{Opened\_Date \@ MMM-dd-yyyy}}

{{TABLEGROUPEND:CASES:TYPE}}

Subject	Number	Contact	Description
{{TableStart:Cases}} {{Subject}}	{{Case_Number}}	{{Contact_Name}}	{{Description}} {{TableEnd:Cases}}
			{{TableGroupFooterStart:Cases}} {{TableGroupFooterEnd:Cases}}

On merge, the output is formatted as below:

Case Review: {{ACCOUNT\_NAME}}

---

**{{TABLEGROUPSTART:CASES:TYPE}}**

TYPE: {{Type}}                      Status: {{Status}}                      Opened: {{Opened\_Date \@ MMM-dd-yyyy}}

{{TABLEGROUPEND:CASES:TYPE}}

Subject	Number	Contact	Description
{{TableStart:Cases}} {{Subject}}	{{Case_Number}}	{{Contact_Name}}	{{Description}} {{TableEnd:Cases}}
			{{TableGroupFooterStart:Cases}} {{TableGroupFooterEnd:Cases}}

## Conditional TableHide for Detail Data in Word

The Detail Data Conditional TableHide is a merge field comprised of four elements and allows a table to be hidden if the detail data field referenced contains a particular value. The Detail Data Conditional TableHide evaluates the first row of data in the specified dataset. If the field specified contains the value, the table is hidden.

The Conditional TableHide is functional for detail data sets and will not work with master datasets. Use a [Conditional TableHide with an Operator](#) or an [IF statement in Word](#) to conditionally hide a table with a master data set.

We do not advise using more than 25 TableHide fields in a single template.

This feature is not supported when using nested SOQL queries.

<b>Syntax</b>	{{TableHide:Dataset:FieldName:Value}}
---------------	---------------------------------------

<b>Values</b>	<p>DatasetName is the name of the dataset from the Template Builder window.</p> <p>FieldName is the name of the field from the Template Builder window.</p> <p>Value is the value of the field that will force Composer® to hide the table.</p>								
<b>Notes</b>	<p>Any conditional TableHide using detail data will evaluate only the first row of data in the dataset. For best results, sort your report or query to ensure the appropriate record is returned as the first result.</p>								
<b>Usage</b>	<table border="1"> <thead> <tr> <th data-bbox="264 658 858 788">Product{{TableHide:OPPLINEITEMS:PRODUCT_FAMILY:Misc.}}</th> <th data-bbox="865 658 1075 788">Quantity</th> <th data-bbox="1082 658 1439 788">Total Price</th> </tr> </thead> <tbody> <tr> <td data-bbox="264 797 858 996">           {{TABLESTART:OPPLINEITEMS}}            {{OPPORTUNITY_LINEITEM_PRODUCTNAME}}         </td> <td data-bbox="865 797 1075 996">           {{OPPORTUNITY_LINEITEM_QUANTITY}}         </td> <td data-bbox="1082 797 1439 996">           {{OPPORTUNITY_LINEITEM_TOTALPRICE}}            {{TABLEEND:OPPLINEITEMS}}         </td> </tr> </tbody> </table>			Product{{TableHide:OPPLINEITEMS:PRODUCT_FAMILY:Misc.}}	Quantity	Total Price	{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM_PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUANTITY}}	{{OPPORTUNITY_LINEITEM_TOTALPRICE}} {{TABLEEND:OPPLINEITEMS}}
Product{{TableHide:OPPLINEITEMS:PRODUCT_FAMILY:Misc.}}	Quantity	Total Price							
{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM_PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUANTITY}}	{{OPPORTUNITY_LINEITEM_TOTALPRICE}} {{TABLEEND:OPPLINEITEMS}}							

## Special Merge Fields in Word

Special merge fields for Word templates.

- [Special Merge Fields in Word Templates](#)
- [ADJUST \(Dynamic Dates\) in Word](#)
- [AutoSum\(Above\) in Word](#)
- [Checkboxes for Multi-Select Picklist Fields in Word](#)
- [CHECKBOX in Word](#)
- [Composer - HYPERLINK prefix in Word](#)
- [HTML in Word](#)
- [IMAGE Prefix in Word](#)
- [IncludePicture in Word](#)
- [REPLACE in Word](#)
- [RowCount in Word](#)
- [Row Number Field in Word](#)
- [RTL in Word](#)
- [SEQ \(Sequence\) in Word](#)
- [SUM\(ABOVE\) in Word](#)
- [TableHide in Word](#)
- [TableList in Word](#)



- [TEXTBOX in Word](#)
- [WATERMARK in Word](#)

## Special Merge Fields in Word Templates

Conga Composer lets you customize and format template documents with special merge fields.

Merge Field	Notes
Adjust	Dynamically adjusts date fields to a future or past value. For more information, see <a href="#">Format Date Values</a> .
Checkbox	Include a checkbox that is conditionally selected (checked or ticked) or cleared (not checked). For more information, see <a href="#">CHECKBOX</a> .
HTML	Render text that includes HTML formatting codes as Word® rich text. For more information, see <a href="#">HTML</a> .
Hyperlink	Generates clickable Word hyperlinks. For more information, see <a href="#">HYPERLINK</a> .
Image	Dynamically includes graphic files into your merged output. For more information, see <a href="#">IMAGE</a> .
MERGESEQ	Counts the number of data records that were successfully merged. For more information see <a href="#">MERGESEQ</a> .
Replace	Replaces text in a merge field with another value. For more information, see <a href="#">REPLACE</a> .
RowCount	Provides a count of the number of results in a detail dataset. For more information, see <a href="#">RowCount</a> .
RTL	Specifies the use of a right-to-left language. For more information, see <a href="#">RTL</a> .
SEQ	Establishes table row numbers. For more information, see <a href="#">SEQ</a> .

Merge Field	Notes
SUM(ABOVE)	Computes the SUM of currency values. For more information, see <a href="#">SUM(ABOVE)</a> .
System	Inserts either the current date or current date and time at your location. For more information, see <a href="#">System Fields</a> .
TableGroup	Summarizes a table-based Detail Region on a field. For more information, see <a href="#">Repeating Data</a> .
TableGroupStart	Summarizes a table-based Detail Region on multiple fields. For more information, see <a href="#">Repeating Data</a> .
TableHide	Dynamically hides a table-based Detail Region if it has no data. For more information, see <a href="#">Repeating Data</a> .
TableList	Displays a column of detail-data as a list. For more information, see <a href="#">Repeating Data</a> .
TextBox	Designates fields that should remain editable after Conga® generates a Protected Word Document. For more information, see <a href="#">TEXTBOX</a> .
Watermark	Inserts merged data as a watermark on the page. For more information, see <a href="#">WATERMARK</a> .

## ADJUST (Dynamic Dates) in Word

You can adjust a date merge field by including the ADJUST: prefix. You can adjust a date merge field by including the ADJUST: prefix. For example, you could display the date 30-days past an Opportunity Close Date.

The ADJUST: prefix works with any Salesforce date field or the Conga TODAY or NOW fields from the System dataset, which is provided by default in the Template Builder.

ADJUST	
Syntax	<code>{{ADJUST:MyDateField:Year:Month:Day:Hour:Minute:Second}}</code>

ADJUST	
Values	<p>MyDateField is the name of your date field (e.g. Close_Date), TODAY or NOW</p> <p>Year is the number of years you want to adjust the date</p> <p>Month is the number of months</p> <p>Day is the number of days</p> <p>Hour is the number of hours</p> <p>Minute is the number of minutes</p> <p>Second is the number of seconds</p>
Notes	<p>You must use all of the data elements (years, months, days, hours, minutes, seconds). For any element that you do not want to adjust, insert a zero (0).</p> <p>Dates can be adjusted for the future or past with either a positive or negative integer: future dates are positive integers, past dates negative (e.g. 6 or -6).</p> <p>You may also use a picture switch to edit how the date field will appear, e.g. <code>{{ADJUST:Close_Date:0:0:30:0:0:0 \@ "MMMM d, yyyy"}}</code>. However, picture switches only work with Master fields.</p>
Examples	<p><code>{{ADJUST:Close_Date:0:0:30:0:0:0}}</code> The value of the Close_Date field is adjusted to 30 days in the future.</p> <p><code>{{ADJUST:Created_Date:0:-6:0:0:0:0}}</code> The value of the Created_Date field is adjusted to 6 months ago</p>

## AutoSum(Above) in Word

The "AUTOSUM(ABOVE)" merge field is used as a substitute for a "{=SUM(ABOVE)}" field in the following situations.

In performing merges that include detail regions, you may wish to include the total of a numeric column. This can be done using the `{{AUTOSUM(ABOVE)}}` merge field, which is ideal in the following situations:

- The cells contain international currency values
- The cells contain blank cells
- The cells contain negative values formatted with parentheses

This is particularly useful with currency values that display a multi-letter currency symbol, e.g., Danish Kroner. "50,0 kr." Before AUTOSUM, the native SUM(ABOVE) would find the total, but not display the currency symbol. AUTOSUM finds the SUM and retains the currency symbol and proper thousand and decimal delimiters. Note the number of columns in the row with the AUTOSUM field must match the number of columns on the rows above it.

The line items above this field entry must be formatted with the "\# Currency", "\# Currency0" or "\# Number" merge field formatting code. You may not use any merged cells on the row that contains the "AUTOSUM(ABOVE)" field. Also, AUTOSUM results are static and cannot be updated after the output document has been created.

✔ AUTOSUM fields are also useful for finding the sum of negative values (that are formatted with parentheses), or across blank cells. Word (on its own) won't add these kinds of values.

⚠ When using the AUTOSUM(ABOVE) or SUM(ABOVE) functions to total an international currency field that dynamically formats currency with the [currency culture parameter](#) within an IF statement, the merged total amount displays in the running user's currency (determined by their locale), as opposed to the value determined by the currency culture parameter. A workaround for this scenario is building a Salesforce formula field that converts the international currency field's value into a number value. Then use an aggregated SUM query for that formula field and use the [\# Currency picture switch](#) in the merge field added to the template.

## Checkboxes for Multi-Select Picklist Fields in Word

You can create a checkbox field for each value of a multi-value picklist field. The checkbox is selected if the field contains the value included in the syntax.

<b>Syntax</b>	<code>{{CHECKBOX:FieldName:Value}}</code>
<b>Values</b>	<p>FieldName is the merge field to evaluate.</p> <p>Value determines whether to select or clear the checkbox.</p>


<b>Usage</b>	<pre> {{CHECKBOX:Flavors:Chocolate}} Chocolate {{CHECKBOX:Flavors:Vanilla}} Vanilla {{CHECKBOX:Flavors:Banana}} Banana On merge: <input checked="" type="checkbox"/> Chocolate <input type="checkbox"/> Vanilla <input type="checkbox"/> Banana </pre>
--------------	--

For field values that contain Dashes, you must enclose the entire Merge Field in double-quotes. For example, Value is Strawberry-Banana. Your merge field should look like this:

```

{{"CHECKBOX:Flavors:Strawberry-Banana"}}

```

 If the field contains spaces, you do not need to enclose the field in quotation marks.

## CHECKBOX in Word

You can enhance the appearance of your merged data by employing the CHECKBOX: merge field prefix. This setting works with many kinds of Salesforce fields: Checkbox, Number, Text, and Picklist.

Apply the CHECKBOX: Option to a Merge Field When you apply the CHECKBOX: option to a merge field, Conga Composer will create a square in place of the merge field. The square will either be selected (checked) or not (cleared), depending on the value of the merge field.

<p>Checkboxes are not selected if the field value contains:</p> <ul style="list-style-type: none"> <li>• 0 (zero)</li> <li>• False</li> <li>• No</li> <li>• No value</li> </ul>	<p>Checkboxes are selected if the field value contains:</p> <ul style="list-style-type: none"> <li>• 1</li> <li>• True</li> <li>• Yes</li> <li>• Any non-false value</li> </ul>
---	---

## CheckBox

<b>Syntax</b>	<pre> {{CHECKBOX:FieldName}} </pre>
<b>Usage</b>	<pre> {{CHECKBOX:Has_Cell_Phone}} </pre>

<b>Values</b>	CHECKBOX: is the checkbox field prefix.  FieldName is the name of your Checkbox, Number, Text, or Picklist field, as it appears in the Template Builder.
---------------	--

## Checkbox Fields in Protected Documents

You may also use Checkbox fields in conjunction with the “Word Protected Document” Conga Composer parameter WPD. With this mode enabled, Composer will create a “protected” (non-editable) Word document—except for Checkbox and TextBox fields, which remain editable in the output file.

For more information, see [TEXTBOX](#).

## Composer - HYPERLINK prefix in Word

Conga Composer can render Salesforce URL fields as clickable links. The field within Salesforce must contain an absolute (complete) URL address. You can also specify the link text to display in the merged file instead of the URL value.

HYPERLINK	
Syntax	<code>{{HYPERLINK:FieldName:DisplayText}}</code>
Values	FieldName is the name of the field from the Template Builder window.  DisplayText is an optional suffix to provide friendly display text rather than the entire URL.
Usage	<code>{{HYPERLINK:Company_Website:Visit our home page.}}</code> After merging, appears as:  Visit our home page.


## HTML in Word

Conga Composer can render Salesforce fields that contain HTML markup using the HTML prefix. This is particularly useful for long text fields that contain the HTML line break tag (<br />) and for Salesforce rich text fields. You can also use these prefix formula fields that build HTML strings.

<b>Syntax</b>	<code>{{HTML:FieldName}}</code>
<b>Values</b>	<code>FieldName</code> is the name of the field from the Template Builder window.
<b>Notes</b>	<p>Images stored directly in rich text fields are not supported and do not display in the output file.</p> <p>However, if you retrieve the Salesforce download URL and add that to your rich text field, the Composer can format the image in the generated document. Your URL might look something like this: <i>https://c.na88.content.force.com/sfc...sion/download/0681Y000004rEToQAM?asPdf=false&amp;operationContext=CHATTER.</i></p>

## IMAGE Prefix in Word

You can merge images into a Word Template by using the IMAGE prefix and a public URL that is not subject to anti-scraping protection. You can also use this prefix to specify the height or width of the merged image.

<b>Syntax</b>	<p><code>{{IMAGE:FieldName}}</code></p> <p>“FieldName” is the name of the field from Template Builder that contains the image URL.</p>
<b>Usage</b>	<ul style="list-style-type: none"> <li>Automatically resize merged images by specifying a height or width value in pixels. The aspect ratio of the image is maintained <ul style="list-style-type: none"> <li>To specify the height, include a colon after the merge field name, followed by H=[height in pixels]</li> <li>To specify the width, include a colon after the merge field name, followed by W=[width in pixels]</li> </ul> <div style="border: 1px solid #ffc107; padding: 5px; margin: 10px 0;"> <p> Specify either height or width, but not both.</p> </div> </li> <li>You may use public images by storing the URL in a field in Salesforce</li> <li>You may use Salesforce Files or Documents by copying the download URL and storing it in a field in Salesforce (see URLs section below)</li> <li>The file type must be JPG or PNG</li> </ul>


<p><b>Examples</b></p>	<p>For a field called Photo URL that contains a public URL:</p> <div style="border: 1px solid #ccc; padding: 2px; margin: 5px 0;"> <p><b>Photo URL</b> <a href="https://na3.salesforce.com/servlet/servlet.FileDownload?file=00P50000001ySk7">https://na3.salesforce.com/servlet/servlet.FileDownload?file=00P50000001ySk7</a></p> </div> <p>The Word Template merge field syntax will be:</p> <pre>{{IMAGE:Photo_URL}}</pre> <p>To specify an image with a width of 100 pixels:</p> <pre>{{IMAGE:OPPORTUNITY_IMAGE_FIELD:W=100}}</pre> <p>To specify an image with a height of 100 pixels:</p> <pre>{{IMAGE:OPPORTUNITY_IMAGE_FIELD:H=100}}</pre>
<p><b>URLs</b></p>	<p>To use the public URL for Salesforce Files and Documents:</p> <ol style="list-style-type: none"> <li>1. Create a Text or URL field on your master object to hold the image URL.</li> <li>2. Upload an image to Salesforce Documents or as a File on your master record.</li> <li>3. Open the image and click <b>Public Link</b>.</li> <li>4. Follow all prompts to generate the public URL.</li> <li>5. Open the URL into a new window.</li> <li>6. Right-click on the image and copy the image address.</li> <li>7. Paste that link into the field you created in step 1.</li> <li>8. Merge the field holding the URL into your Word template and use the IMAGE prefix as shown in the examples above.</li> </ol>

## IncludePicture in Word

This is the recommended method of merging images to Word® documents when using a publicly accessible image.

<p><b>Syntax</b></p>	<pre>{{IncludePicture:URLFieldName}}</pre>
<p><b>Value</b></p>	<p>URLFieldName is the field's name from the Template Builder window that contains the image URL.</p>
<p><b>Notes</b></p>	<p>The file type must be JPG or PNG.</p> <p>The file must be publicly accessible (not stored in Salesforce®) via a URL and not be subject to anti-scraping protection.</p>



<b>Usage</b>	<pre> {{IncludePicture:Product_Logo_URL}} </pre> <p>On merge, it becomes:</p> 
--------------	---

## REPLACE in Word

The REPLACE merge field prefix lets you replace one value for another. For example, you can replace new line characters in an address field with commas. This is also useful in displaying the values from a multi-select picklist field in a more attractive manner than the default, semi-colon delimiter.

<b>REPLACE</b>	
Syntax	<pre> {{REPLACE:FieldName:{FromValue}:{ToValue} }} </pre>
Values	<p>FieldName is the name of the field from the Template Builder window.</p> <p>{FromValue} is the aspect of the FieldName that you want to alter.</p> <p>{ToValue} is what you want to change the FromValue to instead.</p>

REPLACE	
Notes	<p>A space is needed between the field and the end braces “} }”.</p> <p>Supported FromValue and ToValue entries:</p> <ul style="list-style-type: none"><li>• {COLON}</li><li>• {COMMA}</li><li>• {NEWLINE}</li><li>• {PERIOD}</li><li>• {SEMICOLON}</li><li>• {SPACE}</li><li>• Literal Text</li></ul> <p>The FromValue and ToValue entries are literal text and the curly braces surrounding them need to be included. You can use Traditional Merge fields.</p> <p>To end the ToValue with a space, use {SPACE}.</p>

REPLACE	
Usage	<pre> {{REPLACE:ACCOUNT_FULLADDRESS:{NEWLINE}:{COMMA} {SPACE} }} </pre> <p>Replaces NEW LINE characters in the Full Address field with ", " (a COMMA and a SPACE).</p> <p>2655 Midway Blvd. Broomfield, CO 80023</p> <p>becomes</p> <p>2655 Midway Blvd. Broomfield, CO 80023</p> <pre> {{REPLACE:ACCOUNT_FAVORITE_FLAVORS:{SEMICOLON}:{COMMA} {SPACE} }} </pre> <p>Replaces ";" (SEMICOLONS) in the multi-value picklist field with ", " (a COMMA and a SPACE).</p> <p>Chocolate;Vanilla;Strawberry</p> <p>becomes</p> <p>Chocolate, Vanilla, Strawberry</p> <pre> {{REPLACE:ACCOUNT_FULLADDRESS:CO:CoLorado}} </pre> <p>Replaces literal text with a new literal text value.</p> <p>2655 Midway Blvd. Broomfield, CO 80023</p> <p>becomes</p> <p>2655 Midway Blvd. Broomfield, Colorado 80023</p>

## RowCount in Word

You can display the number of rows in a set of detail data (data retrieved using a report or query) by creating a special merged field, RowCount:DatasetName.

<b>Syntax</b>	<code>{{RowCount:DatasetName}}</code>
<b>Values</b>	DatasetName is the name of the dataset from the Template Builder window.

<b>Usage</b>	<p><code>{{RowCount:Contacts}}</code></p> <p>If the Contacts dataset has 10 rows of data, the number 10 appears in place of this field.</p>
--------------	---

## Row Number Field in Word

The ROW\_NUMBER field numbers each line item in a detail dataset and is included by default in every report or query dataset.

Insert this field into any detail region in the template to number the results in the merged output file.

**Syntax:** `{{Row Number}}`

**Notes:** This merge field must be placed in the far left column of a data table.

**Usage Example:**

Item Number	Product Name	Quantity	Total
<p><code>{{TableStart:OppLineItems}}</code></p> <p><code>{{ROW_NUMBER}}</code></p>	<p><code>{{PRODUCT_NAME}}</code></p>	<p><code>{{OPPORTUNITY_LINEITEM_QUANTITY}}</code></p>	<p><code>{{OP}}</code></p> <p><code>\# Cu</code></p> <p><code>{{Tab</code></p>

The merged output file looks like this:

Item Number	Product Name	Quantity	Total Price
1	GenWatt Gasoline 300kW (Products)	1	\$35,000.00
2	GenWatt Gasoline 750kW (Products)	1	\$75,000.00
3	Installation: Industrial - Low (Installation)	1	\$20,000.00
4	SLA: Gold (Services)	1	\$30,000.00

## RTL in Word

When using a language that is read from right-to-left (e.g. Arabic, Hebrew), you can identify which merge fields contain right-to-left text using the RTL: prefix. This allows you to identify the right-to-left text on a field-by-field basis, which is especially useful if your template will utilize both right-to-left and left-to-right text – some fields can use the RTL prefix and others not so the merged output file has a mixture of both sorts of text.

<b>Syntax</b>	{{RTL:Name}}
<b>Notes</b>	<p>Do not use the &amp;RTLFont=1 parameter in conjunction with the RTL: merge field prefix. The parameter specifies that the entire template should be merged in right-to-left text and will cause errors when used along with the RTL: prefix.</p> <p>Word templates that include right-to-left languages are better supported by Conga Composer's PDF rendering engine. We recommend rendering your Word template as a PDF for optimum results.</p>
<b>Usage</b>	Add the prefix RTL: at the beginning of the field name.

## SEQ (Sequence) in Word

To establish table row numbers, create a Sequence ("SEQ") field and place it inside a detail region (between the TableStart and TableEnd fields). The field must include a label to identify which dataset you want to number. Here's how:

Create a merge field for your sequence field, using the following syntax:

```
{{@SEQ DatasetName}}
```

@SEQ is the sequence field prefix. The at symbol @ is required and differentiates Conga Composer's text-based Sequence field from Word's built-in Sequence field. A space is required between @SEQ and DatasetName.

DatasetName: DatasetName is the name of the dataset (example:\ Contacts) as it appears in the Template Builder whose rows (a.k.a. results) you wish to number.

**⚠ The Sequence field must refer to the same dataset as the detail region.**

Place the sequence field inside of a detail region (note: the sequence field below is in blue font simply for illustrative purposes; this is not required). Below, we're numbering the table rows for our list of Contacts:

<i>[[ACCOUNT_NAME]]</i>		
<b>First Name</b>	<b>Last Name</b>	<b>Title</b>
<i>[[TableStart.Contacts]]</i> <b>[[@SEQ Contacts]]</b> <i>[[First_Name]]</i>	<i>[[Last_Name]]</i>	<i>[[Title]]</i> <i>[[TableEnd.Contacts]]</i>

Since the field is included in the detail region, Conga Composer will display it for each row in the table and the table rows are numbered sequentially:

*Edge Communications*

First Name	Last Name	Title
1 Rose	Gonzalez	SVP, Procurement
2 Sean	Forbes	CFO
3 Pablo	Estrella	Director of Product Development

## SUM(ABOVE) in Word

In performing merges that include detail regions, you may wish to include the total of a numeric column. This can be accomplished in the Word® template.

✔ See [AutoSum\(Above\)](#) if you are working with international currencies.

**Name:** SUM(ABOVE)

**Usage:** See the tables below for example usage, syntax, and output.

**Note:** The number of columns in the row with the SUM(ABOVE) field must match the number of columns in the rows above it. You can turn off table borders to “hide” cells if needed.

**Usage and Syntax in a Table:**

Product	Quantity	Total Price
{{TABLESTART:OPPLINEITEMS}} {{OPPORTUNITY_LINEITEM_PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUANTITY}}	{{OPPORTUNITY_ {{TABLEEND:OPP
		{{=SUM(ABOVE)}}

**Output:**

Product	Quantity	Total Price
GenWatt Diesel 1000kW (Products)	2	200000
GenWatt Gasoline 2000W (Products)	1	150000
		350000

⚠ When using the AUTOSUM(ABOVE) or SUM(ABOVE) functions to total an international currency field that dynamically formats currency with the [currency culture parameter](#) within an IF statement, the merged total amount displays in the running user's currency (determined by their locale), as opposed to the value determined by the currency culture parameter. A workaround for this scenario is building a Salesforce formula field that converts the international currency field's value into a number value. Then use an aggregated SUM query for that formula field and use the [\# Currency picture switch](#) in the merge field added to the template.

## TableHide in Word

You can hide a table that has no data rows from the corresponding report or query dataset. Include `{{TableHide:DatasetName}}` anywhere in the table.

⚠ We do not advise using more than 25 TableHide fields in a single template.

TableHide requires repeating data table using TableStart and TableEnd.

<b>Syntax</b>	<code>{{TableHide:DatasetName}}</code>
<b>Values</b>	DatasetName is the name of the dataset from the Template Builder window.
<b>Notes</b>	<p>The dataset value must be identical to the TableStart and TableEnd fields.</p> <p>You can place TableHide anywhere within the table.</p> <p>TableHide works for child data sets when using nested detail regions; however, nested detail regions are only supported when using queries to retrieve detail data. Reports do not support nested detail regions.</p>

Example:

Product <code>{{TABLEHIDE:OPPLINEITEMS}}</code>	Quantity	Total Price
<code>{{TABLESTART:OPPLINEITEMS}}</code> <code>{{OPPORTUNITY_LINEITEM_PRODUCTNAME}}</code>	<code>{{OPPORTUNITY_LINEITEM_QUANTITY}}</code>	<code>{{OPPORTUNITY_</code> <code>{{TABLEEND:OPP</code>

## TableList in Word

You can display a column from the report or query as a list of values, separated by a punctuation mark you specify. TableList fields may be included within paragraphs.

### TableList

<b>Syntax</b>	{{TableList:DatasetName:FieldName:Punctuation}}
<b>Values</b>	<p>DatasetName is the name of the dataset from the Template Builder window.</p> <p>FieldName is the name of the field from the Template Builder window.</p> <p>Punctuation is the punctuation mark that will separate values.</p>
<b>Notes</b>	<p>You may not use a quote (") or colon (:) as the punctuation value.</p> <p>Duplicate values are removed from the list automatically.</p> <p>TableList does not work in conjunction with TableGroupStart/ TableGroupEnd.</p> <p>TableList does not work in conjunction with Nested SOQL Query Datasets.</p>
<b>Usage</b>	<p>For a list separated by commas, place a comma before the double braces:</p> <p>Attendees are {{TableList:Contacts:First_Name;}}</p> <p>On merge, the field values are separated by a comma.</p> <p>Attendees are Rose, Sean, Pablo.</p>

## TEXTBOX in Word

Conga Composer can create a protected Word document, where the document is locked and cannot be edited after merging. However, you can use the TEXTBOX merge field to specify text that can be edited.

This feature only works with Conga Composer, not Conga Mail Merge.

### TEXTBOX



Syntax	<code>{{TEXTBOX:FieldName}}</code>
Notes	<p>Conga Composer produces an editable Word Text Input box in the output file. Any merge fields that do not include this prefix will be populated with data, but not editable in the output file.</p> <p>To use this feature, you must have a Conga Composer button that contains the WPD parameter value: <code>&amp;WPD=1</code>.</p> <p>If the TEXTBOX merge field is inside of a Word text box the field is not editable.</p>
Usage	<p>Include the <code>&amp;WPD=1</code> parameter in your Composer button URL so that the document that is merged from that button cannot be edited:</p> <pre>https://composer.congamerge.com ?SessionId={!API.Session_ID} &amp;ServerUrl={!\$Api.Partner_Server_URL_290} &amp;Id={!Account.Id} &amp;WPD=1</pre> <p>In the Word template, this TextBox merge field will be available to edit after the document has been merged:</p> <pre>{{TEXTBOX:Note_To_Customer}}</pre>

## WATERMARK in Word

You can apply a watermark to your Word template by using the WATERMARK: field prefix option. When you include the Watermark option to a merge field, Conga Composer will create a document watermark across the background of an output document.

<b>Syntax</b>	<code>{{WATERMARK:FieldName:Font:Width:Height:Rotation:ColorCode}}</code>
---------------	---

<p><b>Values</b></p>	<p><b>WATERMARK:</b> is the watermark field prefix.</p> <p>FieldName is the name of the field (e.g. Account_Type) that you want to display as a watermark in the merged output file, as it appears in the Template Builder.</p> <p>Font is the name of the font in which the watermark will be displayed (optional).</p> <p>Width is the width of the watermark, in pixels (optional).</p> <p>Height is the height of the watermark, in pixels (optional).</p> <p>Rotation is the angle of rotation, in degrees; the supported range must fall within -360 to 360.</p> <p>ColorCode is the HTML hexadecimal color code in which the watermark will be displayed.</p>
<p><b>Notes</b></p>	<p>Supported fonts are Arial, Times, Courier, Verdana, and Calibri.</p> <p>You may only use one Watermark field per template.</p> <p>WATERMARK fields are particularly useful when used in conjunction with the DVO parameter, which inserts a value into the Master data set without requiring a custom field in Salesforce.</p>

## Usage

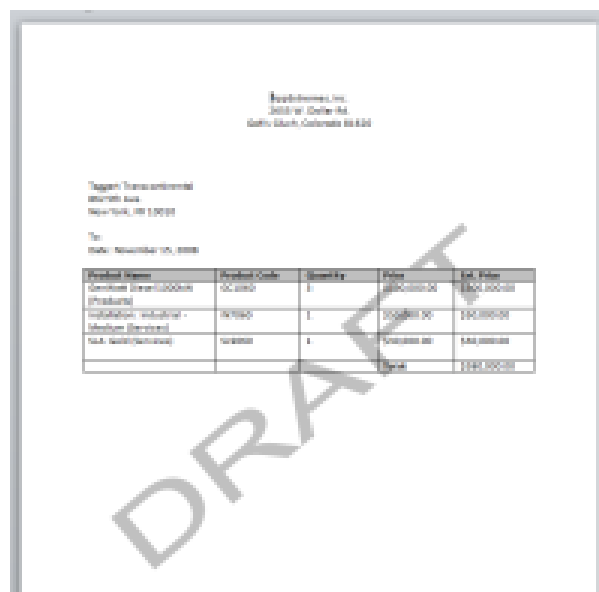
To conditionally watermark the document as draft if the value of the Opportunity Stage is Negotiation/Review, include the DV0 parameter with an if statement that sets the watermark text when conditions are met:

```
https://composer.congamerge.com
?SessionId={!API.Session_ID}
&ServerUrl={!$Api.Partner_Server_URL_290}
&Id={!Opportunity.Id}
&DV0={!IF( ISPICKVAL(Opportunity.StageName, "Negotiation/Review"), "DRAFT",
"" )}
```

In the template, add a Watermark field that references the DV0 field that now exists in your Master data set:

```
{!{WATERMARK:DV0:Arial:700:100:-45:#808080}}
```

This places the content of the DV0 field as a watermark overlaying the text and then styles it in Arial text, 700 pixels wide by 100 pixels high, rotated by 45 degrees, and in the hexadecimal color #808080 (light gray).



## Advanced Word Templates

This section contains information on advanced template-building techniques and tricks.

- [Add Bar-codes to a Merge Templates](#)
- [Add Page Breaks to Repeating Data Tables in Word](#)
- [Automatically Updating Word Fields](#)
- [Conditional TableHide Examples in Word](#)
- [Creating Word Calculation Fields](#)


- [Custom TrueType Fonts](#)
- [Dynamic Page Numbering in Word](#)
- [Hiding Tables Based on Conditions](#)
- [How to Create IF Fields in a Microsoft Word Template](#)
- [How to Create Nested Tables in Word Templates](#)
- [How to Embed Non-Standard Fonts in Conga Word Templates](#)
- [How to Use Bookmarks and References in Word Templates](#)
- [How to Compare Two Values in Word Templates using IF Statements](#)
- [Inserting IF Statements into a Word Template with Word for Mac](#)
- [Linetrिम](#)
- [Master Data Conditional TableHide in Word](#)
- [Merge Google Maps into a Word Template](#)
- [Merging Images from Salesforce Files](#)
- [Merge Images with Salesforce ID in Word](#)
- [Nested SOQL Queries](#)
- [Tablelinetrिम](#)
- [Conditional TableHide Using an Operator in Word](#)
- [Conditional TableHide Comparing Two Field Values in Word](#)
- [Arranging Images Horizontally in Word](#)

## Add Bar-codes to a Merge Templates

You can add a bar-code field to your template by changing the font of a field on your template to a bar-code font. You must have a True-Type bar-code font installed on the machine building the template. This font may also need to be installed on your server for rendering.

To create a bar-code field:

1. Create a template. For more information, see [Create a Template with Template Builder](#).
2. Select the field on the template, and then set the font to the bar-code font.
3. Test your template by merging your template in both 'Same as Template' and 'Adobe PDF'.

 **If a bar-code does not render, please contact support.**

## Add Page Breaks to Repeating Data Tables in Word

In addition to the TableGroup and TableGroupStart features, you can request that Composer insert a page break between groups (except for the first group).

In addition to grouping data in Word templates using the TableGroup or TableGroupStart merge fields, you can also insert a page break between grouped tables (except for the first group) so that each table appears on its own page.

**⚠ Do not mix and match the Tablegroupstart and End method with the Tablegroup method. You must pick one method for your template.**

For example, if you want a table grouped on Product Families to appear on different pages in the output document (Software on one page, Hardware on another, etc.), you would put the following special merge fields in the first cell of the table in your template (see screenshot below):

```

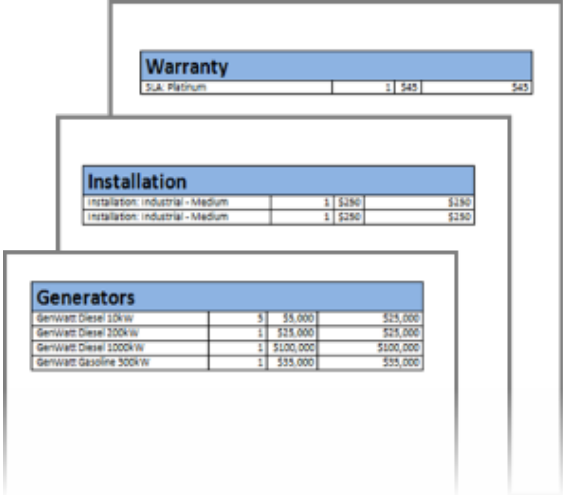
{{TABLEGROUPSTART:OPPLINEITEMS:PRODUCT_FAMILY:PAGEBREAK}}
{{TABLEGROUPEND:OPPLINEITEMS:PRODUCT_FAMILY}}
    
```

{{TABLEGROUPSTART:OPPLINEITEMS:PRODUCT_FAMILY:PAGEBREAK}}                 {{TABLEGROUPEND:OPPLINEITEMS:PRODUCT_FAMILY}}			
PRODUCT	SALES PRICE	QTY	TOTAL PRICE
{{TABLESTART:OPPLINEITEMS}}                     {{PRODUCT_NAME}}	{{OPPORTUNITY_LINEITEM_SALE PRICE \# CURRENCY}}	{{OPPORTUNITY_LINEITEM_QUANTITY ANTITY \# \# #0}}	{{OPPORTUNITY_LINEITEM_TOTALPRICE Currency}}                     {{TABLEEND:OPPLINEITEMS}}
	Subtotal	{{=SUM(Above)}}	{{=SUM(Above)}}
<b>INVOICE TOTAL DUE:</b>		{{OPPORTUNITY_AMOUNT \# CURRENCY}}	

In this example, the name of the dataset is OpplLineItems and the field name is Product\_Family. If your dataset and field names are different, you will need to reference them in the special merge field accordingly.

The table below outlines the details of this feature.

Syntax	{{TableGroupStart:DatasetName:FieldName:PageBreak}} {{TableGroupEnd:DatasetName:FieldName}}
--------	--

<p>Values</p>	<p>DatasetName corresponds to the name of the dataset from the Template Builder window (this would be the alias name for report or query data).</p> <p>FieldName corresponds to the field name displayed in the Template Builder window. It is the field on which you want to group the data (i.e. OPPORTUNITY_STAGE).</p> <p>PageBreak indicates that a page break should precede the table group.</p>
<p>Notes</p>	<p>The FieldName in the {{TableGroupStart}} and {{TableGroupEnd}} fields must not contain hyphens or colons. If the FieldName does contain these characters, use the &amp;SHP parameter to strip punctuation from the merge field. Also, the field names for {{TableStart}} and {{TableEnd}} cannot be abbreviated and must be placed within the same row of the detail table.</p>
<p>Usage</p>	<p>When using the PageBreak, the first group remains where it was originally positioned. Each subsequent group will start on a new page.</p> 

## Automatically Updating Word Fields

Update Word fields: formula fields, =SUM fields, IF fields, etc., regardless of whether the output file is Word or PDF. Macros to update fields are not necessary.

Conga Composer updates all kinds of Word® fields: formula fields, =SUM fields, IF fields, etc., regardless of whether the output file is Word or PDF. Macros to update fields are not necessary (and are not supported by Conga Composer anyway).

### Can Macros Run Before Converting a Document to PDF?

In a word, no, that's not possible. The system is specifically designed to ignore macros built-in to your templates for the safety and security of all Conga Composer users, and therefore the macros cannot execute before converting the document to PDF. Macros stored in your templates will execute when the output document is opened in Word by an end-user.

### Compute the Sum for a Table Column

See [SUM\(ABOVE\) Fields](#) for details on how to calculate the sum for a table column.

## Conditional TableHide Examples in Word

Examples for using conditional TableHide in Word.

This feature is not supported when using nested SOQL queries. This feature applies to Word templates only. Email, Excel, PDF and PowerPoint Templates are not supported.

### Master Dataset

Using the Opportunity as my Master object, I want to provide instructions to my high value customers describing how to access their online premium member profile. However, I do not want these instructions to appear for customers whose orders are less than \$100,000. I'll use the Opportunity Amount field in my Conditional TableHide and use an Operator to indicate the value of \$100,000. (The conditional TableHide field is shown in red font for visibility.)

Your order of {{OPPORTUNITY\_AMOUNT \# Currency}} qualifies you for access to an online premium member profile.

1. To access this premium member profile, please visit [www.MyPerfectGetaway.com](http://www.MyPerfectGetaway.com).
2. Enter the email address you used to complete your order.
3. Enter the temporary password: [DreamVacationNow](#)
4. Click on the My Profile tab in the upper right corner to complete your profile.

{{TableHide:Master:OPPORTUNITY\_AMOUNT:<100000}}

Because the Olenberger Family Membership Opportunity has an Amount of \$160,000 (greater than \$100,000), the table is not hidden:

Your order of \$160,000.00 qualifies you for access to an online premium member profile.

1. To access this premium member profile, please visit [www.MyPerfectGetaway.com](http://www.MyPerfectGetaway.com).
2. Enter the email address you used to complete your order.
3. Enter the temporary password: [DreamVacationNow](#)
4. Click on the My Profile tab in the upper right corner to complete your profile.

### Detail Dataset

I have a table in my proposal template where I do not want to display the widget products my company sells that typically have quantities of hundreds of thousands. Therefore, I will

hide my product table if the first product listed has a quantity of more than 2. Using the OppLineItems detail dataset, I build a table that includes the Conditional TableHide of quantity greater than 2.

Product Name	Quantity	Total Price
<code>{{TableStart OppLineItems}}{{OPPORTUNITY_LINEITEM_PRODUCTNAME}}</code>	<code>{{OPPORTUNITY_LINEITEM_QUANTITY}}</code>	<code>{{OPPORTUNITY_LINEITEM_TOTALPRICE# Currency}}{{TableEnd OppLineItems}}</code>
<code>{{TableHide:OppLineItems:OPPORTUNITY_LINEITEM_QUANTITY:&gt;:2}}</code>		

Because all of my products have quantities of more than two, the table is hidden.

## Additional Examples

```
{{TableHide:Pledges:Amount_Pledged:<:200}}
```

```
{{TableHide:Contracts:Start_Date:>:5/5/2014}}
```

`{{TableHide:Flavors:Favorite_Flavor:<>:Grape}}` this can also be expressed as `{{TableHide:Flavors:Favorite_Flavor:!=:Grape}}`

```
{{TableHide:Master:CONTACT_DISPOSITION:::Satisfied}}
```

## Creating Word Calculation Fields

Leverage Microsoft Word's calculation fields for simple calculations in Word templates.

This method is not supported by Conga. You can create calculated fields in Salesforce. Conga can easily map those values to the Templates. Do not use dynamic culture codes.

1. You start by inserting a traditional word field by going to **Insert** → **Quick Parts** → **Formula**, then click the **Formula** button on the pop-up.
2. Then put numeric fields, operators and a placeholder for any merge fields desired into your Formula: Then right click to toggle open your formula: { =("Mergefield1" \* "Mergefield2") / 100 }
3. Now we can then replace the Mergefield1 text with a word traditional merge field using insert / quick parts / field then mergefield. Then you must nest your merge fields inside of your formula: { =(«Amount 1» \* «Quantity 1») / 100 }
4. When you merge your document, Conga maps the values to the merge fields and Microsoft Word executes the calculations.

Here are some additional details on Word Calculation fields and operators: <https://support.office.com/en-us/art...c-ae3b22b7eb5e>



## Custom TrueType Fonts

The Conga servers automatically support a wide variety of TrueType fonts for Microsoft Office products. However, if your template uses one of the fonts that is not installed with Windows®, your template may not appear as you expected.

In that case, you can submit an embeddable TTF or OTF font file to [Conga Support](#) for deployment on our servers.

Any TTF font file submitted for deployment must be Editable or Installable—it cannot be Restricted. The font must be publicly accessible without license or cost. Font filenames must be identical to the name of the font specified in the template, including letter case and spaces. For example, if the font name specified in the template is Barcode1, the TTF or OTF filename must also be Barcode1. Variations such as barcode 1, Barcode-1, or barcode1 will fail. In that case, the default font for the template will be used.

Fonts submitted for deployment on our servers are also accessible by Microsoft Word, Excel, and PowerPoint templates.

Each customer may submit TTF files not to exceed a cumulative total file size of 10 Mb.

When merging a Word doc to a PDF output and using custom .OTF font files (that are both installed locally and on our servers), the output generated by Composer will populate fine, but if you try to print the document on a Mac, the text with the .OTF font file applied will print out blank. If a customer is printing a PDF using custom fonts generated by Composer on a Mac, make sure they are using custom .TTF fonts.

## Dynamic Page Numbering in Word

If you want to combine multiple Word templates in your solution into a single Word document and retain continuous page numbering, do the following:

- Ensure each template has a page number in the footer
- Use the &AWD parameter to combine the Word documents
- If you want the output converted to PDF, use the [&DefaultPDF](#) parameter

## Hiding Tables Based on Conditions

List of ways to conditionally hide a table based on a field value.

There are a few variations on the regular TableHide, each provides a way to conditionally hide a table based on a field value.

These include:

- [Master Data Conditional TableHide](#)
- [Conditional TableHide for Detail Data](#)
- [Conditional TableHide using an Operator](#)
- [Conditional TableHide Examples](#)

## How to Create IF Fields in a Microsoft Word Template


If fields (also called “if statements”) allow you to compare two values and display document content based on the result of the comparison.

### If Fields / If Statements / Comparison Statements

If fields (also called “if statements”) allow you to compare two values and display document content based on the result of the comparison. When used in a Word template, IF fields are especially useful for comparing the values of Composer merge fields to display an appropriate result.

For example:

- If an Account’s billing country is not in the United States, display information about international shipping terms. Otherwise, show domestic shipping terms.
- If a Contact’s mailing address is blank, show the billing address from his related Account. If it’s not blank, show the Contact’s mailing address.
- If an Opportunity’s amount is greater than \$10,000, show information about a complimentary service package. If not, show nothing.

 Conga does not support nested If statements where the else value is another If statement. Conga recommends building your logic in Salesforce.

- Watch this video on [How to Add IF Statements to Templates](#).

## Create IF Statements in Word Template

You must use traditional merge fields for IF statements. The guidance below uses traditional (not text-based) if statements. For information on traditional merge fields, please see [Create Merge Fields in Mail Merge](#).

“Expression1” Operator “Expression2” “TrueText” “FalseText”

Below is a very basic example using merge fields as Expressions. The single curly braces ({} ) on either end of the IF statement indicate the beginning and end of a traditional merge field when the field codes have been toggled open.

```
{ IF "«Number1»" > "«Number2»" "Number1 is greater" "Number2 is greater" }
```

## Components of an IF Field

Expression1 & Expression2:

- Both Expression1 and Expression2 represent values that you want to compare.
- For IF fields used with Composer, the expressions are usually merge fields, text strings, or numbers. Expression1 and Expression2 should be surrounded with quotation marks (").

### Operator

- A comparison operator. There is always a space before and after the operator.
- Below is a list of available operators:

Operator	Description
=	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

### TrueText & FalseText

- Content that is shown when the comparison is true (TrueText) or false (FalseText). Each must be surrounded with quotation marks.

⚠ If the content of your TrueText or FalseText in Salesforce contains quotation marks (" or "), you need to replace those quotes with a pair of single quotation marks (' ' instead of " "). Otherwise, Word considers the first quotation mark from your merged data as the end of your text string and the IF statement does not work.

## Creating an IF Field - Conga Best Practices

Especially when comparing the values of merge fields, there are a few best practices to follow when creating IF fields in a Composer Word template. For example, if an Opportunity's Amount is great than \$10,000, we want to show information about a complimentary service package. If it's not, we do not want to show anything. Using this example, we recommend using the following approach:

Procedure Step	Result
Create a traditional merge field with a placeholder name.	<<IF_Amount>>
Toggle open the field codes.	{ MERGEFIELD IF_Amount \* MERGEFORMAT}
Delete the contents of the field, entering IF instead.	{ IF }
Add Expression1 to the IF field. In our example, Expression1 is the Opportunity Amount field.	{ IF "«OPPORTUNITY_AMOUNT»" }
Insert your Operator. We're using greater than (>).	{ IF "«OPPORTUNITY_AMOUNT»" > }
Insert Expression2. Our example is \$10,000.	{ IF "«OPPORTUNITY_AMOUNT»" > "10000" }
Insert TrueText. Our example is a merge field that displays the details of a complimentary service package.	{ IF "«OPPORTUNITY_AMOUNT»" > "10000" "«OPPORTUNITY_SERVICE_PACKAGE»" }
Insert FalseText. In our example, we don't want to display anything if the Opportunity Amount is less than \$10,000 so our FalseText is null, which is a pair of double quotation marks.	{ IF "«OPPORTUNITY_AMOUNT»" > "10000" "«OPPORTUNITY_SERVICE_PACKAGE»" "" }

Our finished IF field looks like this:

```
{ IF "«OPPORTUNITY_AMOUNT»" > "10000" "«OPPORTUNITY_SERVICE_PACKAGE»" "" }
```



An IF statement's field code should look like the field above when it is completed. When creating a merge field in Word, the following text will be included at the end of the field by default: \\* MERGEFORMAT. This text needs to be removed to ensure the IF statement renders consistently.

Or, if we toggle the field codes closed, it reverts to the merge field with the placeholder name:

```
<<IF_Amount>>
```

For example, let's say the Merge Nation Opportunity has an Amount of \$25,000. Once we merge the template that uses this IF field, our merged IF statement looks like this (showing the results of the «OPPORTUNITY\_SERVICE\_PACKAGE» field since that is our TrueText:

**Congratulations! You qualify for our complimentary service package. Thank you for your order.**

Now let's say we merge using a different Opportunity called Merge Republic, which only has an Amount of \$8,000. Our result shows nothing ("") since that is our FalseText.

## Caveats

Regarding Checkbox values: Although the Conga View Data Workbook and the Template Builder show the values as TRUE, they do not work in Microsoft Word. Change your value from TRUE to True.

For example: {If "<<Checkbox\_MergeField>>" = "True" "Yes" "No"}}

When placing an IF field inside a Word table to hide a merge field when it has no value, the IF field does not hide the table row where it resides. The result might be a blank table row. Word IF fields cannot be used to hide some rows within a Word table.

**⚠ IF statements do not work in a Conga Email template. Do not use dynamic culture codes.**

## How to Create Nested Tables in Word Templates

This document outlines how to create a "nested table," in other words, one table built inside of another one. This is particularly useful when you want a multi-row / multi-column detail region to repeat.

**⚠ This method is relatively complex. Another option is the TableGroup function. For more information on using TableGroup see, [Repeating Data in Word and Organize Repeating Data Tables into Groups in Word](#).**

While the screen shots below display traditional merge fields, the process of creating the Word tables is the same whether you are running Conga Composer Release 7 or Release 8 (only the style of merge fields may differ between releases).

This [documentation](#) explains how to nest Word tables but does not include guidance on nesting detail regions when using nested queries in Conga Composer Release 8.

The objective of this exercise is to create a table like the one shown below. The example below is for a related list of Contacts, but the concept is applicable to any list.

Contacts			
<b>Name:</b>	Ms. Rose Gonzalez	<b>Title:</b>	SVP, Procurement
<b>Phone:</b>	(512) 757-6000	<b>Email:</b>	rose@edge.com
<b>Name:</b>	Mr. Sean Forbes	<b>Title:</b>	CFO
<b>Phone:</b>	(512) 757-6051	<b>Email:</b>	sean@edge.com
<b>Name:</b>	Mr. Elias Anderson	<b>Title:</b>	SVP
<b>Phone:</b>	(512) 757-6075	<b>Email:</b>	eanderson@edge.com

In this case, the desired layout for each record is a two-row, four-column table (the “repeating table”) that will be repeated for each item retrieved by a Salesforce report (or SOQL query) configured for use with a Conga Composer® solution:

<b>Name:</b>	Ms. Rose Gonzalez	<b>Title:</b>	SVP, Procurement
<b>Phone:</b>	(512) 757-6000	<b>Email:</b>	rose@edge.com

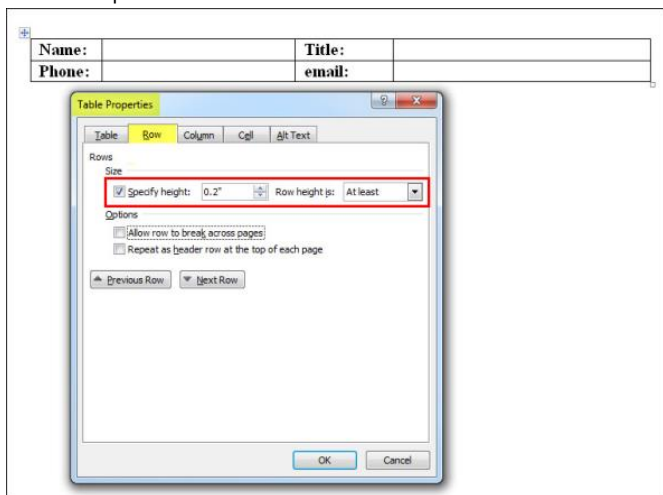
A containing table of three columns and two rows is used to provide a mechanism for controlling presentation of the repeating table:

	<b>Contacts</b>	

This approach is superior to others in that the repeating table can be prevented from “breaking” across pages with the Word option to disallow the breaking of a row across pages (**Table Properties** → **Row** → uncheck “Allow row to break across pages”). A second benefit is the ability to repeat the header row of the table at the top of a subsequent page should the table need to flow across multiple pages.

The steps required to implement this are as follows:

1. Define the desired layout for the repeating items (the interior/nested table) in a stand-alone table. Below we have four columns and two rows. This is the table that will repeat for each result/record in the report or query.
2. Optionally, set the desired row height for each row of the table, or simply allow them to adapt as necessary to accommodate the merge data. We recommend the "At least" option.



3. Insert merge fields into the table you built in step 2.

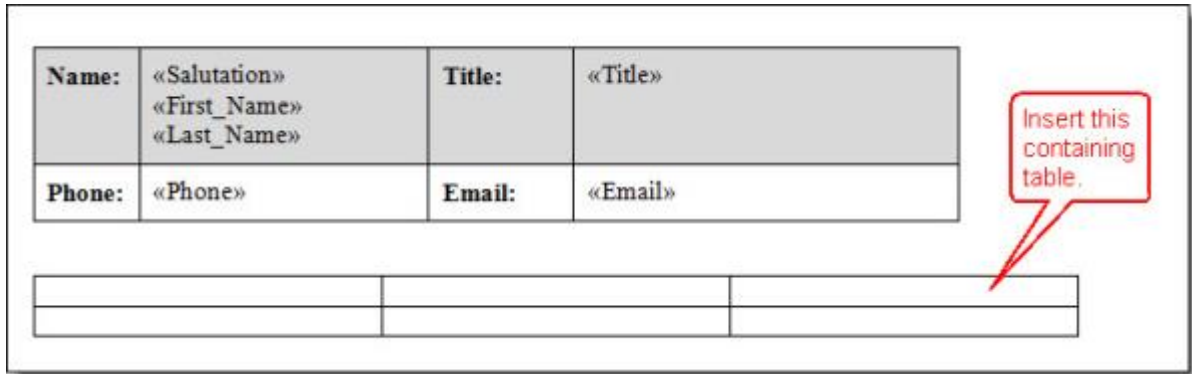
<b>Name:</b>	«Salutation» «First_Name» «Last_Name»	<b>Title:</b>	«Title»
<b>Phone:</b>	«Phone»	<b>Email:</b>	«Email»

4. Adjust the border formatting and row shading, if desired. Here, a color has been chosen for only the top border of the table.

<b>Name:</b>	«Salutation» «First_Name» «Last_Name»	<b>Title:</b>	«Title»
<b>Phone:</b>	«Phone»	<b>Email:</b>	«Email»

5. Next, as a "container" for the repeating table created above, create a second table with three columns and two rows. Insert this table separately from the repeating

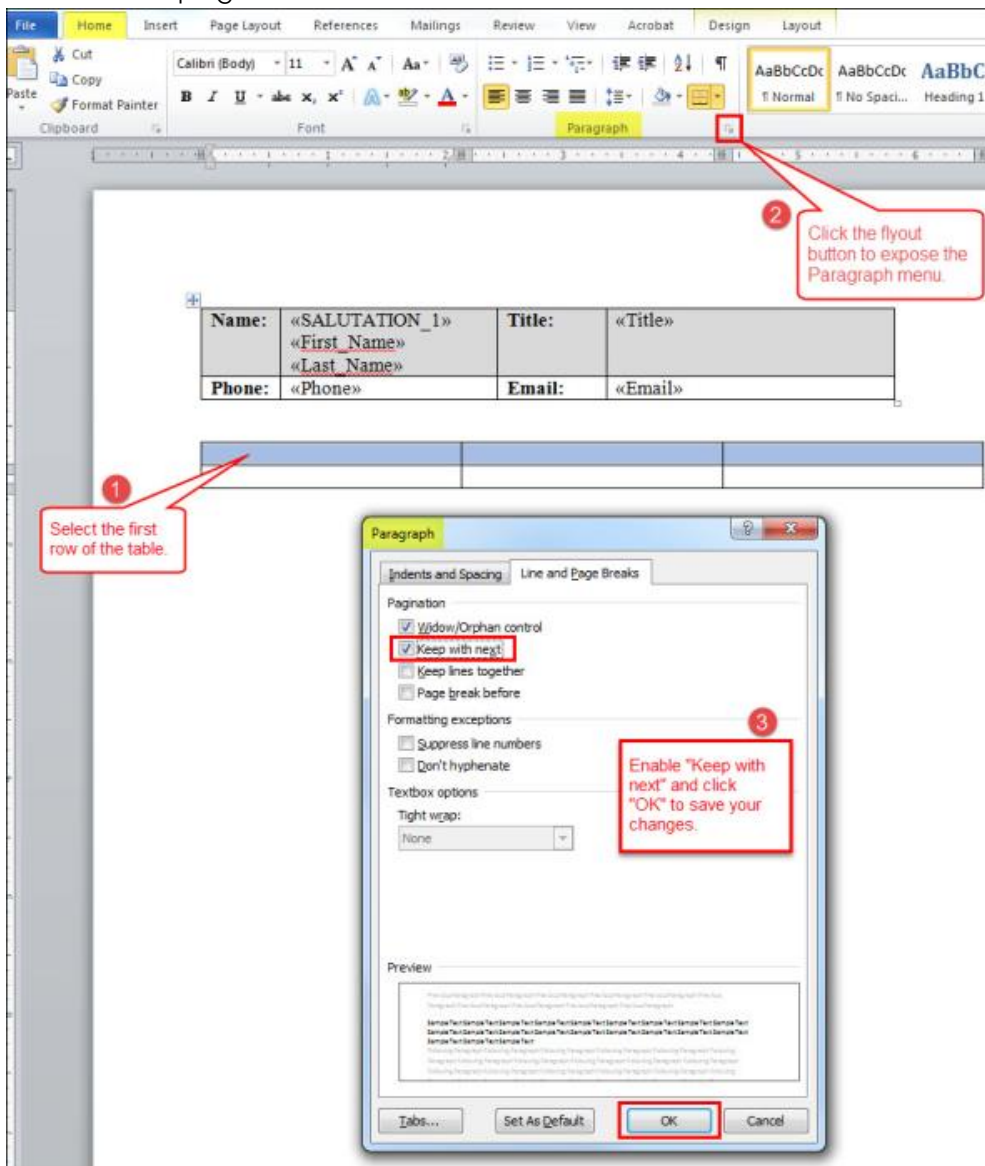
table; we'll nest the two tables in a later step.



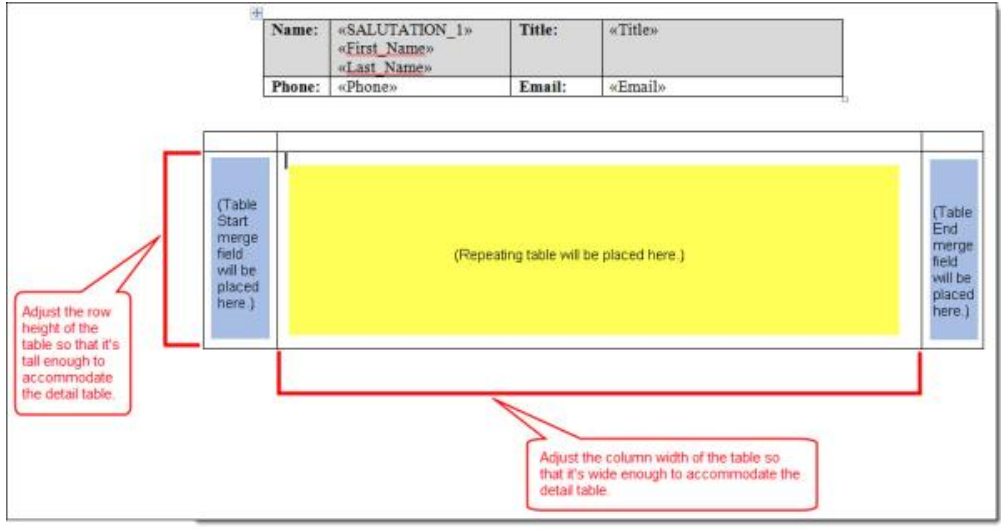
6. In the containing table, select the first row (the "header" row) and from the Paragraph menu in Word, click the Line and Page Breaks tab, and enable the option "Keep with next." This ensures that the header row will never be left standing by itself at the



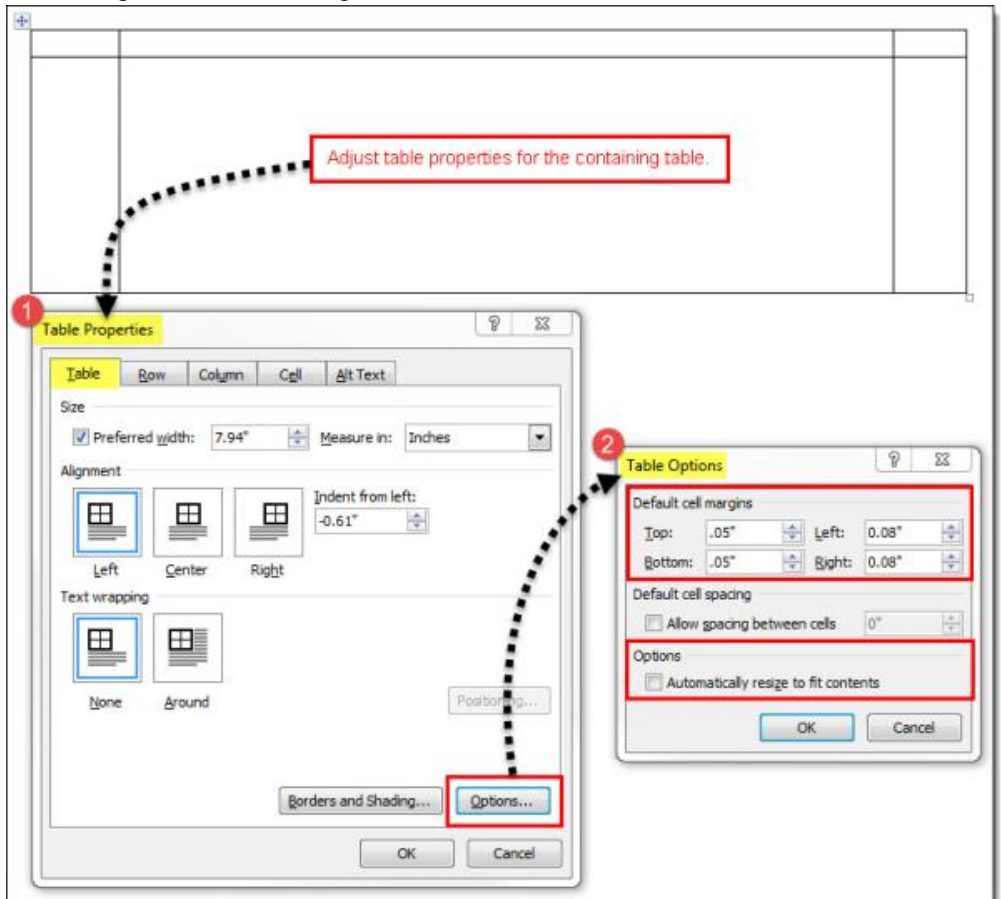
bottom of a page:



- Adjust the height of the second row of the containing table as well as the width of the columns so that the space is large enough to accommodate the repeating table. The middle column of the second row is where the repeating table will be placed while the left and right columns will function as "wings" to place the detail region (TableStart and TableEnd) fields.



- 8. Optionally, adjust the cell margins for the containing table so that the repeating table won't be up against the containing table's borders once you nest it inside. This is done in Table Properties | Options | Default cell margins. We also recommend disabling "Automatically resize to fit contents."



9. Cut and paste the repeating table into the middle column of the second row of the containing table and add your TableStart and TableEnd fields in the "wings" (the left and right columns of the second row).
10. Optionally, add a title to the containing table (here we've used "Contacts"), adjust the row height for the second row of the containing table to make it fit the repeating table a bit better, and turn off the borders for the containing table (we've done this below for all but the bottom border of the middle cell of the top row, which is why the remaining borders are shown as dotted blue lines).



The output from your nested tables will now look like this. We may want to go back and adjust the height of the second row in the containing table so there is not so much white space between each of the repeated tables. We may also want to turn off the border below the heading of "Contacts" if we don't like the look of that border separating each of the repeated tables. Testing your solution is important as it will reveal these sorts of desired formatting tweaks.

Contacts			
<b>Name:</b>	Ms. Rose Gonzalez	<b>Title:</b>	SVP, Procurement
<b>Phone:</b>	(512) 757-6000	<b>Email:</b>	rose@edge.com
<b>Name:</b>	Mr. Sean Forbes	<b>Title:</b>	CFO
<b>Phone:</b>	(512) 757-6051	<b>Email:</b>	sean@edge.com
<b>Name:</b>	Mr. Elias Anderson	<b>Title:</b>	SVP
<b>Phone:</b>	(512) 757-6075	<b>Email:</b>	eanderson@edge.com

## Examples of Nested Tables in Conga Word Templates

Create the black table first, be sure to edit Table Properties so that table will not automatically resize to fit contents

`{{IMAGE:ACCOUNT_PHOTO_URL:W=170}}`

Account Information			
Account Name	{{Account}}	Phone	{{CONTACT_PHONE}}
Address	{{MailingStreet}} {{MailingCity}}, {{MailingState}} {{MailingPostalCode}}	Annual Revenue	{{ACCOUNT_ANNUALREVENUE}}
Website	{{HTML:ACCOUNT_WEBSITE}} {{HYPERLINK:ACCOUNT_WEBSITE :Click Here}}	Valued Member:	{{CHECKBOX:ACCOUNT_VALUED_MEMBER}}
The employees at this location are {{TableList:ReportData:Last_Name;, \*}}			

Insert the TableStart and TableEnd fields into the “wings” of the bottom row of the black table. These “wings” can be skinnier in width so the middle column of the bottom row better matches the width of the top row. The TableStart and Table End will get stretched out vertically, which will force the row to be taller – not to worry, that will go away in the output file as the TableStart and TableEnd disappear.

## How to Embed Non-Standard Fonts in Conga Word Templates

The Conga servers automatically support a wide variety of TrueType fonts. However, if your template uses a non-standard, “embeddable” TrueType font, you may include that font in the template by embedding it into your template file.

Here’s how to embed the font in a Microsoft Word template. Unfortunately, this is supported only in the PC version of Word, not Mac. If you’re a Mac user, we recommend temporarily using a PC in order to embed the font or contact our support team if you’re having difficulty.

In Microsoft Word,

1. Go to **File menu** → **Options**.
2. Click the **Save tab**.
3. Enable Embed fonts in the file. Do not enable “Embed only the characters used in the document” which would prevent merged data from appearing in the desired font.
4. Enable “Do not embed common system fonts” to minimize the file size.

5. Click **OK**.

- ✓ • When you embed a font, the size of your document may increase significantly. Keep in mind that Conga allows templates up to 10 MB.
- Embedded fonts are not currently supported when using the &AWD (Assemble Word Documents) parameter.

## How to Use Bookmarks and References in Word Templates

How to insert Word Bookmarks, add Word References, and use those features within an if statement to get to a desired output.

⚠ This method is not supported by Conga. See a similar warning in this article: [Create Word Calculation Fields](#).

Using Word's Bookmark feature is especially handy when some piece of data or text needs to be referred to at another point in the document. Reference fields make formula fields easier to read and also make changing formulas in the future more manageable. A single Reference field change that is referenced many times throughout a document takes only one update. Here we explain how to insert Word Bookmarks and add Word References to get to the desired output.

### Notes:

- If a formula field can be written in Salesforce® to accomplish the same logic as what is outlined below, that is the preferable option rather than doing so in the Word template.
- Mail Merge is NOT supported.

Inserting Bookmarks:

1. Select the entire data block (merge field or static data) and go to the Insert menu.
2. Choose **Bookmark**.
3. Type in a name for the Bookmark. A simple name is best; spaces are not allowed.
4. Click **Add**.
5. Repeat as necessary for other Bookmarks.

Adding References:

Contacts			
<b>Name:</b>	Ms. Rose Gonzalez	<b>Title:</b>	SVP, Procurement
<b>Phone:</b>	(512) 757-6000	<b>Email:</b>	rose@edge.com
<b>Name:</b>	Mr. Sean Forbes	<b>Title:</b>	CFO
<b>Phone:</b>	(512) 757-6051	<b>Email:</b>	sean@edge.com
<b>Name:</b>	Mr. Elias Anderson	<b>Title:</b>	SVP
<b>Phone:</b>	(512) 757-6075	<b>Email:</b>	eanderson@edge.com

6. Place your cursor in the desired location for the reference to appear. Go to the Insert menu.
7. Click **QuickParts** → **Field**.
8. Choose the Ref option (Alt+I, F, R, R is the keyboard shortcut).
9. Select the appropriate Bookmark to Reference and click **OK**. That Reference is now created.

## How to Compare Two Values in Word Templates using IF Statements

### IF Statements

IF statements (also called IF fields) allow you to compare two values and display document content based on the result of the comparison. When used in a Word template, IF statements are especially useful for comparing the values of Composer merge fields and then displaying the appropriate content.

### Example Use Cases

- If an Account's billing country is not in the United States, display information about international shipping terms. Otherwise, show domestic shipping terms.

- If a Contact's mailing address is blank, show the billing address from his related Account. If it's not blank, show the Contact's mailing address.
- If an Opportunity's amount is greater than \$10,000, show information about a complimentary service package. If not, show nothing.

**⚠** Nested IF statements in Word, where the *else* value is another IF statement, work Conga Composer but are not recommended. **Conga Technical Support does not support nested IF statements.** Conga highly recommends evaluating nested IF statement logic using Salesforce formula fields and then merging the formula field values into the document.

- Watch this video on [How to Add IF Statements to Templates](#).

## Create IF Statements in a Word Template

**⚠** IF statements can use Word merge fields or text-based merge fields, with one exception: the IF statement itself must be a Word merge field. All the fields inside or around the IF statement can be either Word or text-based merge fields, but the IF statement itself must be a Word merge field. Examples using both types of syntax are displayed below. For information on traditional merge fields, see [Create Merge Fields in Mail Merge](#).

### IF Statement Structure

"Expression1" Operator "Expression2" "TrueText" "FalseText"

Below is a basic example using Word merge fields as Expressions. The single braces ({ }) on either end of the IF statement indicate the beginning and end of a Word merge field when the field codes are toggled open.

Example:

```
{ IF "«Number1»" > "«Number2»" "Number1 is greater" "Number2 is greater" }
```


### Components of an IF Field

- Expression1 and Expression2
  - Both Expression1 and Expression2 represent values that you want to compare.
  - For IF fields used with Composer, the expressions are usually Word or text-based merge fields, text strings, or numbers. Expression1 and Expression2 should be surrounded with quotation marks ("").
- Operator

- A comparison operator. There is always a space before and after the operator.
- Below is a list of available operators:

Operator	Description
=	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

- TrueText and FalseText
  - Content that is displayed when the comparison is true (TrueText) or false (FalseText). Each must be surrounded with quotation marks.

 If the content of your TrueText or FalseText in Salesforce contains quotation marks ( " or " ), replace those quotes with a pair of single quotation marks ( ' instead of " ). Otherwise, Word considers the first quotation mark from your merged data as the end of your text string and the IF statement does not work.

## Creating an IF Statement - Best Practices

Follow best practices to ensure the stability and sustainability of your IF statements. In the example below, we display in detail the best practice method for creating IF statements. We also show the syntax for both Word merge fields and text-based merge fields.

**Example Scenario:** If the Opportunity Amount is greater than \$10,000, display information about a complimentary service package. If it's not, do not show anything.


Procedure Step	Result
Create a traditional merge field with a placeholder name.	<<IF_Amount > 10000>>
Open the merge field by toggling field codes.	{ MERGEFIELD IF_Amount \* MERGEFORMAT}
Delete the contents of the field, entering IF instead.	{ IF }
Add Expression1 to the IF field. In our example, Expression1 is the Opportunity Amount field.	Word: { IF "«OPPORTUNITY_AMOUNT»" } Text-based: { IF "{{OPPORTUNITY_AMOUNT}}" }



Insert your Operator. We're using greater than (>).	Word: { IF "«OPPORTUNITY_AMOUNT»" > } Text-based: { IF "{{OPPORTUNITY_AMOUNT}}" > } > }
Insert Expression2. Our example is \$10,000.	Word: { IF "«OPPORTUNITY_AMOUNT»" > "10000" } Text-based: { IF "{{OPPORTUNITY_AMOUNT}}" > "10000" }
Insert TrueText. Our example is a merge field that displays the details of a complimentary service package.	Word: { IF "«OPPORTUNITY_AMOUNT»" > "10000" "«OPPORTUNITY_SERVICE_PACKAGE»" } Text-based: { IF "{{OPPORTUNITY_AMOUNT}}" > "10000" "{{OPPORTUNITY_SERVICE_PACKAGE}}" }
Insert FalseText. In our example, we don't want to display anything if the Opportunity Amount is less than \$10,000, so our FalseText is null. This is represented by a pair of quotation marks.	Word: { IF "«OPPORTUNITY_AMOUNT»" > "10000" "«OPPORTUNITY_SERVICE_PACKAGE»" "" } Text-based: { IF "{{OPPORTUNITY_AMOUNT}}" > "10000" "{{OPPORTUNITY_SERVICE_PACKAGE}}" "" } "" }

Our finished IF field looks like this:

- Word IF Field: { IF "«OPPORTUNITY\_AMOUNT»" > "10000" "«OPPORTUNITY\_SERVICE\_PACKAGE»" "" }
- Text-based IF Field: { IF "{{OPPORTUNITY\_AMOUNT}}" > "10000" "{{OPPORTUNITY\_SERVICE\_PACKAGE}}" "" }

 An IF statement that is toggled open should look exactly like the examples above when completed. When initially creating the IF statement, Word includes the following text at the end: \\* MERGEFORMAT. This text needs to be removed to ensure the IF statement renders consistently.

When we toggle the field codes closed, the IF statement reverts to the Word merge field with the placeholder name: «IF\_Amount»


Using the IF Statement above, let's consider the following example. The Merge Nation Opportunity has an amount of \$25,000. Once we merge the Composer template that uses our IF statement, our merged document will look like this:

**Congratulations! You qualify for our complimentary service package. Thank you for your order.**

We display the value of the OPPORTUNITY\_SERVICE\_PACKAGE field since that is our TrueText. If we use a different Opportunity called Merge Republic, which only has an amount of \$8,000, the merged document will show nothing (""), since that is our FalseText.

## Caveats

- Checkbox fields
  - Although the Composer View Data Workbook and the Template Builder show the values of checkbox fields as TRUE and FALSE, these values do not work in an IF statement. Change your value from TRUE to True.
  - Example: { IF "<<Checkbox\_MergeField>>" = "True" "Yes" "No" }
- Tables
  - IF statements placed inside a table row cannot hide the table row when the statement evaluates to show nothing. You will instead have a blank table row. Word IF statements cannot be used to hide rows within a table.
- Salesforce number fields
  - Do not put double quotes around a Salesforce number field's corresponding merge field in an IF Statement. If double quotes surround the number field's merge field, it will not merge as expected. See the example below for the correct syntax.
  - Example syntax for a number field: { IF <<Item\_Number\_Field>> > "2" "Free Shipping" "Free Shipping Not Included" }
- When using the AUTOSUM(ABOVE) or SUM(ABOVE) functions to total an international currency field that dynamically formats currency with the [currency culture parameter](#) within an IF statement, the merged total amount displays in the running user's currency (determined by their locale), as opposed to the value determined by the currency culture parameter.
  - A workaround for this scenario is building a Salesforce formula field that converts the international currency field's value into a number value. Then use an aggregated SUM query for that formula field and use the [\# Currency picture switch](#) in the merge field added to the template.

 IF statements do not work in a Conga Email template. Do not use dynamic culture codes.


## Inserting IF Statements into a Word Template with Word for Mac

This is Microsoft Office functionality and Conga support is limited. Refer to Microsoft Office documentation for additional information about "IF" statements.

Regarding Checkbox values: Although the Conga View Data Workbook and the Template Builder show the value as TRUE, they do not work in Microsoft Word. Change your value from TRUE to True.

For example: {If "<<Checkbox\_MergeField>>" = "True" "Yes" "No"}}

Nested IF statements, where the else value is another conditional IF statements, work successfully with Conga Composer.

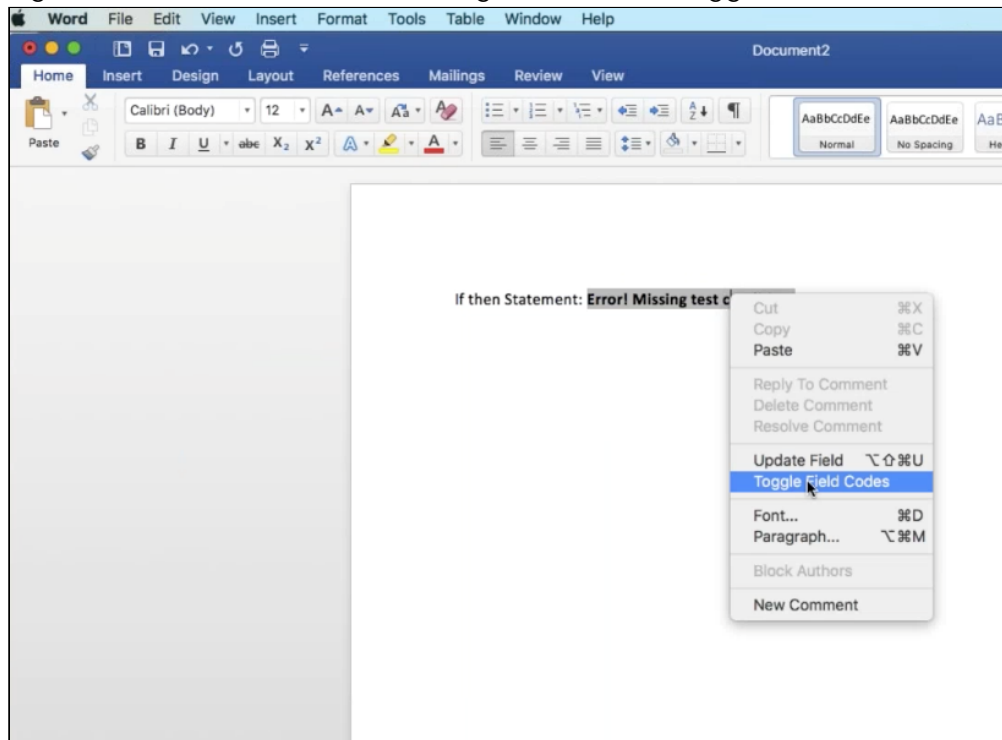
 Due to the complicated nature of nested IF statements, Conga does not support nested IF statements in Microsoft Word. The recommended alternatives to nested IF statements in Microsoft Word is to use the native IF and CASE Logical Functions in Salesforce formulas.

Mac users can insert If Statements in Word templates with Microsoft Word.

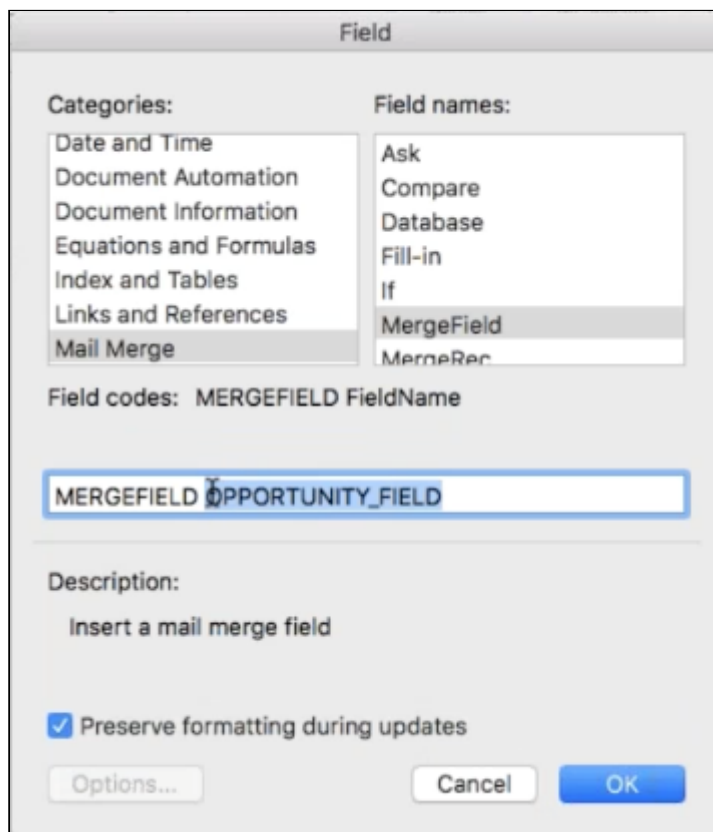
To insert If statements:

1. In Word, select **Insert** → **Field**.
2. Select **Mail Merge** in the Category section, and IF in the Field Names section. Click **OK**. You see what looks like an error message.

3. Right-mouse click the error message and select Toggle Field Codes.



4. Remove the text: "\\* MERGEFORMAT", leaving the word IF between the brackets.
5. The syntax for an IF statement is: {IF <<Expression1>> Operator ExpressionValue "TrueText" "FalseText" }
6. To insert Expression1, ensure a space remains after IF, then from the Insert drop-down menu, again select the Field option.
7. Select the **Mail Merge** option in the Category section, select **MergeField** in the Field Names section, and paste the field name. Click **OK**.



Now you're ready to write the rest of the IF Statement. Here's an example of proper syntax:

**Example:** { IF <<OPPTY\_FIELD>> = "True" "Then Populate Data" "Do Not Populate Data" }

## Caveats

Using Preview in the Conga UI does not always work properly. If you receive an error, run the complete solution to test instead. If you need to replace the field in the Expression1 part of the IF Statement, you cannot delete the field name between the double-chevrons ("<<<" and ">>") and replace with the new field. You must replace the expression by inserting an alternate field using the Insert Field process again. If the final document produced from running the solution does not display the results of the IF statement correctly, this might be resolved by simply right-clicking the IF Statement and selecting Toggle Field Codes.

Regarding Checkbox values: Although the Conga View Data Workbook and the Template Builder show the value as TRUE, they do not work in Microsoft Word.

Change your value from TRUE to True.

- For example: {If "<<checkbox\_MergeField>>" = "True" "Yes" "No"}}


## Linetrिम

Use the linetrिम picture switch in a Word template to eliminate extraneous spaces from rich text fields.

Extraneous spaces are sometimes inserted when using Composer to merge rich text fields that contain bulleted lists. Add the linetrिम picture switch to a merge field's syntax to eliminate extraneous spaces.

The linetrिम picture switch works specifically for rich text fields that contain bulleted lists merged in a Word template. Linetrिम is used in conjunction with [HTML function](#) in a merge field's syntax.

- **Linetrिम Syntax:** \c linetrिम
- **Merge Field Example:** {{HTML:ACCOUNT\_LIST\_RTf \c linetrिम}}

 Linetrिम cannot be used in conjunction with [tablelinetrिम](#).

## Master Data Conditional TableHide in Word

The Master Data Conditional TableHide is a merge field comprised of three elements and allows a table to be hidden if the Master field referenced contains a particular value.

 We do not advise using more than 25 TableHide fields in a single template.

<b>Syntax</b>	<code>{{TableHide:MasterField:Value}}</code>
<b>Values</b>	<p>TableHide: is the TableHide field prefix</p> <p>MasterField is the name of the Master field you want to evaluate (e.g. Special_Terms), as it appears in the Template Builder</p> <p>Value is the value of the field that will force Composer® to hide the table</p>
<b>Usage</b>	<p>Congratulations, your {{ACCOUNT_SLA}} service level agreement entitles you to a 10% discount on your next order! {{TableHide:ACCOUNT_SLA:Bronze}}</p> <p>On merge, all levels other than Bronze display the discount offer:</p> <p>Congratulations, your Silver service level agreement entitles you to a 10% discount on your next order!</p>

## Merge Google Maps into a Word Template

How to merge Google maps into a Word template.

⚠️ Google Map integration is beyond the scope of Conga Support and may require Conga Professional Services.

To merge a Google Map into a Word template:

1. Create a formula field with a Return Type of text. For example:

```
IMAGE("'" + SUBSTITUTE(BillingStreet, " ", "+") + "," +
SUBSTITUTE(BillingCity, " ", "+") + "," + BillingState + " " +
BillingPostalCode + "&zoom=17&size=400x200
&motype=road&markers=color:green|" + SUBSTITUTE(BillingStreet, " ", "+") +
"," + SUBSTITUTE(BillingCity, " ", "+") + "," + BillingState + " " +
BillingPostalCode + "&sensor=false", "No map available", 200, 400)
```

The zoom, size, color, and whether you want a satellite or roadmap is customizable. The address fields in the formula. The Google Map API will need to be generated directly from [Google](#) and is placed in the API section.

⚠️ Conga Support does not support advanced Salesforce Formula Fields, please contact Salesforce for further configuration.

2. Use `{{HTML:FieldName}}` as the merge field, in which Field\_Name references the formula field created above.

## Merging Images from Salesforce Files

How to merge an image from Salesforce Files into Microsoft.

You can merge images from Salesforce Files in Conga Composer.

✔️ We highly recommend queries and formulas are copied and pasted into supported Text Editors. Ensure single-quote marks are used. Avoid curly brackets and double-quote marks.

To merge an image from Salesforce Files:

1. Build a custom formula field on the Salesforce Files Object to form a URL to the Content document: (Salesforce Classic) Navigate to **Setup** → **Build** → **Customize** → **Salesforce Files** → **Field**

**Content Version Fields**

This page allows you to specify the fields available for files. You can create up to 500 custom Content Version fields.

**Content Version Standard Fields**

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed	Track History
Edit	Description	Description	Text Area(1000)			<input type="checkbox"/>
Edit	File Type	FileType	Text(20)			<input type="checkbox"/>
Edit	Language	Language	Picklist			<input type="checkbox"/>
Edit	Owner	Owner	Lookup(User)		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Edit	Prevent others from sharing and unsharing	SharingOption	Picklist			<input checked="" type="checkbox"/>
Edit	Tags	TagCev	Text Area(2000)			<input type="checkbox"/>
Edit	Title	Title	Text(255)			<input type="checkbox"/>

**Content Version Custom Fields & Relationships**

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By	Track History
Edit   Del	Field Image URL	Field_image_URL__c	Formula (Text)			GeoDemo.Evangelista 11/05/2017 11:13 AM	<input type="checkbox"/>

- (Salesforce Lightning) Navigate to **Setup** → **Object Manager** → **Content Version** → **Fields and Relationships** → **New**
2. Create a new formula Field.
    - For example, Field Image URL: `LEFT( $Api.Enterprise_Server_URL_100, FIND( ".com", $Api.Enterprise_Server_URL_100 ) + 4)& "sfc/servlet.shepherd/version/renditionDownload?rendition=THUMB720BY480&versionId=" & Id & "&operationContext=CHATTER"`
  3. Write a Conga query to retrieve the formula field. Use the dot syntax to go from the ContentDocumentLink Object up through the ContentDocument through the LatestPublishedVersion Object to retrieve the custom formula:
 

```
SELECT ContentDocument.Latestpublishedversion.Field_Image_URL__c FROM ContentDocumentLink WHERE LinkedEntityId = '{pv0}' AND ContentDocument.FileType IN ('PNG', 'JPG', 'JPEG')
```
  4. Use the Detail Regions and the **IMAGE** special merge field to merge the image into the template.
 

For example in Microsoft Word: `{{TableStart:FileImage}}`

`{{IMAGE:LATESTPUBLISHEDVERSION_FIELD_IMAGE_URL:W=350}}{{TableEnd:FileImage}}`

## Merge Images with Salesforce ID in Word

You can include an image into a template document by using a Salesforce ID instead of a complete URL address.

If the image exists in Salesforce as a Document (record ID beginning in "015") or Attachment (record ID beginning in "00P"), you can reference the ID of the Document or Attachment and Conga Composer calculates the rest of the image URL. You will still use the field prefix **IMAGE:** but do not have to build a custom field to house the image's URL. Instead, you retrieve the image's ID from a SOQL query.

**⚠ Getting attachment images to embed in a document only works with Word and is not an option with Conga Email, PDF, or Excel templates.**



You have uploaded an image as an Attachment (on the Notes & Attachments related list) on an Account and want to merge that image into your Word® template. Rather than building a custom field on your Account to house the URL of that image, you reference the image by its Salesforce ID.

First, build a SOQL query to retrieve the ID of that Attachment: `SELECT Id FROM Attachment WHERE ParentId = '{pv0}'`

Alternately, if you only want to return Attachments that contain .jpg, .gif, or .png in the file name (to indicate that you only want images retrieved), you could use this SOQL query:

```
SELECT Id FROM Attachment WHERE ParentId = '{pv0}' AND (Name LIKE '%.jpg%' OR Name LIKE '%.gif%' OR Name LIKE '%.png%')
```

Once you have the query created you need to add the query ID to the button you are using for the solution. You add the query ID by using the QueryID parameter:

```
&QueryID=[Attachment]a0Ai0000009hdid
```

Lastly, you reference the Id field in your template, using the IMAGE: prefix:

```
{{TableStart:Attachment}}{{IMAGE:id}}{{TableEnd:Attachment}}
```

The image then populates in the merged output file. To control the file size, use:

```
{{TableStart:Attachment}}{{IMAGE:id:W=350}}{{TableEnd:Attachment}}
```

## Nested SOQL Queries

In Composer Release 8, detail regions can be nested inside one another if the detail data is retrieved through a nested SOQL query.

⚠ Building nested queries are not supported by Conga Support. This functionality is supported in Microsoft Word templates only.

### Example

```
{{TableStart:QueryData}}
```

```
{{ACCOUNT_NAME}} {{ACCOUNT_BILLINGADDRESS}}
```

Name	Title
<pre>{{TableStart:QueryData_SubQueryData}} {{First_Name}} {{Last_Name}}</pre>	<pre>{{Title}} {{TableEnd:QueryData_SubQueryData}}</pre>

```
{{TableEnd:QueryData}}
```

## Benefits of Using Nested Queries

Nested queries—also known as a left outer join relationship—are a top-down approach to creating a SOQL query.

A classic example of this would be the standard Account object with many child objects that are unrelated to one another: Opportunities, Contacts, Cases, and Attachments. A Joined report would be necessary in order to create a Salesforce report incorporating all of these objects.

**⚠ Conga Composer does not support joined reports.**

When using nested queries, you should be familiar with writing SOQL queries. For assistance, see:

- [Force.com SOQL Reference](#)
- [A Deeper Look at SOQL and Relationship Queries on Force.com](#)

## The Conga Query Manager section of the Conga Composer Product Guide

Using Nested Queries in Word Templates:

Perhaps you would like to create a Word® document showing the Account information and underneath that display all open Opportunities, recent Cases, active Contacts, and any attached photos (all related to that initial Account), followed by the next Account with all open Opportunities, recent Cases, active Contacts, and so on. You could accomplish this using a nested query, which would look something like this:

```
SELECT Name, ID,(SELECT Name, Amount FROM Opportunities),
(SELECT Id, CaseNumber FROM Cases),
(SELECT Id, LastName, Email FROM Contacts),
(SELECT Id FROM Attachments)FROM Account
ORDER BY Name ASC
```

**⚠ Caution the query above the output below produces Org Wide data that is NOT filtered. You should use this style of a query if you want to limit the number of Accounts to an individual record.**

```
SELECT Name, ID,
(SELECT Name, Amount FROM Opportunities),
(SELECT Id, CaseNumber FROM Cases),
(SELECT Id, LastName, Email FROM Contacts),
```

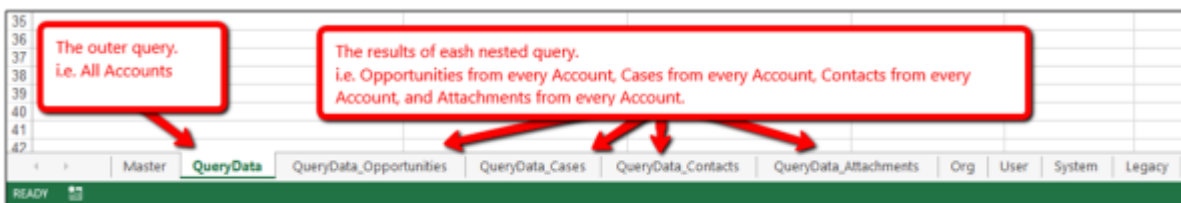
```
(SELECT Id FROM Attachments)FROM Account
Where ID = '{pv0}'
ORDER BY Name ASC
```

**⚠ Results below are Org Wide**

This query would return the following results:

Name	Id	Opportunity	Case	Contact	Attachment																																																
1 Aethna Home Products	001E000000028MvUJAE	<table border="1"> <thead> <tr> <th>Name</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1 Commercial Generators</td> <td>1985000.0</td> </tr> <tr> <td>2 Residential Generators</td> <td>1920000.0</td> </tr> <tr> <td>3 Hospital Generators</td> <td>905000.0</td> </tr> </tbody> </table>	Name	Amount	1 Commercial Generators	1985000.0	2 Residential Generators	1920000.0	3 Hospital Generators	905000.0	<table border="1"> <thead> <tr> <th>Id</th> <th>CaseNumber</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028PndIAC</td> <td>00001027</td> </tr> <tr> <td>2 002E000000028PndIAC</td> <td>00001026</td> </tr> </tbody> </table>	Id	CaseNumber	1 002E000000028PndIAC	00001027	2 002E000000028PndIAC	00001026	<table border="1"> <thead> <tr> <th>Id</th> <th>LastName</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028PndIAC</td> <td>Akan</td> <td>castrap1344@gmail.com</td> </tr> <tr> <td>2 002E000000028PndIAC</td> <td>James</td> <td>jgibbons15@intechus.com</td> </tr> <tr> <td>3 002E000000028PndIAC</td> <td>Smith</td> <td>jgibbons15@intechus.com</td> </tr> <tr> <td>4 002E000000028PndIAC</td> <td>Rodriguez</td> <td>castrap1344@gmail.com</td> </tr> </tbody> </table>	Id	LastName	Email	1 002E000000028PndIAC	Akan	castrap1344@gmail.com	2 002E000000028PndIAC	James	jgibbons15@intechus.com	3 002E000000028PndIAC	Smith	jgibbons15@intechus.com	4 002E000000028PndIAC	Rodriguez	castrap1344@gmail.com	<table border="1"> <thead> <tr> <th>Id</th> </tr> </thead> <tbody> <tr> <td>1 00PE000000028PndIAC</td> </tr> <tr> <td>2 00PE000000028PndIAC</td> </tr> </tbody> </table>	Id	1 00PE000000028PndIAC	2 00PE000000028PndIAC																
Name	Amount																																																				
1 Commercial Generators	1985000.0																																																				
2 Residential Generators	1920000.0																																																				
3 Hospital Generators	905000.0																																																				
Id	CaseNumber																																																				
1 002E000000028PndIAC	00001027																																																				
2 002E000000028PndIAC	00001026																																																				
Id	LastName	Email																																																			
1 002E000000028PndIAC	Akan	castrap1344@gmail.com																																																			
2 002E000000028PndIAC	James	jgibbons15@intechus.com																																																			
3 002E000000028PndIAC	Smith	jgibbons15@intechus.com																																																			
4 002E000000028PndIAC	Rodriguez	castrap1344@gmail.com																																																			
Id																																																					
1 00PE000000028PndIAC																																																					
2 00PE000000028PndIAC																																																					
2 American Banking Corp.	001E000000028MvUJAE		<table border="1"> <thead> <tr> <th>Name</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1 American Banking Corp.</td> <td></td> </tr> </tbody> </table>	Name	Amount	1 American Banking Corp.		<table border="1"> <thead> <tr> <th>Id</th> <th>LastName</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028PndIAC</td> <td>Abbott</td> <td>egert.psum.donac@loremvehicula.org</td> </tr> <tr> <td>2 002E000000028PndIAC</td> <td>Aguirre</td> <td>erim@mitus.edu</td> </tr> <tr> <td>3 002E000000028PndIAC</td> <td>Accosta</td> <td>non@justo.eu.ca</td> </tr> <tr> <td>4 002E000000028PndIAC</td> <td>Agular</td> <td>curabitur@donecporttortellus.edu</td> </tr> <tr> <td>5 002E000000028PndIAC</td> <td>Adams</td> <td>lorem.ut aliquam@scis.com</td> </tr> <tr> <td>6 002E000000028PndIAC</td> <td>Abbott</td> <td>castrap1344@gmail.com</td> </tr> <tr> <td>7 002E000000028PndIAC</td> <td>Abbott</td> <td>jack@sihon.edu</td> </tr> <tr> <td>8 002E000000028PndIAC</td> <td>Bar</td> <td>bbair@abankingco.com</td> </tr> <tr> <td>9 002E000000028PndIAC</td> <td>Abbott</td> <td>ntrum.justo@idmagna.edu</td> </tr> <tr> <td>10 002E000000028PndIAC</td> <td>Accosta</td> <td>molis.vitae.posuere@dictum.ca</td> </tr> </tbody> </table>	Id	LastName	Email	1 002E000000028PndIAC	Abbott	egert.psum.donac@loremvehicula.org	2 002E000000028PndIAC	Aguirre	erim@mitus.edu	3 002E000000028PndIAC	Accosta	non@justo.eu.ca	4 002E000000028PndIAC	Agular	curabitur@donecporttortellus.edu	5 002E000000028PndIAC	Adams	lorem.ut aliquam@scis.com	6 002E000000028PndIAC	Abbott	castrap1344@gmail.com	7 002E000000028PndIAC	Abbott	jack@sihon.edu	8 002E000000028PndIAC	Bar	bbair@abankingco.com	9 002E000000028PndIAC	Abbott	ntrum.justo@idmagna.edu	10 002E000000028PndIAC	Accosta	molis.vitae.posuere@dictum.ca	<table border="1"> <thead> <tr> <th>Id</th> </tr> </thead> <tbody> <tr> <td>1 00PE000000028PndIAC</td> </tr> <tr> <td>2 00PE000000028PndIAC</td> </tr> <tr> <td>3 00PE000000028PndIAC</td> </tr> <tr> <td>4 00PE000000028PndIAC</td> </tr> <tr> <td>5 00PE000000028PndIAC</td> </tr> <tr> <td>6 00PE000000028PndIAC</td> </tr> <tr> <td>7 00PE000000028PndIAC</td> </tr> <tr> <td>8 00PE000000028PndIAC</td> </tr> <tr> <td>9 00PE000000028PndIAC</td> </tr> <tr> <td>10 00PE000000028PndIAC</td> </tr> </tbody> </table>	Id	1 00PE000000028PndIAC	2 00PE000000028PndIAC	3 00PE000000028PndIAC	4 00PE000000028PndIAC	5 00PE000000028PndIAC	6 00PE000000028PndIAC	7 00PE000000028PndIAC	8 00PE000000028PndIAC	9 00PE000000028PndIAC	10 00PE000000028PndIAC
Name	Amount																																																				
1 American Banking Corp.																																																					
Id	LastName	Email																																																			
1 002E000000028PndIAC	Abbott	egert.psum.donac@loremvehicula.org																																																			
2 002E000000028PndIAC	Aguirre	erim@mitus.edu																																																			
3 002E000000028PndIAC	Accosta	non@justo.eu.ca																																																			
4 002E000000028PndIAC	Agular	curabitur@donecporttortellus.edu																																																			
5 002E000000028PndIAC	Adams	lorem.ut aliquam@scis.com																																																			
6 002E000000028PndIAC	Abbott	castrap1344@gmail.com																																																			
7 002E000000028PndIAC	Abbott	jack@sihon.edu																																																			
8 002E000000028PndIAC	Bar	bbair@abankingco.com																																																			
9 002E000000028PndIAC	Abbott	ntrum.justo@idmagna.edu																																																			
10 002E000000028PndIAC	Accosta	molis.vitae.posuere@dictum.ca																																																			
Id																																																					
1 00PE000000028PndIAC																																																					
2 00PE000000028PndIAC																																																					
3 00PE000000028PndIAC																																																					
4 00PE000000028PndIAC																																																					
5 00PE000000028PndIAC																																																					
6 00PE000000028PndIAC																																																					
7 00PE000000028PndIAC																																																					
8 00PE000000028PndIAC																																																					
9 00PE000000028PndIAC																																																					
10 00PE000000028PndIAC																																																					
3 Burlington Textiles Corp of America	001E000000028MvUJAE	<table border="1"> <thead> <tr> <th>Name</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1 Burlington Textiles Weaving Plant Generator</td> <td>235000.0</td> </tr> <tr> <td>2 Burlington Textiles Weaving Plant Generator</td> <td></td> </tr> </tbody> </table>	Name	Amount	1 Burlington Textiles Weaving Plant Generator	235000.0	2 Burlington Textiles Weaving Plant Generator		<table border="1"> <thead> <tr> <th>Id</th> <th>CaseNumber</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028MvUJAE</td> <td>00001020</td> </tr> <tr> <td>2 002E000000028MvUJAE</td> <td>00001019</td> </tr> </tbody> </table>	Id	CaseNumber	1 002E000000028MvUJAE	00001020	2 002E000000028MvUJAE	00001019	<table border="1"> <thead> <tr> <th>Id</th> <th>LastName</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028MvUJAE</td> <td>Rogers</td> <td>jgibbons@intechus.com</td> </tr> </tbody> </table>	Id	LastName	Email	1 002E000000028MvUJAE	Rogers	jgibbons@intechus.com	<table border="1"> <thead> <tr> <th>Id</th> </tr> </thead> <tbody> <tr> <td>1 00PE000000028MvUJAE</td> </tr> <tr> <td>2 00PE000000028MvUJAE</td> </tr> <tr> <td>3 00PE000000028MvUJAE</td> </tr> <tr> <td>4 00PE000000028MvUJAE</td> </tr> <tr> <td>5 00PE000000028MvUJAE</td> </tr> <tr> <td>6 00PE000000028MvUJAE</td> </tr> <tr> <td>7 00PE000000028MvUJAE</td> </tr> <tr> <td>8 00PE000000028MvUJAE</td> </tr> <tr> <td>9 00PE000000028MvUJAE</td> </tr> <tr> <td>10 00PE000000028MvUJAE</td> </tr> </tbody> </table>	Id	1 00PE000000028MvUJAE	2 00PE000000028MvUJAE	3 00PE000000028MvUJAE	4 00PE000000028MvUJAE	5 00PE000000028MvUJAE	6 00PE000000028MvUJAE	7 00PE000000028MvUJAE	8 00PE000000028MvUJAE	9 00PE000000028MvUJAE	10 00PE000000028MvUJAE																			
Name	Amount																																																				
1 Burlington Textiles Weaving Plant Generator	235000.0																																																				
2 Burlington Textiles Weaving Plant Generator																																																					
Id	CaseNumber																																																				
1 002E000000028MvUJAE	00001020																																																				
2 002E000000028MvUJAE	00001019																																																				
Id	LastName	Email																																																			
1 002E000000028MvUJAE	Rogers	jgibbons@intechus.com																																																			
Id																																																					
1 00PE000000028MvUJAE																																																					
2 00PE000000028MvUJAE																																																					
3 00PE000000028MvUJAE																																																					
4 00PE000000028MvUJAE																																																					
5 00PE000000028MvUJAE																																																					
6 00PE000000028MvUJAE																																																					
7 00PE000000028MvUJAE																																																					
8 00PE000000028MvUJAE																																																					
9 00PE000000028MvUJAE																																																					
10 00PE000000028MvUJAE																																																					
4 Dickenson plc	001E000000028MvUJAE		<table border="1"> <thead> <tr> <th>Name</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1 NEW</td> <td>600000.0</td> </tr> <tr> <td>2 OLD</td> <td>675000.0</td> </tr> <tr> <td>3 Dickenson Mobile Generators</td> <td>1035000.0</td> </tr> </tbody> </table>	Name	Amount	1 NEW	600000.0	2 OLD	675000.0	3 Dickenson Mobile Generators	1035000.0	<table border="1"> <thead> <tr> <th>Id</th> <th>LastName</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1 002E000000028MvUJAE</td> <td>Young</td> <td>a_young@dickenson.com</td> </tr> </tbody> </table>	Id	LastName	Email	1 002E000000028MvUJAE	Young	a_young@dickenson.com	<table border="1"> <thead> <tr> <th>Id</th> </tr> </thead> <tbody> <tr> <td>1 00PE000000028MvUJAE</td> </tr> <tr> <td>2 00PE000000028MvUJAE</td> </tr> <tr> <td>3 00PE000000028MvUJAE</td> </tr> <tr> <td>4 00PE000000028MvUJAE</td> </tr> <tr> <td>5 00PE000000028MvUJAE</td> </tr> <tr> <td>6 00PE000000028MvUJAE</td> </tr> <tr> <td>7 00PE000000028MvUJAE</td> </tr> <tr> <td>8 00PE000000028MvUJAE</td> </tr> <tr> <td>9 00PE000000028MvUJAE</td> </tr> <tr> <td>10 00PE000000028MvUJAE</td> </tr> </tbody> </table>	Id	1 00PE000000028MvUJAE	2 00PE000000028MvUJAE	3 00PE000000028MvUJAE	4 00PE000000028MvUJAE	5 00PE000000028MvUJAE	6 00PE000000028MvUJAE	7 00PE000000028MvUJAE	8 00PE000000028MvUJAE	9 00PE000000028MvUJAE	10 00PE000000028MvUJAE																							
Name	Amount																																																				
1 NEW	600000.0																																																				
2 OLD	675000.0																																																				
3 Dickenson Mobile Generators	1035000.0																																																				
Id	LastName	Email																																																			
1 002E000000028MvUJAE	Young	a_young@dickenson.com																																																			
Id																																																					
1 00PE000000028MvUJAE																																																					
2 00PE000000028MvUJAE																																																					
3 00PE000000028MvUJAE																																																					
4 00PE000000028MvUJAE																																																					
5 00PE000000028MvUJAE																																																					
6 00PE000000028MvUJAE																																																					
7 00PE000000028MvUJAE																																																					
8 00PE000000028MvUJAE																																																					
9 00PE000000028MvUJAE																																																					
10 00PE000000028MvUJAE																																																					

In the Conga Composer View Data workbook, each aspect of the query is parsed into its own worksheet one tab for the outer query (the Accounts) and another tab for each nested query (Opportunities, Contacts, Cases, and Attachments).



The naming convention of the worksheets in the View Data workbook is "DatasetName\_ API Name of the Child Object." If the child object has a long API name, the worksheet in the View Data workbook will truncate the name because an Excel worksheet name can only contain so many characters. If this is the case you'll see something like, "QueryData\_ APXTConga4\_\_Conga\_Custom\_Obj..." If there is an ellipsis (...) at the end of the worksheet name, it indicates that the dataset name has been truncated and will not work if referenced as such in the Word template. Instead, refer to the Conga Template Builder in order to retrieve the full dataset name.

**⚠** TableGroup or TableGroupStart are great alternatives to using nested SOQL queries. TableGroup and TableGroupStart are not compatible with the nested query method shown in this documentation.

## Using Nested Query Data in a Word Template

Use nested `{{TableStart}}` / `{{TableEnd}}` detail regions in the Word template to display the results of a nested query. Some important factors to keep in mind:

Everything between the outermost TableStart / TableEnd detail region repeats for every row of data in your outer query. Continuing our previous example, each unique Account (dataset name QueryData) displays for the out-most detail region (`{{TableStart:QueryData}}` / `{{TableEnd:QueryData}}`). Between the outer-most TableStart and TableEnd merge fields, build a detail region for each sub-query dataset (for example, QueryData\_Opportunities, QueryData\_Cases, etc.).

- These are the nested detail regions and repeats the rows of data from the corresponding sub-query, each of which is related to the outermost query (for example, parent Account).
- In other words, for the first Account, all Opportunities from the first Account will appear in the yellow table, Cases in the blue table, and Contacts in the green table.
- The TableHide field is a nice optional feature that prevents a table from merging (hide it) if the corresponding dataset returns no rows. For example, if one of the Accounts has Opportunities and Contacts but no Cases, the Cases table does not display. However, if the next Account does have Cases, the table appears for that iteration.

**NESTED DETAIL REGION EXAMPLE**

```

{{ Now \@ "MM/dd/yyyy hh:mm:ss" }}
{{TableStart:QueryData}}
Account Name: {{ ACCOUNT_NAME }}
UNID: {{UNID}}
Account ID: {{ACCOUNT_ID}}
Opportunities: {{TableHide:QueryData_Opportunities}}


| NAME                                                        | AMOUNT                                                                 |
|-------------------------------------------------------------|------------------------------------------------------------------------|
| {{TableStart:QueryData_Opportunities}} {{OPPORTUNITY_NAME}} | {{OPPORTUNITY_AMOUNT}}   Currency {{TableEnd:QueryData_Opportunities}} |
| TOTAL: {{AutoSum(Above)}}                                   |                                                                        |


Cases: {{TableHide:QueryData_Cases}}


| ID                                         | NUMBER                                           |
|--------------------------------------------|--------------------------------------------------|
| {{TableStart:QueryData_Cases}} {{CASE_ID}} | {{CASE_CASENUMBER}} {{TableEnd:QueryData_Cases}} |


Contacts: {{TableHide:QueryData_Contacts}}


| Last Name                                              | Email                                             |
|--------------------------------------------------------|---------------------------------------------------|
| {{TableStart:QueryData_Contacts}} {{CONTACT_LASTNAME}} | {{CONTACT_EMAIL}} {{TableEnd:QueryData_Contacts}} |


{{TableEnd:QueryData}}
    
```

*Annotations:*

- Everything between these outermost TableStart / TableEnd fields will repeat for each row of data in your outermost query (e.g. Account).
- Within the outer TableStart / TableEnd region, use the nested dataset names to display the rows of data from each sub-query.

The merged output file from the above example looks like this:

<b>NESTED DETAIL REGION EXAMPLE</b>	
09/17/2014 02:01:15	
<hr/>	
<b>Account Name: Aethna Home Products</b>	
UNID: fb556350-5c1a-4346-8b7a-c7d3a2816f1c	
Account ID: 001000000aMWrGAAW	
<b>Opportunities:</b>	
<b>NAME</b>	<b>AMOUNT</b>
Aethna Home Products - Propane Generator	\$15,000.00
Aethna Home Products - Diesel Engine	5500.00
Aethna Home Products - Big Deal	\$160,000.00
<b>TOTAL: \$175,500.00</b>	
<b>Cases:</b>	
<b>ID</b>	<b>NUMBER</b>
5000000007F4QQA0	00001026
<b>Contacts:</b>	
<b>Last Name</b>	<b>Email</b>
Akin	kakin@athenahome.com
Jones	bjones@email.com
<hr/>	
<b>Account Name: Burlington Textiles Corp of America</b>	
UNID: a97ad374-f2fc-4a05-befe-9a978a703683	
Account ID: 001000000aJalNAAS	
<b>Opportunities:</b>	
<b>NAME</b>	<b>AMOUNT</b>
Burlington Textiles Weaving Plant Generator	\$235,000.00
<b>TOTAL: \$235,000.00</b>	
<b>Cases:</b>	
<b>ID</b>	<b>NUMBER</b>
5000000007wBkIAAU	00001020
5000000007wBkHAAU	00001019
<b>Contacts:</b>	
<b>Last Name</b>	<b>Email</b>
Jones	bjones@gmail.com
Rogers	jrogers@burlington.com

You can add a Page Break just before the outermost TableEnd field (for example, TableEnd:QueryData) in order to display the results from each Account on a separate page.

With regard to writing a nested query for custom objects, often, it can be challenging to find the correct child relationship name to use in the sub-queries. We recommend using the [Force.com Workbench](https://workbench.developerforce.com) in order to find these child relationship names.

1. Navigate to <https://workbench.developerforce.com/login.php>
2. **Info tab** → **Standard & Custom Objects** → Choose your custom object → **Child Relationships** → Choose the child object you'd like to query → **relationshipName**  
The relationshipName is what you need to use in the FROM clause of the nested SOQL statement.

## Tablelinetrim

Use the tablelinetrim picture switch in a Word template to eliminate extraneous spaces from rich text fields.


Extraneous spaces are sometimes inserted when using Composer to merge rich text fields with bulleted lists within a table. Add the tablelinetrim picture switch to a merge field's syntax to eliminate extraneous spaces.

The tablelinetrim picture switch works specifically for rich text fields (that contain bulleted lists) in datasets with repeating data that require a table in a Word template.

The tablelinetrim picture switch is used in conjunction with [HTML function](#) in a merge field's syntax.


**Syntax:** \c tablelinetrim


**Merge Field Example:** `{{TableStart:QueryData}}{{HTML:ACCOUNT_BACKSLASH_RTF \c tablelinetrim}}{{TableEnd:QueryData}}`

 Tablelinetrim cannot be used in conjunction with [linetrim](#).

## Conditional TableHide Using an Operator in Word

Taking the previous Conditional TableHides one step further, you can evaluate whether a specific field value is equal to, not equal to, greater than, or less than a static value or the value of another field. If the evaluation is true, the table will be hidden. When specifying a detail dataset (report or query), the Conditional TableHide evaluates the first row of data in the specified dataset.

 If using many TableHide merge fields in a template, we recommend switching to [IF statements](#) for more consistent performance.

 Conditional TableHide does not support null values.

Syntax	<code>{{TableHide:Dataset:FieldName:Operator:Value}}</code>
--------	---

Values	<p>DatasetName is the name of the dataset from the Template Builder window.</p> <p>FieldName is the name of the field you want to evaluate, as it appears in the Template Builder.</p> <p>Operator specifies if the evaluation should be "equal to," "not equal to," "greater than," or "less than." The supported operators are:</p> <table border="1" data-bbox="264 488 587 824"> <tr> <td data-bbox="264 488 389 573">=</td> <td data-bbox="389 488 587 573">Equals</td> </tr> <tr> <td data-bbox="264 573 389 658">&lt;&gt; or !=</td> <td data-bbox="389 573 587 658">Not equal to</td> </tr> <tr> <td data-bbox="264 658 389 743">&gt;</td> <td data-bbox="389 658 587 743">Greater than</td> </tr> <tr> <td data-bbox="264 743 389 824">&lt;</td> <td data-bbox="389 743 587 824">Less than</td> </tr> </table> <p>Value is the value of the field that will force Composer® to hide the table.</p>	=	Equals	<> or !=	Not equal to	>	Greater than	<	Less than
=	Equals								
<> or !=	Not equal to								
>	Greater than								
<	Less than								
Notes	<p>Both Master and detail datasets are supported.</p> <p>Date fields should be formatted as month/day/year (e.g. May 13, 2014 would be 5/13/2014).</p>								
Usage	<pre data-bbox="264 1144 1114 1290"> {{TableHide:Pledges:Amount_Pledged:&lt;:200}} {{TableHide:Contracts:Start_Date:&gt;:5/5/2014}} {{TableHide:Master:CONTACT_DISPOSITION:=:Satisfied}} </pre>								

 This feature is not supported when using nested SOQL queries.

## Conditional TableHide Comparing Two Field Values in Word

You can determine whether to hide a table by evaluating the values from two separate fields.

Conga Composer can collect the value of the first field and compare it to the value of the second field, based on the operator used. For example, you could specify that a table of back-ordered product is hidden when the value of the first field (amount shipped) is equal to the value of the second field (amount ordered).

When specifying a detail dataset, the conditional TableHide evaluates the first row of data in the specified dataset. This feature is not supported when using nested SOQL queries. Also, we do not advise using more than 25 TableHide fields in a single template.

TableHide (comparing two values)									
Syntax	<code>{{TableHide:Dataset:FieldName1:Operator:FIELDVALUE(FieldName2)}}}</code>								
Values	<p>DatasetName is the name of the dataset from the Template Builder window.</p> <p>FieldName1 is the name of the field you want to evaluate, as it appears in the Template Builder.</p> <p>Operator specifies if the evaluation should be "equal to," "not equal to," "greater than," or "less than." The supported operators are:</p> <table border="1"> <tbody> <tr> <td>=</td> <td>Equals</td> </tr> <tr> <td>&lt;&gt; or !=</td> <td>Not equal to</td> </tr> <tr> <td>&gt;</td> <td>Greater than</td> </tr> <tr> <td>&lt;</td> <td>Less than</td> </tr> </tbody> </table> <p>FIELDVALUE(FieldName2) is the value of the field to be compared to the first field.</p>	=	Equals	<> or !=	Not equal to	>	Greater than	<	Less than
=	Equals								
<> or !=	Not equal to								
>	Greater than								
<	Less than								
Notes	<p>Both Master and detail datasets are supported.</p> <p>Both fields (FieldName1 and FieldName2) must come from the same dataset; either both from Master or both from the same detail dataset.</p> <p>Value must be a number or date field.</p> <p>Date fields should be formatted as numbers in the month/day/year format, such as 12/21/2015.</p> <p>Format a date field using this date appearance parameter format: <code>\@ MM/dd/yyyy</code>.</p>								
Usage	<p>This example hides the table if the Opportunity Amount field is less than the Amount Due field:</p> <pre> {{TableHide:Opptys:OPPORTUNITY_AMOUNT:&lt;:FIELDVALUE(AMOUNT_DUE)}}                     </pre> <p>This example hides the table if the values for Account Status and Account Type are not the same.</p> <pre> {{TableHide:Master:ACCOUNT_STATUS:&lt;&gt;:FIELDVALUE(ACCOUNT_TYPE)}}                     </pre>								

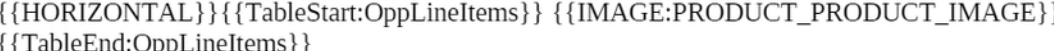


## Arranging Images Horizontally in Word

Composer can arrange images horizontally into a Word template by using the HORIZONTAL tag as mentioned in the following example:

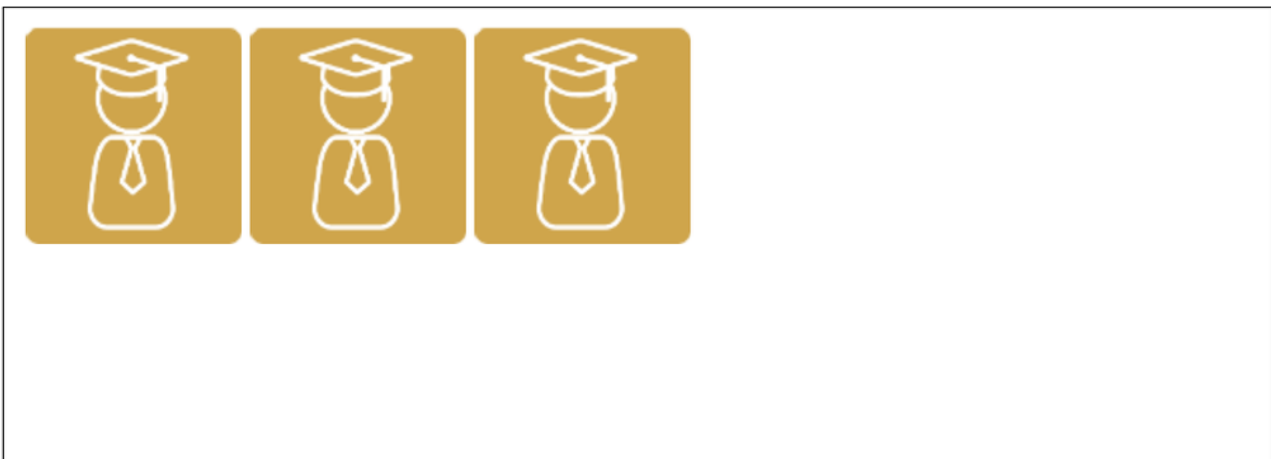
### Template Example:

```
{{HORIZONTAL}}

|                                                                                    |
|------------------------------------------------------------------------------------|
|  |
|------------------------------------------------------------------------------------|

{{TableEnd:OppLineItems}}
```

### Output Example:



## PowerPoint Templates

You can merge data from Salesforce into formatted PowerPoint slides that can contain images, text, graphs, charts, and tables.

- [PowerPoint Template Basics](#)
- [Advanced PowerPoint Templates](#)

## PowerPoint Template Basics

The following sections contain basic information for creating PowerPoint templates.

- [PowerPoint Template Guide](#)
- [Designing Microsoft PowerPoint \(PPTX\) Merge Templates](#)
- [Add a Merge Field to a PowerPoint Template](#)

- [Display the Merge Date in PowerPoint](#)
- [List of Date Value Formatting Parameters for PowerPoint](#)
- [Formatting Numeric Values in PowerPoint](#)
- [Merging Detail Data in PowerPoint](#)
- [Add a Table to a PowerPoint Template](#)
- [Add Multiple Slides using a Detail Region in PowerPoint](#)
- [Unsupported PowerPoint Features](#)

## PowerPoint Template Guide

### Learn PowerPoint Template Basics

- [Designing Microsoft PowerPoint \(PPTX\) Merge Templates](#)

### Create PowerPoint Templates

- [Add a Merge Field to a PowerPoint Template](#)
- [Special Merge Fields in PowerPoint](#)
- [HTML Fields in PowerPoint](#)
- [Hyperlink Fields in PowerPoint](#)
- [Image Fields in PowerPoint](#)
- [Picture Placeholders in PowerPoint](#)
- [Image Merge Fields in PowerPoint](#)
- [Row Number Field in PowerPoint](#)
- [SlideHide Fields in PowerPoint](#)
- [Display Table Elements as a List in PowerPoint](#)
- [Grouping Detail Data with TableGroupStart in PowerPoint](#)
- [Display the Merge Date in PowerPoint](#)
- [Add Charts to PowerPoint Templates](#)
- [List of Date Values Formatting Parameter for PowerPoint](#)
- [Formatting Numeric Values in PowerPoint](#)
- [Merging Detail Data in PowerPoint](#)
- [Add a Table to a PowerPoint Template](#)
- [Add a Group Detail Region on a PowerPoint Template](#)
- [Add Multiple Slides using a Detail Region in PowerPoint](#)
- [Individual values from ReportData or QueryData in PowerPoint](#)
- [Referring to the Nth Row in a ReportData or QueryData set in PowerPoint](#)

## Troubleshoot PowerPoint Templates

- [Unsupported PowerPoint Features](#)
- [Product Data Limits](#)


## Designing Microsoft PowerPoint (PPTX) Merge Templates

Conga Composer can merge data into a Microsoft PowerPoint PPTX template.

### Before You Begin


These instructions assume the following:

- You know how to build a Conga Composer button. See the [Product Overview Video](#).
- You know how to use the Conga Template Builder. See [The Template Builder](#).

 Conga Composer can merge data into a Microsoft PowerPoint PPTX template. (PowerPoint templates work only with Conga Composer not Conga Mail Merge.)

### What Exactly is a PowerPoint PPTX Template?

It is a PowerPoint file (.pptx), created in PowerPoint 2007 or PowerPoint 2010, which contains one or more merge fields.

 Conga Composer release 8 does not support PowerPoint 2003 (.ppt) templates. You must use PowerPoint 2007 or later (.pptx).

The Template Builder is an essential tool in the development of Conga Composer solutions. The Template Builder exposes the data retrieved by Conga Composer from the Master Object (and from any Salesforce Reports or Conga Queries referenced in the Conga Composer URL). You can then use these fields to build templates.

To learn more on how to use the Conga Template Builder, see [About Data in the Template Builder](#).

### Add a Merge Field to a PowerPoint Template

To add a merge field to a PowerPoint template:

1. In PowerPoint, create or open a presentation to use as your template.
2. Open the Template Builder window. Locate the field you'd like to use and copy it to the clipboard.

3. In PowerPoint, navigate to the location on the template where you want to add the merge field and click **Insert** → **Text Box**. Best practice suggests putting each merge field in its own text box. You may also use Shapes (Rectangles, Circles, etc.) WordArt, or Click-Here Placeholders as merge fields.
4. Name the field by following this convention:  
`{{FieldName}}` where FieldName is the field name that you copied to the clipboard from Template Builder.
5. Position the merge field.

This field naming convention works for fields from the Master, Org, or User datasets. To reference fields in a ReportData or QueryData dataset, you may use the “format” naming convention or a Detail Region. Don’t include new line characters within a merge field.

## Display the Merge Date in PowerPoint

You can insert the current date into the merged document by using a merge field called `{{Date.Today}}` or just `{{Today}}`.

To create a field to display the date on which the merged document is created:

1. In your template, create a merge field named `{{Today}}`.
2. Optionally, apply an [appearance parameter](#) to format the date, as below:

A rectangular box with a thin black border containing the text `{{Today \@ MMMM d, yyyy}}`. The text is in a monospace font and is centered within the box.

## List of Date Value Formatting Parameters for PowerPoint

The date formatting parameter begins with `\@`, followed by any combination of the following parameters. Please see the examples following this table.

PARAMETER	DESCRIPTION
Month (M)	<p>The letter "M" must be uppercase to distinguish months from minutes.</p> <p>M Displays the month as a number without a leading 0 (zero) for single-digit months. For example, February is "2".</p> <p>MM Displays the month as a number with a leading 0 (zero) for single-digit months. For example, February is "02".</p> <p>MMM Displays the month as a three-letter abbreviation (e.g. Feb).</p> <p>MMMM Displays the month as its full name (e.g. February).</p>
Day (d)	<p>Displays the day of the month or the day of the week. The letter "d" can be either uppercase or lowercase.</p> <p>d Displays the day of the week or month as a number without a leading 0 (zero) for single-digit days. For example, the sixth day of the month is displayed as "6".</p> <p>dd Displays the day of the week or month as a number with a leading 0 (zero) for single-digit days. For example, the sixth day of the month is displayed as "06".</p> <p>ddd Displays the day of the week or month as a three-letter abbreviation. For example, Tuesday is displayed as "Tue".</p> <p>dddd Displays the day of the week as its full name.</p>
Year (y)	<p>Displays the year as two or four digits. The letter "y" can be either uppercase or lowercase.</p> <p>yy Displays the year as two digits with a leading 0 (zero) for years 01 through 09. For example, <code>{{DATE \@ "yy"}}</code> displays 1999 as "99" or 2006 as "06."</p> <p>yyyy Displays the year as four digits. For example, <code>{{DATE \@ "yyyy"}}</code> displays 2015 as "2015."</p>

PARAMETER	DESCRIPTION
Hours (h)	<p>A lowercase "h" bases time on the 12-hour clock. An uppercase "H" bases time on the 24-hour, or military, clock.</p> <p>h or H Displays the hour without a leading 0 (zero) for single-digit hours. For example, <code>{{TIME \@ "h"}}</code> displays 9 P.M. as "9" while <code>{{TIME \@ "H"}}</code> displays 9 P.M. as "21".</p> <p>hh or HH Displays the hour with a leading 0 (zero) for single-digit hours. For example, <code>{{TIME \@ "hh"}}</code> displays 9 P.M. as "09" while <code>{{TIME \@ "HH"}}</code> displays 9 P.M. as "21".</p>
Minutes (m)	<p>The letter "m" must be lowercase to distinguish minutes from months.</p> <p>m Displays minutes without a leading 0 (zero) for single-digit minutes. For example, <code>{{TIME \@ "m"}}</code> displays "2".</p> <p>mm Displays minutes with a leading 0 (zero) for single-digit minutes. For example, <code>{{TIME \@ "mm"}}</code> displays "02".</p>
A.M. and P.M. (AM/PM)	<p>am/pm Displays A.M. and P.M.</p> <p>AM/PM Displays A.M. and P.M. as uppercase. For example, <code>{{TIME \@ h AM/PM}}</code> and <code>{{TIME \@ h am/pm}}</code> display "9 AM" or "5 PM".</p>

## Formatting Numeric Values in PowerPoint

The number formatting parameter begins with `\#`, followed by any combination of the following parameters. When Conga Composer populates a numeric value, the digit grouping symbol (the "thousands separator") is defined according to your Locale settings (as defined in Salesforce's Setup > Personal Information) or according to the Culture or CurrencyCulture Conga Composer parameter.

Despite the cultural settings that govern the output from Conga Composer, within an appearance parameter, you must use a comma (",") for the digit grouping symbol (for thousands) and a period (".") as the decimal symbol.

## List of Parameters

PARAMETER	DESCRIPTION
0 (zero)	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a 0 (zero).
#	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a space.
, (digit grouping symbol)	Separates a series of three digits.  Note: The digit grouping symbol is always entered as a comma into the formatting code, regardless of the country's cultural preference. Conga Composer will format the output according the cultural preference.
%, \$, *, and so on	Includes the specified character in the result.
"positive; negative"	Specifies different number formats for positive and negative results.
Currency	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol ("thousands separator") and two decimal places  Example: <code>{{OPPORTUNITY_AMOUNT \# Currency}}</code>
Currency0	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol ("thousands separator") and zero decimal places  Example: <code>{{OPPORTUNITY_AMOUNT \# Currency0}}</code>  In Conga Composer solutions, you can override the Salesforce Locale with the "CurrencyCulture" or "Culture" parameter.

PARAMETER	DESCRIPTION
Number	<p>Formats the value according to the Salesforce Locale, excluding a currency symbol, but including a digit grouping symbol (“thousands separator”) and two decimal places</p> <p>Example:</p> <pre>{{OPPORTUNITY_AMOUNT \# Number}}</pre>
Number0	<p>Formats the value according to the Salesforce Locale, excluding a currency symbol, but including a digit grouping symbol (“thousands separator”) and zero decimal places</p> <p>Example:</p> <pre>{{OPPORTUNITY_AMOUNT \# Number0}}</pre>
Percent	<p>Formats the value according to the Salesforce Locale, by multiplying the value by 100 and including a digit grouping symbol (“thousands separator”), two decimal places and a percent sign.</p> <p>Example:</p> <pre>{{OPPORTUNITY_PROBABILITY \# Percent}}</pre>

In Conga Composer solutions, you can override the Salesforce Locale with the CurrencyCulture or Culture parameter.

## Merging Detail Data in PowerPoint

Merge fields, along with any other text, placed within a Detail Region will automatically repeat for each row of detail data.

Conga Composer has the unique ability to use data from a Salesforce Report or Conga Query as detail data (for example, invoice line items for each invoice). Because reports and queries can retrieve multiple rows of data, Conga Composer uses detail regions to dynamically grow an area on a slide to accommodate an unlimited number of detail rows.

Detail Regions let you identify a portion of your template that can dynamically grow. Merge fields, along with any other text, placed within a Detail Region will automatically repeat for each row of detail data.

In PPTX templates, you can define detail regions in three ways:



- [Table-based detail region](#)
- [Slide-based detail region](#)
- [Group-based detail region](#)

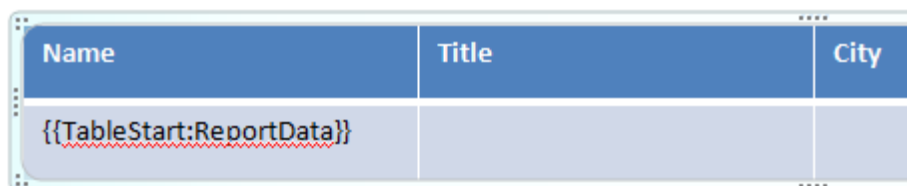
## Add a Table to a PowerPoint Template

How to add a table to a PowerPoint template.

For arranging data in columns and rows, you'll find Table-based detail regions particularly useful. You create a table including column headers and a row to hold your merge fields. Conga will automatically expand the table to accommodate the number of rows in your dataset.

To add a table to a PowerPoint Template:

1. In your PPTX template, create a table with at least two rows (one for column headers and the other for merge fields) and as many columns as necessary to accommodate your data.
2. In your Template Builder window, locate a field from a dataset that may contain multiple rows of data.
3. Copy it to the clipboard.
4. Switch to PowerPoint and move to the first cell in the row that will contain your merge fields.



Name	Title	City
{{TableStart:ReportData}}		

5. To mark the beginning of the Detail Region, create a merge field with the name `{{TableStart:DatasetName}}` where DatasetName corresponds to the name of the dataset from the Template Builder window.
6. To mark the end of the Detail Region, in the last cell of the row, create a merge field with the name `{{TableEnd:DatasetName}}` where DatasetName corresponds to the name of the dataset from the Template Builder window.



Name	Title	State
{{TableStart:ReportData}}		{{TableEnd:ReportData}}

7. Between the TableStart and TableEnd fields, place merge fields that correspond to the fields of your report data.

These merge fields will be populated with data from the first row of the detail data, then the whole region will be repeated, the new fields will be populated with data from the

second row, and so on.

This table:

Name	Title	State
{{TableStart:ReportData}}{{Full_Name}}	{{Title}}	{{Mailing_State/Province}}{{TableEnd:ReportData}}

When merged, shows this data:

Name	Title	State
Jack Rogers	VP, Facilities	NC
Mary Ann Powell	VP Marketing	NC
Quilton Froil	COO	NC

A slide may only contain one set of TableStart / TableEnd fields. Use “simple” field names within Detail Regions. Do not include DatasetName to refer to the dataset since the dataset is identified by the TableStart field. Change the color of the TableStart and TableEnd fields to help visually distinguish them from other merge fields. We suggest red.

Please follow these guidelines when creating a table-based Detail Region:

- A slide may contain only one detail region.
- If you have more rows of data than will fit on a slide, new slides will be generated to accommodate your data.
- PowerPoint does not have the ability to perform calculations (like =SUM(ABOVE)), so please use a formula field in Salesforce to perform calculations and merge the result into the table.
- PowerPoint does not support images within cells. If, however, you include an Image: merge field, the Conga will apply the image to the cell background. For more information, see [Image Merge Fields](#).
- Do not use the “formal” field naming convention within tables, i.e., do not reference the DatasetName.

## Add Multiple Slides using a Detail Region in PowerPoint

How to add Multiple Slides using a Detail Region.

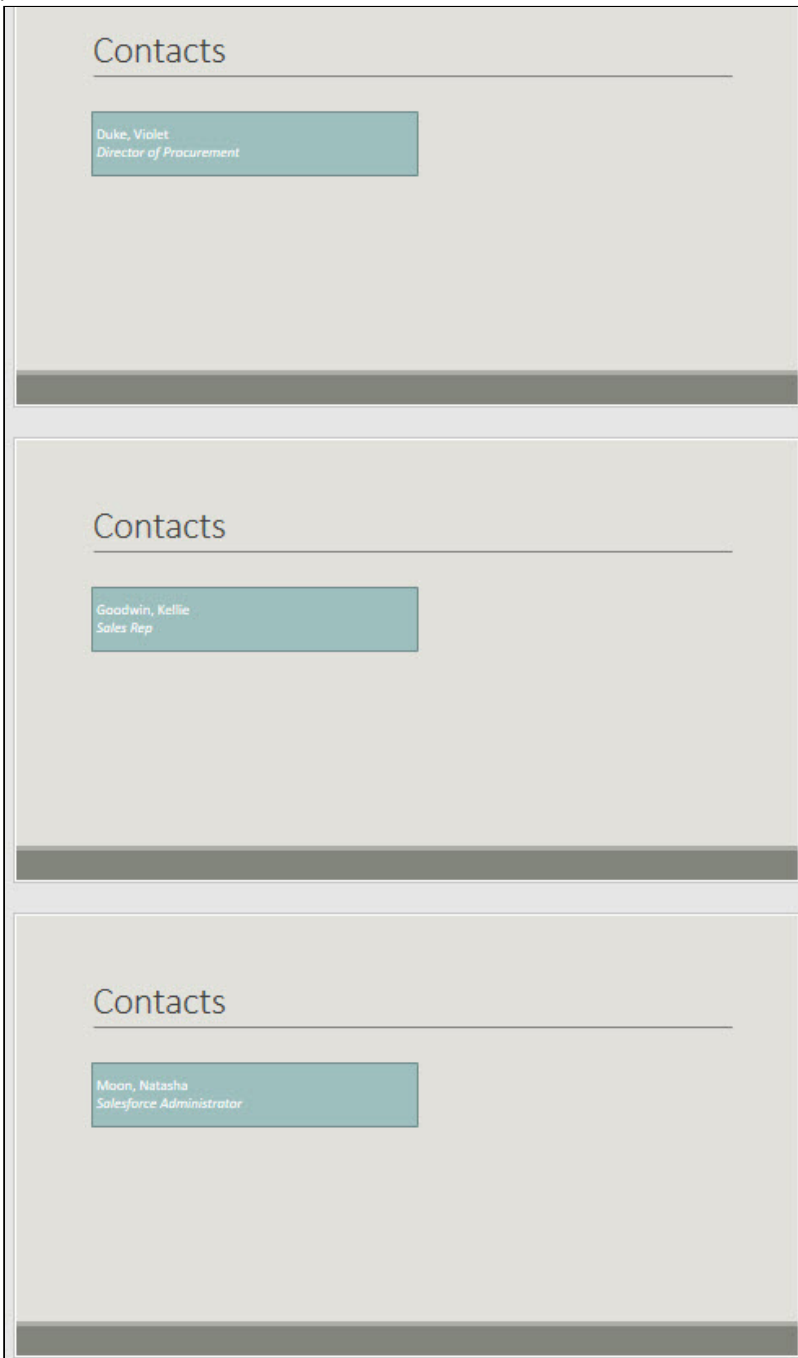
To repeat the entire slide for each row of detail data, do the following:

1. Copy and paste merge fields from the detail dataset into the slide using the Template Builder. These can be entered into Text Boxes, Shapes, or Picture Frames.

2. Create a separate Text Box on the slide and enter `{{SlideRepeat:DatasetName}}`, where DatasetName refers to the name of the detail dataset you want repeated on each slide. The template might look like this:



The output from the slide-based detail region above would look like this (notice one slide per row of detail data):



**⚠** If the detail dataset has no data, the Slide with SlideRepeat is removed. SlideRepeat will generate slides from the first 50 rows of data in the dataset.

## Unsupported PowerPoint Features

The following features in PowerPoint are not supported:

### General

- It is recommended to close the PowerPoint template before “local” merging.
  - Leaving the template open in PowerPoint during a “local” merge operation can sometimes produce unusual results in the output file.
- RichText and Image fields when using dot notation or formal style merge fields:
- Field types do not produce the desired result if the style of merge field is like the following example: `{{HTML:QueryName.FieldName}}` or `{{Image:QueryName.FieldName}}`
  - Special fields referenced here: [Individual values from ReportData or QueryData in PowerPoint](#)

### Tables

- PowerPoint tables do not accept an image in-line with text, however Conga can insert an image as a cell background.

### Text Boxes and Shapes

- SmartArt (PowerPoint’s tool to create charts and process flows) is not currently available in a programmatic fashion.
- When merging data that contains new line characters, it is recommended to use a single font for the entire text box or shape.

### IF Statements

- If statements are only functional in Microsoft Word templates when using Conga Composer.
  - PowerPoint does not have this functionality. A workaround is to create an IF formula field in Salesforce.
- PowerPoint cannot perform calculations such as like `=SUM(ABOVE)`.
  - A workaround is to use a formula field in Salesforce to perform calculations and merge the result into the table.

### SOQL Queries

- Nested SOQL Queries used to gather data are not supported in PowerPoint templates

## Advanced PowerPoint Templates

The following sections contain information on advanced template-building techniques and tricks.

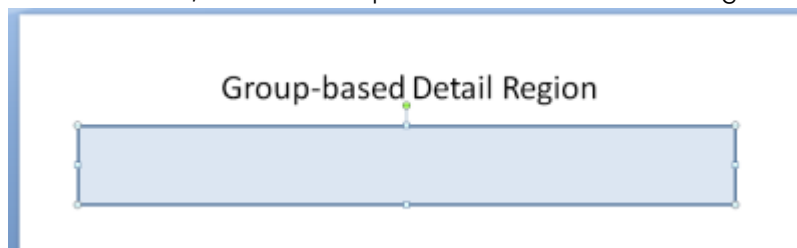
- [Add a Group Detail Region on a PowerPoint Template](#)
- [Grouping Detail Data with TableGroupStart in PowerPoint](#)
- [Add Charts to PowerPoint Templates](#)
- [Special Merge Fields in PowerPoint](#)
- [HTML Fields in PowerPoint](#)
- [Hyperlink Fields in PowerPoint](#)
- [Image Fields in PowerPoint](#)
- [Image Merge Fields in PowerPoint](#)
- [Picture Placeholders in PowerPoint](#)
- [Row Number Field in PowerPoint](#)
- [SlideHide Fields in PowerPoint](#)
- [Display Table Elements as a List in PowerPoint](#)
- [Individual values from ReportData or QueryData in PowerPoint](#)
- [Referring to the Nth Row in a ReportData or QueryData set in PowerPoint](#)

### Add a Group Detail Region on a PowerPoint Template

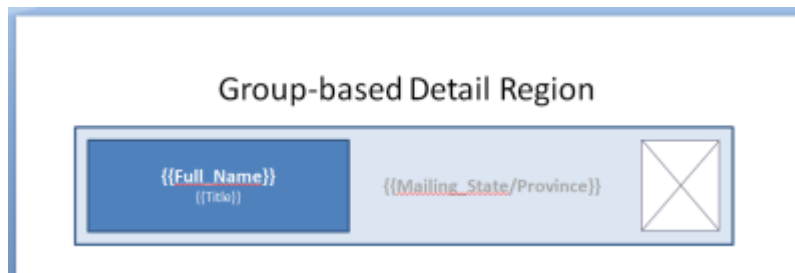
You can group text boxes, shapes, and picture frames within a shape and create a group-based detail region. Conga Composer repeats the grouped-shape for each row of detail data.

To create a group detail region:

1. In Template Builder, locate a field from a Report Data dataset that contains multiple rows of data.
2. Copy the field to the clipboard.
3. In PowerPoint, insert a shape to act as the boundary of the detail region.

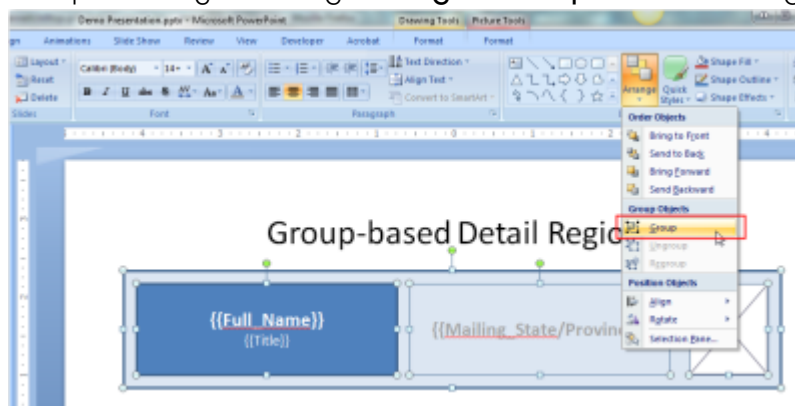


4. Within the boundary shape, insert Text Boxes, Shapes or Picture Frames to serve as the merge fields for each row of detail data.

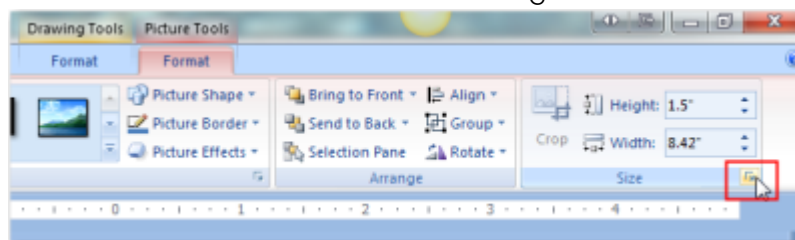


You may also refer to Master Fields

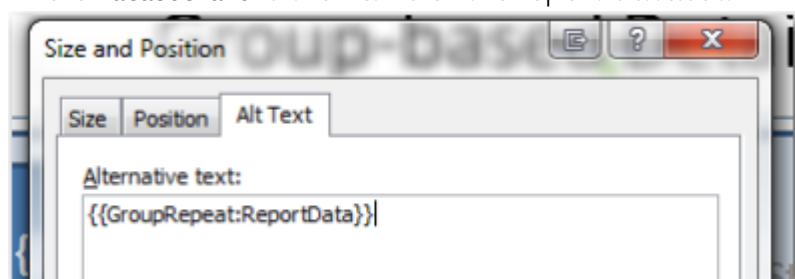
5. Select all of the individual elements plus the boundary element.
6. Group them by choosing **Arrange** → **Group** in the Drawing group.



7. In PowerPoint, open the Size & Position dialog box. Then right-click the object and click **Size & Position**. Or From the Ribbon: Select the grouped object → select the **Format** ribbon → click the lower-right corner of the Size group.

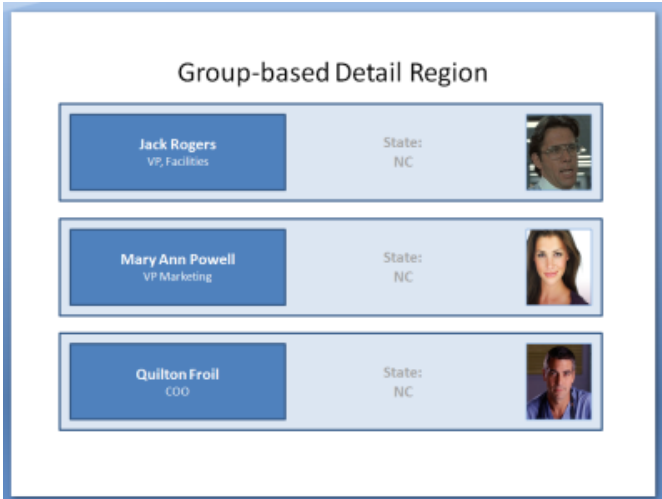


8. Click the **Alt Text** tab on the Size & Position dialog box.
9. On the Size & Position dialog box, type `{{GroupRepeat:DatasetName}}`, where DatasetName is the name of the report dataset.



10. Save your work.

When merged, the output of the Group-based Detail Region from Conga appears similar to this:



To learn more about merging images, please see [Image Merge Fields](#). If the detail data set has no data, the Group with GroupRepeat is removed. Consider combining this feature with [SlideHide](#), which would remove the entire slide under certain conditions.

GroupRepeat will generate slides for the first 50 rows of data in the data set. A slide may only contain one GroupRepeat field

Use "simple" field names within Detail Regions. Do not include DatasetName to refer to the dataset since the dataset is identified by the Grouped object.

## Grouping Detail Data with TableGroupStart in PowerPoint

You can group detail-data in PowerPoint PPTX templates, much as you would in a Word template, by including a TableGroupStart: merge field on the first row of a detail table. Each resulting group is placed on its own slide.

The syntax for a PPTX TableGroupStart field is `{{TableGroupStart:DatasetName:FieldName}}` followed by static text and standard merge fields from the dataset to display for each grouping.

The `{{TableGroupStart}}` field must be placed in the first row of the table, and it must be followed by a standard `{{TableStart}}` and `{{TableEnd}}` detail region.

This field:



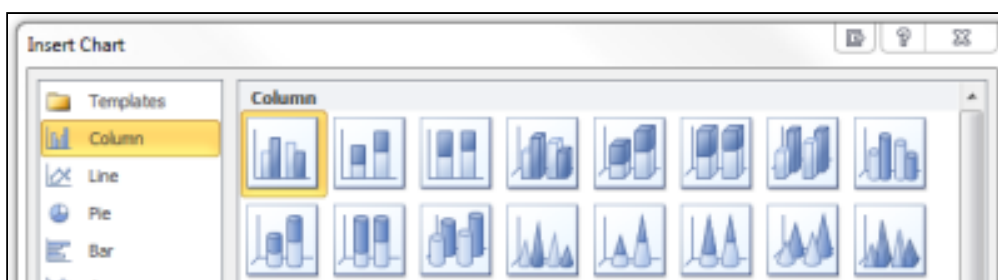
Results in:



**⚠** TableGroup is supported only in Conga Composer, not Conga Mail Merge. Only one TableGroup or TableGroupStart may be used per dataset. You may duplicate a dataset in order to use an additional TableGroup or TableGroupStart; ensure you give this dataset a unique alias.

## Add Charts to PowerPoint Templates

PowerPoint has a rich array of charting capabilities and Conga Composer lets you generate charts using Salesforce data. Your Salesforce data should be aggregated on some numeric field. Because Salesforce SOQL Queries (and Conga Queries) make it easy to aggregate on numeric fields, they're ideal for retrieving data compatible with PowerPoint charts.



To create a chart within a PowerPoint template:

1. Create a Conga Query (SOQL) that pulls two fields from Salesforce: a category and an aggregated numeric field.
  - a. To gather data for a bar chart that depicts the total of the Amount of each Opportunity, grouped by Lead Source:

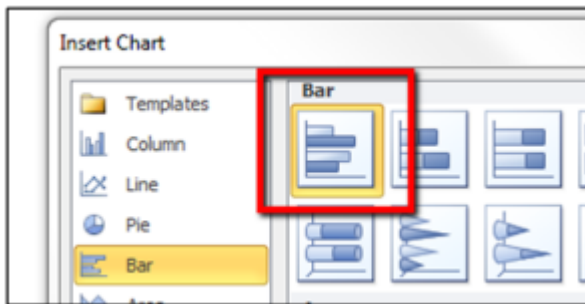
```
SELECT LeadSource, SUM(Amount) FROM Opportunity GROUP BY LeadSource OR SELECT
Status, COUNT(Id) FROM Case WHERE AccountId = '{pv0}' GROUP BY Status ORDER BY
Status ASC
```

After the Conga Query is saved and included in the Conga Composer button, it appears on the Template Builder window as QueryData. In this example, the first column will be used for the column labels (category) and the numeric values will be used for the bar height (series).

	A	B	C
	OPPORTUNITY_GROUP_BY LEADSOURCE_LEADSOUR	OPPORTUNITY_G ROUP_BY_LEADS OURCE_EXPRO	
1	CE		
2		1450000	
3	Web	1940000	
4	Phone Inquiry	100000	
5	Purchased List	15000	
6	Employee Referral	350000	
7	External Referral	4860000	
8	Partner	7740000	
9	Public Relations	140000	
10	Trade Show	220000	
11	Word of mouth	120000	
12			

Use the &DisableRowNums parameter in your button or link URL to remove the row numbers from your query dataset. Otherwise, the row numbers will appear in the chart.

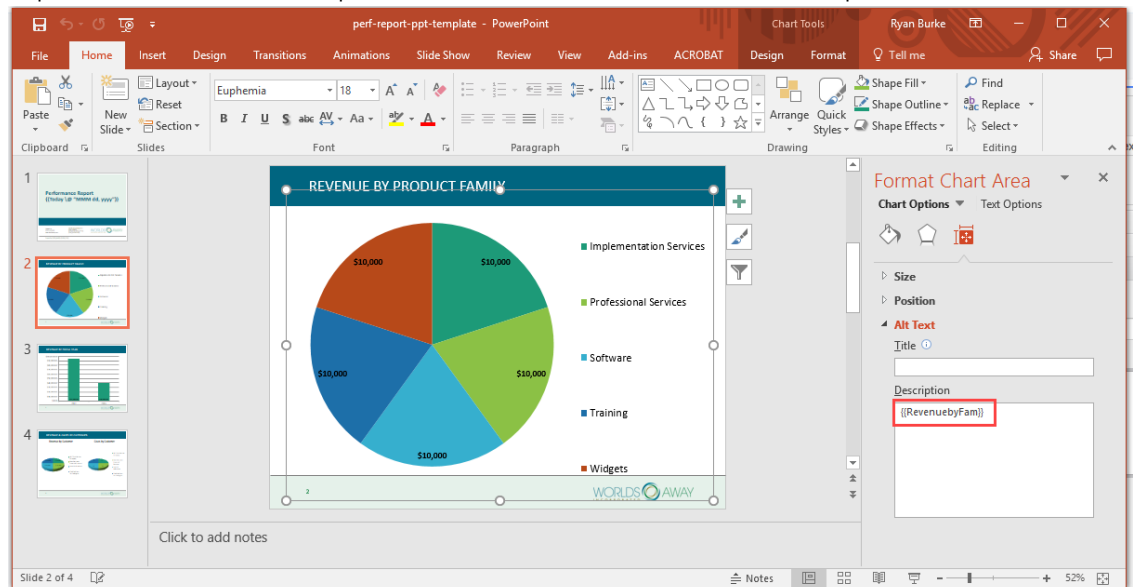
2. In your PPTX template, select the slide to contain the chart.
3. Click **Insert** → **Chart**.
4. Select a chart style.



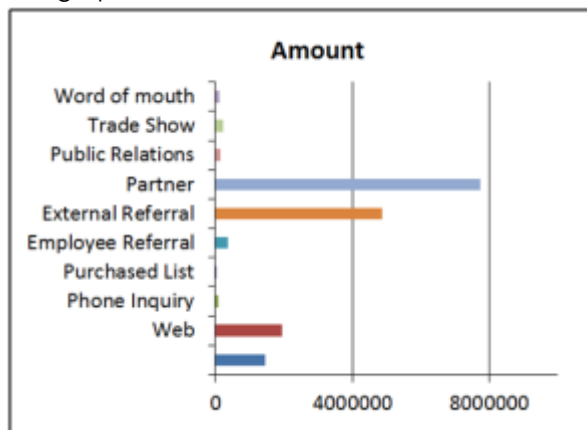
PowerPoint opens an Excel workbook with Sample data.

5. (Optional) Format the chart.
6. Link the chart to your Conga Query:
  - a. Click on the chart. The Format Chart Area panel will appear. Click **Chart Options**, then click **Size & Properties**.

- b. Expand the Alt Text drop-down to show the Title and Description fields.



- c. On the **Alt Text** tab in the Description field, enter the DatasetName that contains the data to chart using the following convention: `{{DatasetName}}`. In the image above, we used `{{RevenuebyFam}}`. The data found on the QueryData sheet will be inserted into the Chart because we referenced `{{QueryData}}`. The merge process renders the chart with the data from the dataset referenced.



The chart can be formatted (in the template) in a variety of ways: the axis values, chart legend, chart title, etc.

## Special Merge Fields in PowerPoint

Conga Composer offers additional merge fields for even more flexibility in your merge templates.

- **HTML:** fields let you render text that includes HTML formatting codes as rich text
- **Hyperlink:** fields let you generate clickable links

- **Image:** fields let you dynamically include graphic files into your merged output, like photos or signature images
- **Row Number Field:** The Row Number field numbers each line item in a detail dataset and is included by default in every report or query dataset. The field name is ROW\_NUMBER.
- **SlideHide:** fields let you hide slides based on criteria in your data
- **TableList:** fields let you display a column of detail-data as a list
- **Today's Date:** The Date.Today field inserts today's date, and may be formatted with an appearance parameter

## HTML Fields in PowerPoint

Conga Composer can render Salesforce fields that contain HTML formatting codes as PowerPoint rich-text. This is particularly useful for long text fields that contain the HTML "<br />" (line break) code and for Salesforce Rich Text fields. With this kind of merge field, Conga renders the "<br />" tags as line-breaks.

Type HTML: to the left of the field name. Include the colon, but no spaces. For example:

```

{{HTML:Account_Description}}

```

Use "simple" field names with this feature, i.e., do not refer to DatasetName. HTML fields work only with Master data and SlideRepeat Detail Regions Images stored directly in Rich Text fields are not supported. To retrieve the rich text formatting, any rich text fields that are part of detail data need to be retrieved via a query rather than a report.

## Hyperlink Fields in PowerPoint

Conga Composer can render Salesforce fields that contain URLs as clickable hyperlinks. The field from Salesforce must contain an "absolute" (complete) URL. You may optionally specify the text to display in PowerPoint instead of the URL value.

Create a merge field for your hyperlink field, using the following syntax:

```

{{HYPERLINK:FieldName:DisplayText}}

```

- **HYPERLINK** is the hyperlink field prefix.
- **FieldName** is the name of your URL field as it appears in the Template Builder.
- **DisplayText** is an optional suffix to provide friendly display text rather than the entire URL.

When adding a Hyperlink field to a template, format the field in blue, underlined text to provide a visual cue that the text is a link.

**Example:** `{{HYPERLINK:Company_Website:Visit our home page.}}`

After merging, appears as: [Visit our home page.](#)

Hyperlink fields set the value of the entire text box (or shape) and as such, should not be part of a paragraph. Use “simple” field names with this feature—do not refer to DatasetName. Hyperlink fields work only with Master data and SlideRepeat Detail Regions.

## Image Fields in PowerPoint

Merge images in PowerPoint with Conga Composer.

Conga Composer can merge images, in addition to typical field values. A common use of this is to include the logo of an Account or a job-site photo attached to an Opportunity. Image fields can be created either using picture placeholders or by merge fields. Each has advantages.

To merge images, the following pre-requisites must be met regarding the image files:

- File type must be JPG or PNG
- File must be saved with the desired height and width (recommended)
- File must be stored in Salesforce (or on a web server) and accessible through a URL
- If stored in Salesforce, the image file must be stored in the Documents Tab or as an attachment to a record

When you store an image in Salesforce, either on the Documents Tab or as an attachment to a record, Salesforce will assign an ID to the image record. Conga Composer can use the URL of the image record to retrieve the image and include it in your document.

To store an image in Salesforce to hold the URL of the image:

1. Create a text field in Salesforce to hold the URL of the image.
2. Store an image attachment on the Documents tab or as an attachment to a record.
3. Click the **View link** that accompanies the attachment.
4. Copy the complete URL of the image record to the clipboard.
5. Paste the URL into a record that has the new field created in Step 1 and save the record.

In the event that the field storing the URL does not contain a URL of an image and is therefore an empty field, the user may see an error in the merged output file. To overcome this, the URL field must contain a value.

If the field will sometimes be empty, one way to overcome the error seen in the output file will be to add an IF statement to a Salesforce formula field to make sure that if that URL field is empty, populate it with a generic URL of another image (for example, an empty image).

Example merge field syntax: `{{IMAGE:Account_Image_URL:100:75}}`

**⚠ The 100 and 75 in the merge field example above refer to the designated height and width of the image.**

For more information on merge field syntax for images in PowerPoint templates, see [Image Merge Fields in PowerPoint](#).

## Image Merge Fields in PowerPoint

You can insert an IMAGE merge field instead of using a picture placeholder. Image merge fields have the advantage of dynamically sizing an image while maintaining its proportions and working within table cells.

Image merge fields use the following syntax: `{{IMAGE:FieldName:[Height]:[Width]}}`

### Merge Field Syntax Components

- IMAGE: indicates the field type.
- FieldName: is the name of the Salesforce field.
- Height and Width are optional parameters to define the size of the image in pixels. If only one of the size parameters is specified, the resulting image will maintain its original proportions.

To specify the width and scale the height proportionately, leave the height value blank and include two colons (::) after the FieldName and before the pixel value of the width.

Image with a fixed height of 50 pixels.	<code>{{IMAGE:OPPORTUNITY_PHOTO_URL:50}}</code>
Square image of 50 pixels high and wide.	<code>{{IMAGE:OPPORTUNITY_PHOTO_URL:50:50}}</code>

**⚠ When image fields are used in table cells, the image is applied to the cell background.**

## Picture Placeholders in PowerPoint

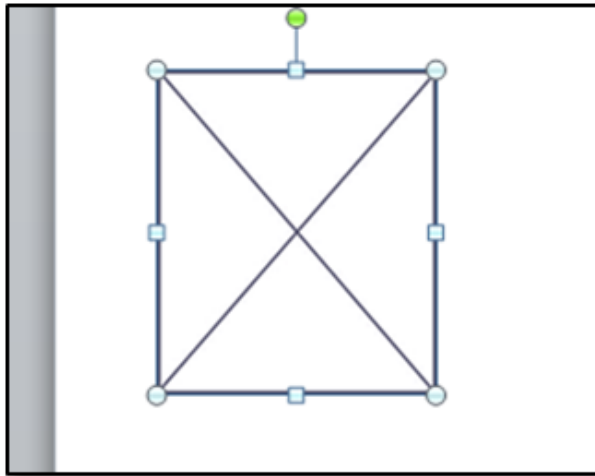
A picture placeholder lets you graphically designate the location for an image and has the added benefit of letting you pre-determine the size of the image (height and width).

To use a picture placeholder:

1. On the desired slide, choose **Insert** → **Picture**.

2. Select an image from your computer to stand-in for the merged image. (Any image will do.)

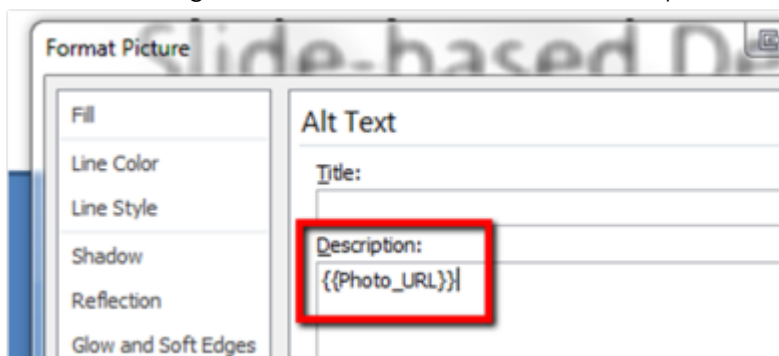
- Placeholder Image:



⚠ We suggest creating or downloading a “placeholder” image to use as the stand-in for the merged image.

3. Using your mouse, size the image to the desired dimensions to determine which field will contain the image’s URL, add the merge field to the AltText property (located on Size & Position).

- Right-click the image and choose **Size & Position**. Alternate method: Click **Format** (ribbon) → **Size** (group) → **Size & Position** (lower right corner)
- Enter the merge field on the Alt Text tab (Description field).



⚠ Since the picture placeholder determines the dimensions of the final output, the merged image may be distorted if it isn’t designed for the dimensions of the placeholder.

Picture placeholders won’t work in Table cells. The output from this picture placeholder looks like this:



## Row Number Field in PowerPoint

The Row\_Number field numbers each line item in a detail dataset and is included by default in every report or query dataset. The field name is ROW\_NUMBER.

Insert this field into any detail region in the template to number the results in the merged output file. In the below example, the `{{ROW_NUMBER}}` merge field is placed in the far left column:

Item Number	Product Name	Quantity	Total Price
<code>{{TableStart:OppLineItems}}{{ROW_NUMBER}}</code>	<code>{{PRODUCT_NAME}}</code>	<code>{{OPPORTUNITY_LINEITEM_QUANTITY}}</code>	<code>{{OPPORTUNITY_LINEITEM_TOTAL PRICE}}{{TableEnd:OppLineItems}}</code>

The merged output file looks like this:

Item Number	Product Name	Quantity	Total Price
1	GenWatt Diesel 10kW	1	5000
2	GenWatt Propane 100kW	2	30000
3	GenWatt Diesel 200kW	5	125000
4	GenWatt Gasoline 300kW	3	105000

## SlideHide Fields in PowerPoint

Conga Composer lets you dynamically hide (remove) particular slides based on data from Salesforce. You can hide slides based on whether the report returns no corresponding data rows or if a field in the Master set equals a value you specify.



## Report-based (or Query-based) SlideHide

In report-based SlideHide, you can hide a slide that has no data rows on the corresponding ReportData sheet (or QueryData sheet) by including a special merge field, SlideHide:DatasetName, anywhere on the slide.

The syntax for Report/Query-based SlideHide is: `{{SlideHide:DatasetName}}` where DatasetName is the name of the dataset from the Template Builder window.

If the ReportData2 report has no records, the slide containing the field is removed from the output:



## Field-Based SlideHide

In Field-based SlideHide, you can hide a slide if a field in the Master set of data equals a particular value. The syntax for this variation is: `{{SlideHide:FieldName:Value}}` where FieldName is the merge field name in the Master data set.

Value is the value against which the field is compared. If the field equals the value, the slide is removed from the output.

If the Opportunity\_IsClosed field is True, the slide is removed from the output.

`{{SlideHide:OPPORTUNITY_ISCLOSED:True}}`

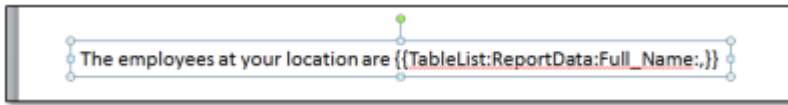
## Display Table Elements as a List in PowerPoint

Typically, a detail dataset (a report or a query) is displayed in a Detail Region. However, using a TableList field, you can display a column from the Report or Query as a list of values, separated by a punctuation mark you specify.

For example, suppose you're using a report to gather a list of Contacts at a given Account:

Salutation	First Name	Last Name	Title	Phone	Email
Ms.	Rose	Gonzalez	SVP, Procurement	(512) 757-6000	rose@edge.com
Mr.	Sean	Forbes	CFO	(512) 757-6000	sean@edge.com
Mr.	Pablo	Estrella	Director of Product Development	(512) 757-6123	pablo@edge.com
<b>Grand Totals (3 records)</b>					

The merge field would look like the following. Notice the comma (,) as the last component of the field:



Using a TableList field, you can display the results in a list:

The employees at your location are Rose Gonzalez, Sean Forbes, Pablo Estrella.

To create a TableList field:

Create a merge field with the following syntax:

```
{{TableList:DatasetName:FieldName:Punctuation}}
```

- Where:
  - DatasetName: is the name of the detail data set
  - FieldName: is the merge field name as displayed in the Template Builder window

Punctuation is the character that will separate the list values. You may not use a quote (") or colon (:) as the punctuation value. Duplicate values from the list are removed automatically.

## Individual values from ReportData or QueryData in PowerPoint

To refer to a field value from only the first row of ReportData or QueryData dataset, you may create a field with the "formal" naming convention: `{{DatasetName.FieldName}}`

- Where:
  - DatasetName corresponds to the name of the dataset from the Template Builder window.
  - FieldName is the field name copied to the clipboard from the Template Builder.

For fields from Master, Org or User, you may omit the DatasetName (and the dot) and simply refer to the field name.

```
{{ReportData.Full_Name}}
```

## Tips and Tricks

If this technique is used with a ReportData or QueryData sheet, data is drawn from the first row of the data. To use other rows from the data set, use either the Nth Row naming convention or a [Detail Region](#).

This naming convention only works within Text Boxes and Shapes. It does not work within [Detail Regions](#) or [Special Fields](#).

## Referring to the Nth Row in a ReportData or QueryData set in PowerPoint

If you know ahead of time that a detail dataset (ReportData or QueryData) will contain a certain number of rows, you may explicitly refer to a field in a particular row.

To refer to a specific row in a dataset, append a dot and the row index number to the field name. The first row's index number is 0 (zero), the second row index number is 1, and so on.

To refer to the Full\_Name field in the second row of data from ReportData, use:

```
{{ReportData.Full_Name.1}}
```

To merge all rows of data from a ReportData or QueryData set, use a [Detail Region](#).

## Excel Templates

You can merge data from Salesforce into formatted Excel spreadsheets that can contain text and tables.

Select one of the following topics for more information:

- [Excel Template Basics](#)
- [Advanced Excel Templates](#)

## Excel Template Basics

Basic information for creating Excel templates.

- [Excel Template Guide](#)
- [Getting Started with Excel Templates](#)
- [Designing Microsoft Excel Merge Templates](#)
- [Excel Merge Field Functions](#)
- [Known Limitations with Excel Templates](#)
- [Unsupported Excel Features](#)
- [Excel FAQs](#)

## Excel Template Guide

### Learn Excel Template Basics

- [Designing Microsoft Excel Merge Templates](#)

### Create Excel Templates

- [Getting Started with Excel Templates](#)
- [Excel Merge Field Parameters](#)
- [Special Excel Merge Fields](#)
- [Dynamic Formulas in Excel](#)
- [Set the Number of Rows Dynamically in a Conga Excel Template](#)
- [Use Named Ranges for Calculations in Excel Templates](#)
- [Group and Subtotal Data in an Excel Template](#)

### Troubleshoot Excel Templates

- [Excel FAQs](#)
- [Known Limitations with Excel Templates](#)
- [Unsupported Excel Features](#)
- [Product Data Limits](#)

## Getting Started with Excel Templates

Using the Template Builder to create your Microsoft Excel templates.

### The Template Builder

The Template Builder is an essential tool in the development of Conga Composer solutions. The Template Builder exposes the data retrieved by Conga Composer from the Master Object (and from any Salesforce Reports or Conga Queries referenced in the Conga Composer URL). You can then use these fields to build templates.

To learn more on how to use the Conga Template Builder, see [About Data in the Template Builder](#).

## Configuring for Excel

The Template Builder supports multiple template types. By default, it is set to create text-based merge fields for Microsoft Word.

1. Select the Excel radio button to format the merge fields for Microsoft Excel.
2. Select the field and copy it to the clipboard.
3. Paste that field into your template.

The format of an Excel merge field is:

- `&=Datasetname.Fieldname`
- `&=Master.ACCOUNT_NAME`

## Tips and Tricks

- The name of the dataset is case-sensitive. It must match the case as indicated on the Template Builder window. The Salesforce fieldname portion is not case-sensitive. In the image below the dataset name is Master with an upper case "M".

The screenshot shows an invoice template with the following content:

**Logo**

**&=Master.ACCOUNT\_** **INVOICE**

Account Accor&=Master.ACCOUNT\_ACCOUNTNUMBER  
 &=Master.ACCOUNT\_BILLINGADDRESS  
 Phone: &=Master.ACCOUNT\_PHONE  
 Prepared by: &=User.USER\_FULLNAME

DATE: 11/27/2013  
 Customer ID: ACCOUNT\_ACCOUNTNUMBER  
 Valid Until: 12/27/2013

**Customer**  
 &=Master.CONTACT\_FULLNAME  
 &=Master.ACCOUNT\_NAME  
 &=Master.ACCOUNT\_BILLINGADDRESS  
 &=Master.ACCOUNT\_PHONE

Product Name	Quantity	Unit Price	TOTAL

**TOTAL Due** &=Master.OPPORTUNITY\_AMOUNT

**TERMS AND CONDITIONS**  
 1. Customer will be billed after indicating acceptance of this quote

- You may only use one merge field per cell. Merge fields can be used as part of a more complex formula but require special adaptations in order to work with Conga Composer. To use a merge field in a formula, put the merge field in a non-printing region (or a different sheet) and then refer to that cell in the formula.

- Since you can retrieve multiple rows of data for each master object (e.g. opportunity line items for each opportunity), your Excel template will automatically expand to accommodate an unlimited number of detail rows.

Example

Add the following fields from the Master - Account dataset:

- ACCOUNT\_NAME
- ACCOUNT\_BILLINGADDRESS
- ACCOUNT\_ACCOUNTNUMBER,
- ACCOUNT\_PHONE

Then add the following field from the Master - Contact dataset:

- CONTACT\_FULLNAME

Format the fields to your liking in Excel, and save your template.

Now we'll add line items related to an opportunity from the OppLineItems dataset.

Copy and paste the following fields as shown below:

- Opportunity\_Lineitem\_ProductName
- Opportunity\_LineItem\_Qty(Numeric)
- Opportunity\_Lineitem\_UnitPrice(Numeric)
- Opportunity\_LineItem\_Total(Numeric)

When you copy the field from the Template Canvas, the Template Builder will automatically add the dataset name (in this case OppLineItems) to the field name. Conga Composer will automatically insert rows to the spreadsheet to accommodate the number of rows in the dataset. If, for example, the first LineItem will be inserted into cell A23, then a new row will be inserted into A24, and so on.

Now save the template and run the solution.

The final output would look like this:

University of Arizona		INVOICE	
888 N Euclid Hallis Center, Room 501Tucson, AZ 85721		DATE:	12/4/2013
Phone: (520) 773-9050		Customer ID:	CD736025
Prepared by: Estaban Blanco		Valid Until:	1/3/2014
<b>Customer</b>			
Jane Grey			
University of Arizona			
888 N Euclid Hallis Center, Room 501Tucson, AZ 85721United States			
(520) 773-9050			
Product Name	Quantity	Unit Price	TOTAL
GenWatt Diesel 1000kW	12	100000	\$ 1,200,000.00
GenWatt Diesel 10kW	2	5000	\$ 10,000.00
GenWatt Diesel 200kW	2	25000	\$ 50,000.00
SLA: Platinum	1	40000	\$ 40,000.00
SLA: Silver	1	20000	\$ 20,000.00
TOTAL Due			\$ 1,320,000.00
<b>TERMS AND CONDITIONS</b>			

Rows Inserted Dynamically.

- ✓ To learn more on how to use Excel templates with Conga Composer, including how to merge detail data and use functions and formulas, watch this [video](#).

## Designing Microsoft Excel Merge Templates

Conga Composer can merge data into an Excel template.

Conga Composer supports Excel templates saved in the Excel 2003 (".xls"), or Excel 2007/2010 (".xlsx") format. However, we recommend the Excel 2003 format (".xls") to ensure file format compatibility across a broad audience.

Excel templates work only with Conga Composer, not Conga Mail Merge.

The [Template Builder](#) supports multiple template types. By default, it is set to create text-based merge fields for Microsoft Word.

Select the Excel radio button to format the merge fields for Microsoft Excel.

## Excel Merge Field Functions

You can refine Conga Composer's behavior when merging to Excel with field parameters. Each of the following parameters may be appended to the end of a merge field within parentheses, separating each with a comma.

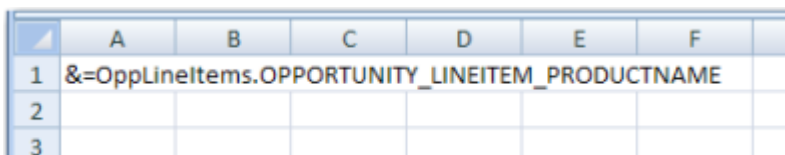
Functions	Description
Numeric	<p>Converts text data to numeric data, if possible.</p> <p>After converting to numeric, you may apply numeric formatting to the field.</p> <p>Since ReportData columns are retrieved as text values, this parameter can be particularly useful in converting a ReportData column value from text to numeric</p> <p><code>&amp;=ReportData.AMOUNT(Numeric)</code></p>
Horizontal	<p>Writes data left-to-right, instead of top-to-bottom.</p> <p>Applies to datasets with multiple rows of data (reports, queries, OppLineItems, QuotLineItems)</p> <p><code>&amp;=ReportData.CONTACT_LASTNAME(Horizontal)</code></p>
Skip:n	<p>Skip n number of rows for each row of data.</p> <p><code>&amp;=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME(Skip:1)</code></p>
Noadd	<p>Do not add extra rows (or columns) to fit data, i.e., detail data is written to cells without adding new rows (or columns) during the merge process</p> <p><code>&amp;=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME(Noadd)</code></p>
Group:Normal	<p>Groups the merge field values and displays the grouped value once per data group. Frequently accompanied by "Skip".</p> <p>See "FAQ: Grouping Data" for more details.</p> <p><code>&amp;=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTFAMILY(Group:Normal)</code></p>



Functions	Description
Group:Merge	<p>Groups the merge field values and displays the grouped value once per data group. The cells in the group-by field for each group set are merged together. Frequently accompanied by "Skip".</p> <p>&amp;=</p> <p>OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTFAMILY(Group:Merge)</p>
SubtotalN: GroupByColumn	<p>Performs a summary operation for a specified field related to a grouped field. The summary action, determined by N, is in the range 1 through 10 (see table below). The GroupByColumn is the field on which the data is grouped (separately).</p> <p>Note: This function requires a separate column in which the Group parameter has been utilized.</p> <p>N value choices:</p> <ul style="list-style-type: none"> <li>• 1=AVERAGE</li> <li>• 2=COUNT</li> <li>• 3=COUNTA</li> <li>• 4=MAX</li> <li>• 5=MIN</li> <li>• 6=PRODUCT</li> <li>• 7=STDEV</li> <li>• 8=STDEVP</li> <li>• 9=SUM</li> <li>• 10=VAR</li> </ul>

**Example 1:**

Say we want to merge Opportunity Line Items on an Excel template. To merge them vertically (the typical method), your template would look like this:



	A	B	C	D	E	F
1	&=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME					
2						
3						

Assuming we have three items from the OppLineItems sheet, the output would look like this:

	A	B	C	D	E	F
1	Conga Courier					
2	Conga Merge					
3	PharmaNow CRM					
4						

**Example 2:**

To write data left-to-right, include the Horizontal parameter. In this example, the Product Names from an opportunity are written left-to-right.

```
&=ReportData.OPPORTUNITY_LINEITEM_PRODUCTNAME(Horizontal)
```

	A	B	C	D	E	F	G
1	&=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME(Horizontal)						
2							

The output after adding the Horizontal parameter would look like:

	A	B	C	D	E
1	Conga Courier	Conga Merge	PharmaNow CRM		
2					
3					
4					

Data arranged horizontally

**Example 3:**

Extending the previous example, say we want Opportunity Line Items (and Sales Price) to appear from left-to-right across a page. We've also included headers each product column (Product 1, Product 2 and Product 3). To prevent Conga Composer from inserting new columns, we've added the Noadd parameter:

	A	B	C	D	E
1	Product 1	Product 2	Product 3		
2	&=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME(Horizontal, Noadd)				
3	&=OppLineItems.OPPORTUNITY_LINEITEM_SALESPRICE(Horizontal, Noadd)				

```
&=ReportData.OPPORTUNITY_LINEITEM_PRODUCTNAME(Horizontal, Noadd)
```

The output from this horizontal merge would look like:

	A	B	C	D
1	Product 1	Product 2	Product 3	
2	Conga Courier	Conga Merge	PharmaNow CRM	
3	\$360	\$144	\$780	
4				

#### Example 4:

The skip:n parameter can be helpful when you want to insert alternating data on every other row (from fields containing multiple values).

In this example, we'd like to have an Opportunity Line Item Product appear, then the Sales Price below the product, followed by another product and price, and so on.

Usually, the Noadd parameter accompanies the skip:n parameter.

	A	B	C	D	E	F
1	=&OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME(Skip:1, Noadd)					
2	=&OppLineItems.OPPORTUNITY_LINEITEM_SALESPRICE(Skip:1, Noadd)					
3						
4						
5						

The output from this template would look like:

	A	B	C	D
1	Conga Courier			
2		\$360		
3	Conga Merge			
4		144		
5	PharmaNow CRM			
6		780		
7				
8				


## Known Limitations with Excel Templates

- Macros in Excel templates will not be executed upon merging. If the output file type is chosen as Excel, a user can manually execute the macros but Conga will not do so automatically in the merge process. If the output file type is chosen as PDF, the macros in the Excel template will not execute before conversion to PDF.
- The use of large exponential equations has inconsistent results. At this time Conga has not been able to isolate a particular "limit" with regard to the size and complexity of the formula that might result in error: =A5^(B5+C2)
- Some chart types when converting to PDF: We've found that some complex chart types revert to more simple versions when converting to PDF.

- Excel does not support nested tables. To leverage nested tables, use a Word template.
- Nested SOQL queries are not functional with Excel. The resulting data from a nested SOQL query is returned as a separate dataset. Pivot tables are an alternative for displaying nested SOQL query data in Excel. Pivot Tables can produce an end result that is similar to nested SOQL functions that are supported in Word. Conga Support does not provide support for advanced Excel formulas such as pivot tables.
- Conga Composer cannot merge data in Excel headers and footers.
- Page breaks in an Excel template are not supported.
- We recommend saving your Excel template as .xls (rather than .xlsx) format to ensure the widest compatibility for end users.

## Unsupported Excel Features

The following Excel items are unsupported in Composer.

 A partial list of features are listed below and can be used successfully, but Conga customer support is unable to provide direct support for advanced Excel features.

- Array formulas
- Images
- Excel picklists
- Page breaks on an individual sheet
- Pivot tables and other advanced Excel features (Vlookups, sumif(s), countif(s), IF(s), Textjoin, and so on)
- XML Maps
- Text boxes

Using Skip:x in two different columns leads to unexpected output behavior and is not supported. Conga Composer does not support any formula containing (but not limited to) the following functions (see table below):

ASC	CUBEMEMBERPROPERTY	JIS
BAHTTEXT	CUBERANKEDMEMBER	PHONETIC
CLEAN	CUBESET	REGISTER.ID
CALL	CUBESETCOUNT	RTD
CODE	CUBEVALUE	SQL.REQUEST

CUBEKPIMEMBER	EUROCONVERT	YIELD
CUBEMEMBER	INFO	YIELDDISC

## Excel FAQs

### How Can I Use a Merge Field within a Formula?

For merge fields that will be referenced by other fields using Excel formulas, we've found the easiest way to approach this is to place merge in a non-printing area of the worksheet or on a separate worksheet.

Then, in formulas elsewhere on the spreadsheet, use the cell coordinates to refer to the value.

To use one of these cells in a formula, you would simply refer to the cell reference in the formula. Here, we're combining the city, state, and postal code values with a comma-space after the city field and a space between the state and postal code.

```
=CONCATENATE(I9, ", ", I10, " ", I11)
```

Use =CONCATENATE(...) to join field values together. The abbreviated ("&") function is not supported. When converting from Excel to PDF, use CONCATENATE instead of CONCAT

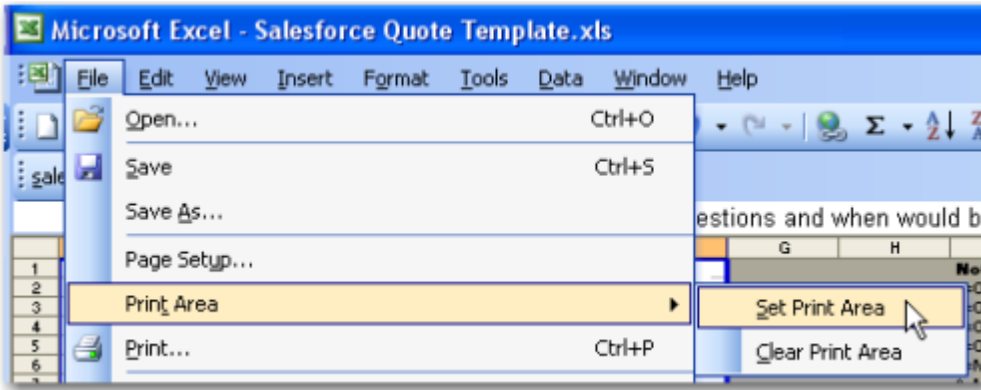
### Converting to CSV

Composer is not compatible with CSV as a template type. However, after merging an Excel template you can download the output file and save it as a CSV (comma delimited) file if desired.

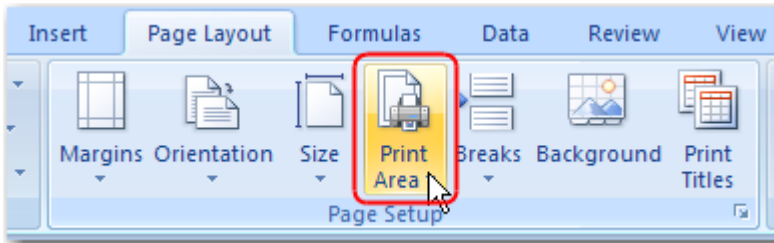
### How Do I Set the Printable Area of the Spreadsheet?

You can flag the portion of the spreadsheet you want to print. To identify the part of the spreadsheet currently specified to print, choose **View** → **Page Break Preview**.

To change the printable area: Select the portion of the spreadsheet that should print using your mouse.



To set the printable area to the selected cells:

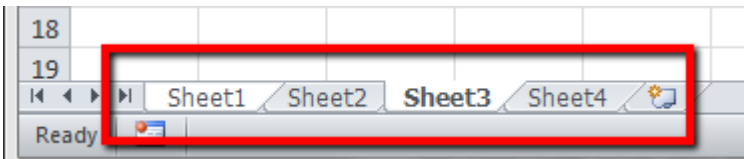


Word 2003: Choose **File** → **Print Area** → **Set Print Area**.

Word 2007: On the **Page Layout** → **Print Area**, and then click **Set Print Area**. Choose **View** → **Normal** to return to the previous view mode.

### How Can I Determine which Sheets to Include in an Output PDF?

You can control which sheets are included in an output PDF from a Conga Excel template. To do so, select the sheets you want included by holding the Ctrl key and clicking on each desired sheet.



Excel will highlight the "selected" sheets in white, while the non-selected sheets will remain with a gray background.

In this example, "Sheet1" and "Sheet3" are selected and would be the sheets included in a PDF generated from this template.

### What's the Best Way to Format Text Paragraphs?

Conga Composer cannot dynamically increase row height to accommodate multiple lines of text. For this reason, it is usually helpful to merge (not mail merge, just merge) a number of rows sufficient to your needs (**Format Cells** → **Alignment Tab** → **Merge cells**), leaving unmerged rows to the right or left that will preserve the height of the merged area after Salesforce field content is merged.

23							
24	Thank you for taking the time to meet with &=Org.Org_Name and to learn how our solutions can						<b>Cells preserved</b>
25	meet your organization's needs. As promised, I have put together a quote covering the products						
26	and services that we discussed during our recent meeting.						
27							

Here is the formula used in the cell above:

```
=CONCATENATE("Thank you for taking the time to meet with ", A7, " and to learn how our solutions can meet your organization's needs. As promised, I have put together a quote covering the products and services that we discussed during our recent meeting.")
```

## Advanced Excel Templates

This section contains information on advanced template-building techniques and tricks.

- [Dynamic Formulas in Excel](#)
- [Graphing in Excel](#)
- [Group and Subtotal Data in an Excel Template](#)
- [Set the Number of Rows Dynamically in a Conga Excel Template](#)
- [Special Excel Merge Fields](#)
- [Use Named Ranges for Calculations in Excel Templates](#)

## Dynamic Formulas in Excel

Dynamic Formulas allow you to insert Excel's formulas into cells even when the formula must reference rows that will be inserted later during the merge process. They can repeat for each inserted row or use only the cell where the field is placed.

Dynamic Formulas begin with &= and are followed by an Excel formula.

Repeating Dynamic Formulas begin with &=&= and are followed by an Excel formula.

You may use most of Excel's functions in a Dynamic Formula. The following parameters apply to Dynamic Formulas.

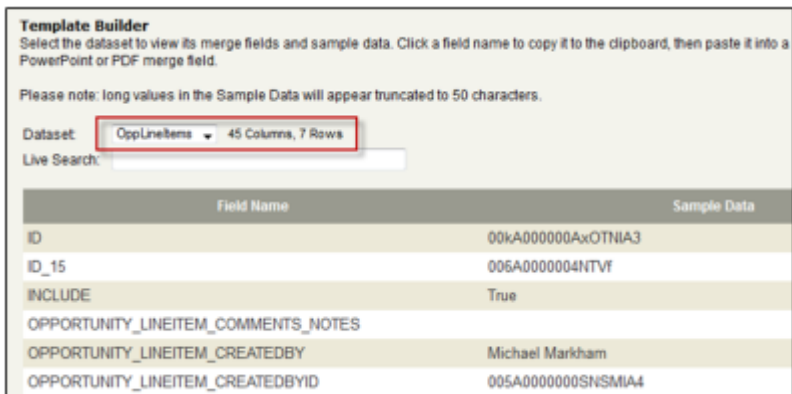
PARAMETER	DESCRIPTION
{r}	<p>Indicates to use the current row number</p> <p>Used in conjunction with a column letter to indicate a cell on the current row.</p> <p>When the following is placed in cell C1:</p> $=&A\{r\}*B\{r\}$ <p>results in the formula =A1*B1</p>
{integer}	<p>Offset to current row number, where integer is a positive or negative number that represents the number of rows above or below the current row.</p> <p>Used in conjunction with a column letter to indicate a particular cell offset by the number of rows indicated by the integer.</p> <p>{2} would mean to reference a cell two rows below the current row</p> <p>{-1} would mean to reference a cell one row above the current row</p> <p>When placed in cell A10:</p> $=&E\{1\}*D\{1\}$ <p>results in the formula =E11*D11</p>

**Example 1:**

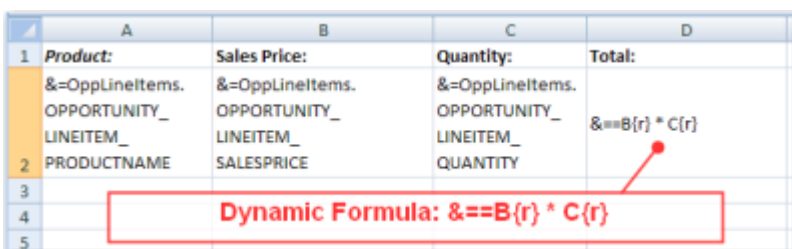
Say we're creating a quote using Opportunities with Opportunity Line Items and we'd like to do some of the math for each Line Item in Excel. (This example is purely academic - Salesforce already computes the product for you.)

We'll start with a Template Builder window that looks like this:





We'll create a template that includes a Dynamic Formula to find the product of the price multiplied by the quantity.



The output from this merge would look like this:

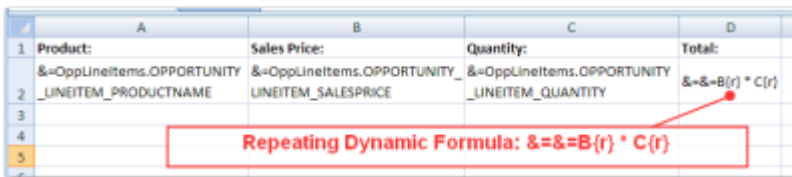


In this case, the total does not repeat for each row. If you want it to repeat, use a Repeating Dynamic Formula as shown next.

**Example 2:**

Let's extend the previous example by changing the Dynamic Formula to a Repeating Dynamic Formula.

Here's the modified template:



The output from this merge would look like this:

	A	B	C	D
1	<b>Product:</b>	<b>Sales Price:</b>	<b>Quantity:</b>	<b>Total:</b>
2	Conga Courier	\$360	5	\$1,800
3	Conga Merge	\$144	60	\$8,640
4	PharmaNow CRM	\$780	1	\$780

## Graphing in Excel

This technique shows how to You can create dynamic graphs in Microsoft Excel. The indicated region expands with the merged data.

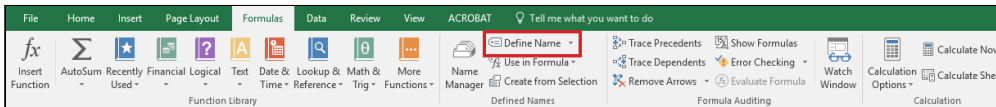
To setup templates in Excel:

1. In a new Excel worksheet, type the following information:

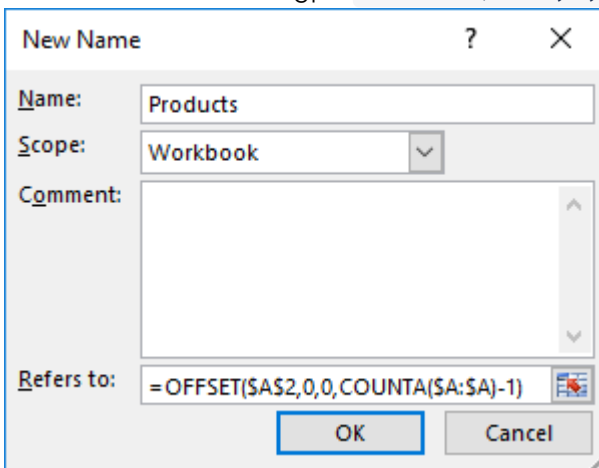
	A	B
1	Product Name	Sales Price
2	<code>&amp;=OppLineItems.OPPORTUNITY_LINEITEM_PRODUCTNAME</code>	<code>&amp;=OppLineItems.OPPORTUNITY_LINEITEM_SALESPRICE(numeric)</code>

i (numeric) is required to ensure proper formatting of merge data numbers or currency values.

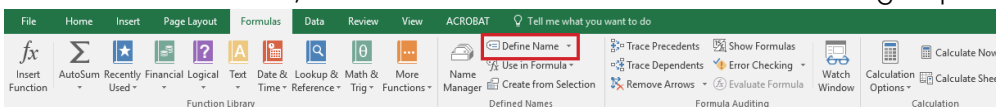
2. On the Formulas tab, click **Define Name** in the Defined Names group.



3. In the Name box, type: Products.
4. In the Refers To box type `=OFFSET($A$2,0,0,COUNTA($A:$A)-1)` and then click **OK**.

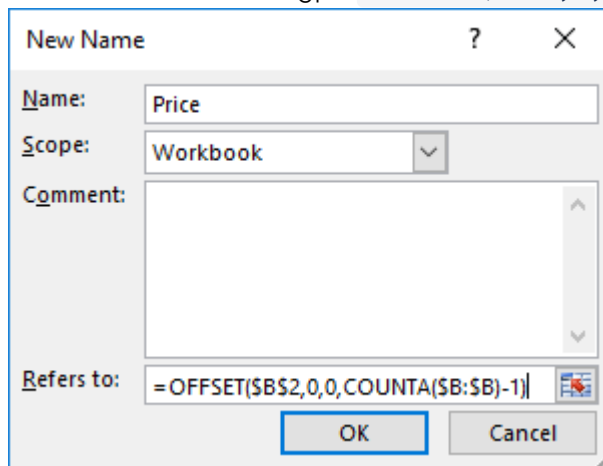


5. On the Formulas tab, click **Define Name** in the Defined Names group.

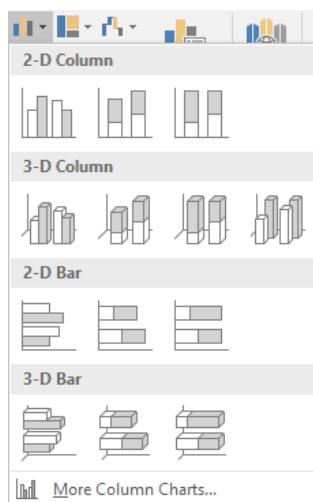
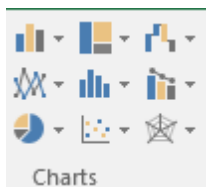


6. In the Name box, type: **Price**.

7. In the Refers To box type `=OFFSET($B$2,0,0,COUNTA($B:$B)-1)` and then click **OK**.



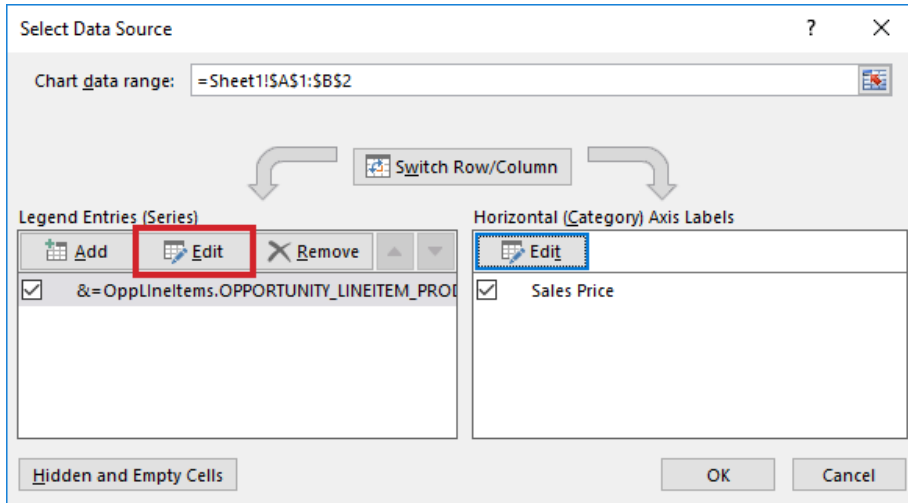
8. In cell C1, type the following formula: `=RAND()*0`. This formula updates the OFFSET formulas in the defined names.
9. Select cells A1:B2.
10. On the **Insert** tab, click a chart, and then click a chart type.



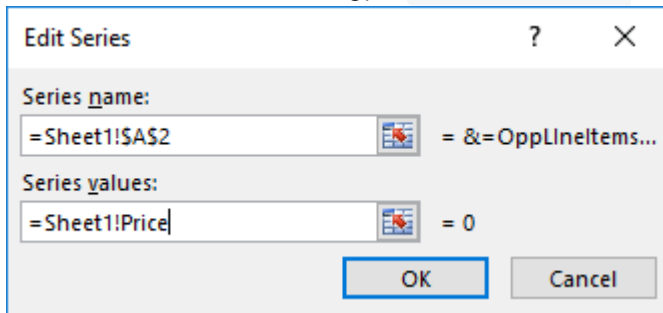
11. Click the **Design** tab click the **Select Data** in the Data group.



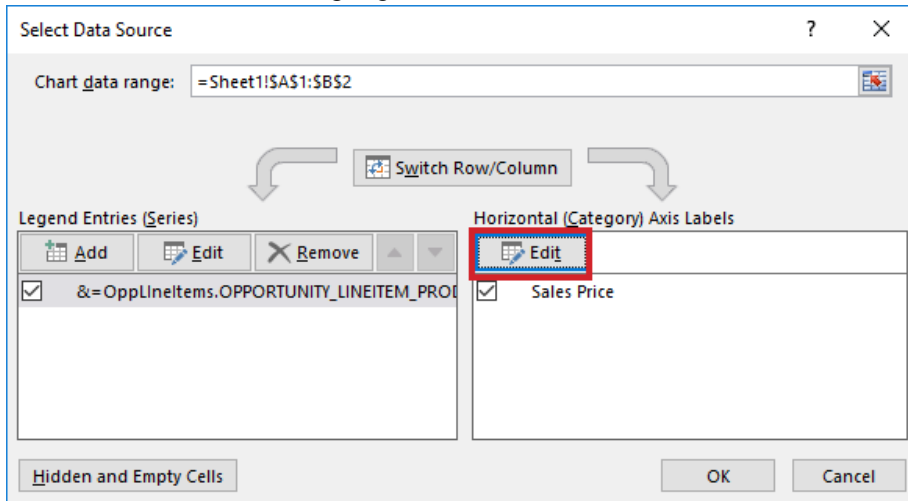
12. Under Legend Entries (Series) click **Edit**.



13. In the Series values box type '=Sheet1!Price' and click **OK**.



14. Under Horizontal (Category) Axis Labels click **Edit**.



15. In the Axis label range box type '=Sheet1!Products' and click **OK**.

16. Save your workbook.

17. On the Opportunity in [Salesforce.com](https://www.salesforce.com), click on your custom Composer button/link.

18. Choose **Select Template** from your Local Computer and browse to your new workbook.

19. Click **Merge**.

## Group and Subtotal Data in an Excel Template

You can group and subtotal columns of merged data using the “Group” and “Subtotal” parameters. For more information, see [Excel Merge Field Parameters](#).

In the following example, we demonstrate how to group and subtotal Opportunity Line Items by Product Family.

**⚠** For Excel to sort the fields properly post merge, sort by the same field on your query or report.

### Step 1: Non-Grouped Template

Start with a simple template that displays the non-grouped data.

Template:

<b>&amp;=Master.ACCOUNT_NAME</b>		
<b>&amp;=Master.MailingCity</b>		
<b>Product Family</b>	<b>Product Name</b>	<b>Total Price</b>
&=OppLineItems.Product_Family	&=OppLineItems.OPP ORTUNITY_LINEITEM_ PRODUCTNAME	&=OppLineItems.OPPORTUNITY_ LINEITEM_TOTALPRICE
		0

Output:

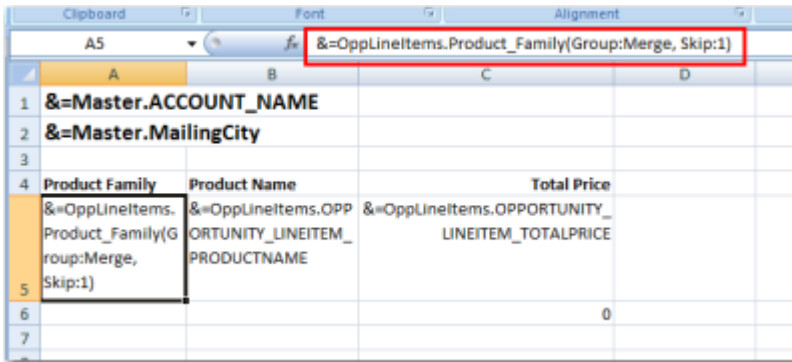
<b>Mini Pharma's Revenge</b>		
<b>Aspen</b>		
<b>Product Family</b>	<b>Product Name</b>	<b>Total Price</b>
Products	PharmaNow CRM	-17,160
Products	Conga Courier	1,800
Products	German Recycle Truck	10,000
Products	German Recycle Truck	10,000
Services	Conga Merge Service	800
Services	Courier Services	770
Services	PharmaNow Services	1,680
Training	Conga Merge	6,950
Training	Conga Courier Point	2,385
		17,225

### Step 2: Group by Product Family

Next, modify the merge field that represents the Product Family. Add the parameter to group on that field and to skip a row between Product Family values to make room for a subtotal (in a subsequent step).

Sort your report by the same field you are using as a group. For example, sort the example report by Product Family.

Template:



Output:

Product Family	Product Name	Total Price
Products	PharmaNow CRM	-17,160
	Conga Courier	1,800
	German Recycle Truck	10,000
	German Recycle Truck	10,000
Services	Conga Merge Service	800
	Courier Services	770
	PharmaNow Services	1,680
Training	Conga Merge	6,950
	Conga Courier Point	2,385

Using the SUBTOTAL9 function as depicted below ensures that each group maintains a subtotal and prevents blank lines from appearing below the merge results.

### Step 3: Subtotal on Price

Next, insert a formula to subtotal on Total Price. We added bold formatting to the formula to emphasize it.

The Subtotal function must evaluate the same field on which you are grouping. For example, if the field being grouped on is Product Family, then the group parameter will state `&=OppLineItems.Product_Family(Group:Merge, Skip:1)`, and Subtotal will be entered as `&=Subtotal9:OppLineItems.Product_Family`.

Template:

Output:

	A	B	C
1	<b>Mini Pharma's</b>	<b>Revenue</b>	
2	<b>Aspen</b>		
3			
4	<b>Product Family</b>	<b>Product Name</b>	<b>Total Price</b>
5	Products	PharmaNow CRM	-17,160
6		Conga Courier	1,800
7		German Recycle Truck	10,000
8		German Recycle Truck	10,000
9			<b>4,640</b>
10			
11	Services	Conga Merge Service	800
12		Courier Services	770
13		PharmaNow Services	1,680
14			<b>3,250</b>
15			
16	Training	Conga Merge	6,950
17		Conga Courier Point	2,385
18			<b>9,335</b>
19			

Notice each subtotal in bold text.

#### Step 4: Compute Grand Total

Finally, insert a formula to find the grand total of the Price column. Do not use the =SUM function to add this column, as it will include the subtotal values in the result. Instead, we'll include the Excel formula:

```
=SUBTOTAL(9, C5:C5)
```

We also added a label for the total ("Grand Total") and formatted the values as currency (USD).

Template:

The screenshot shows an Excel spreadsheet with the following content:

	A	B	C
1	<b>&amp;=Master.ACCOUNT_NAME</b>		
2	<b>&amp;=Master.MailingCity</b>		
3			
4	<b>Product Family</b>	<b>Product Name</b>	<b>Total Price</b>
5	<b>&amp;=OppLineItems.</b>	<b>&amp;=OppLineItems.OPP</b>	<b>&amp;=OppLineItems.OPPORTUNITY_</b>
6	<b>&amp;=Subtotal9:OppLineItems.Product Family</b>		
7		<b>Grand Total</b>	<b>\$0</b>
8			
9			

The formula bar at the top shows `=SUBTOTAL(9, C5:C5)`. A red arrow points from the formula bar to the cell containing '\$0'.

Output:

The screenshot shows the output of the Conga Excel template with the following data:

	A	B	C
1	<b>Mini Pharma's</b>	<b>Revenue</b>	
2	<b>Aspen</b>		
3			
4	<b>Product Family</b>	<b>Product Name</b>	<b>Total Price</b>
5	Products	PharmaNow CRM	-\$17,160
6		Conga Courier	\$1,800
7		German Recycle Truck	\$10,000
8		German Recycle Truck	\$10,000
9			<b>\$4,640</b>
10			
11	Services	Conga Merge Service	\$800
12		Courier Services	\$770
13		PharmaNow Services	\$1,680
14			<b>\$3,250</b>
15			
16	Training	Conga Merge	\$6,950
17		Conga Courier Point	\$2,385
18			<b>\$9,335</b>
19			
20		<b>Grand Total</b>	<b>\$17,225</b>
21			

## Set the Number of Rows Dynamically in a Conga Excel Template

Set the number of rows dynamically in a Conga Excel template.

When using `=ROW()` you must subtract the number of rows above the formula.

If, for example, your formula goes in cell A5:

If you put `"=ROW()"` in that cell, the result is "5". So by subtracting 4 (the number of rows above it), the result gives you a 1. Thus, the formula is `=ROW()-1` at this point.

Finally, to get it to repeat for each detail row of merge data, let's make it a dynamic repeating formula: `&=&=ROW()-4`

That should do the trick! Just adjust the "4" accordingly.



## Special Excel Merge Fields

Conga Composer offers several additional merge fields for even more flexibility in your Excel merge templates.

### Today's Date (Text)

Create a `&=Date.Today` merge field to insert a Date field and insert the current date as a text string.

 You cannot apply a date-format to this value because it is a text string.

### Today's Date (numeric)


Create a `&=Date.Today(numeric)` merge field to insert the current date and apply Excel cell formatting.

### Today's Date (Date/Time Value)

Create a `&=Date.TodayDT` merge field to insert the current date as a date/time value. You can format the resulting date with any of Excel's date-formatting options.

### Rich Text to HTML (HTML)

If `<br>` appears after the document is merged, add `(HTML)` to the end of the merge field.

 The HTML function is limited to standard text characters. The HTML function will not render advanced formatting and content such as bullets, images, and font colors.

Example: `&=Master.MailingAddress(HTML)`

### Hyperlinks in Excel

Conga cannot create hyperlinks in Excel. You can use Excel link a merged Conga field. Merge the URL into the first column. Then use a hyperlink formula in the next column, for example: `&=HYPERLINK(A{r}, "click here")`. Optionally, you can hide Column A so only "click here" in Column B is visible to the end user.

### Percentage Fields

Percentage fields in Excel are not actually a percent in Salesforce. You must apply an Excel formula to your output. A best practice is to create two columns: one for the merge field and another for the calculation. After testing, hide the merge field cell.

<code>&amp;=Opportunity.Probability_%(numeric)</code>	<code>&amp;=&amp;=C{r}/100</code>
---	-----------------------------------

### Row Number Field

The ROW\_NUMBER field numbers each line item in a detail dataset and is included by default in every report or query dataset. Insert this field into any detail region in the template to number the results in the merged output file.

When the `{{ROW_NUMBER}}` merge field is placed in the far left column:

A	B	C	D
Row Number	Product Name	Opportunity Lineitem Quantity	Opportunity Lineitem Totalprice
<code>&amp;=OppLineItems ROW_NUMBER</code>	<code>&amp;=OppLineItems PRODUCT_NAME</code>	<code>&amp;=OppLineItems OPPORTUNITY_LINEITEM_QUANTIT</code>	<code>&amp;=OppLineItems OPPORTUNITY_LINEITEM_TOTALPRICE</code>

The merged output looks like this:

A	B	C	D
Row Number	Product Name	Opportunity Lineitem Quantity	Opportunity Lineitem Totalprice
1	GenWatt Diesel 10kW	1	5000
2	GenWatt Propane 100kW	2	30000
3	GenWatt Diesel 200kW	5	125000
4	GenWatt Gasoline 300kW	3	105000

### Hiding Merged Cells

1. Select the cell or range of cells that contains values that you want to hide. For more information, see [Select cell contents in Excel](#). The selected cells appear blank on the worksheet, but a value appears in the formula bar when you click one of the cells.
2. On the Home tab, click the **Dialog Box Launcher**.
3. In the Category box, click **Custom**.
4. In the Type box, select the existing codes.
5. Type `;;;` (three semicolons).
6. Click **OK**.

## Use Named Ranges for Calculations in Excel Templates

Named Ranges in Excel can be used to define blocks of data and create a single data point out of the block. This is especially useful when functions need to be calculated on an unknown number of rows of data.

## Defining a Named Range in a Conga Composer Excel Merge Template

1. In Excel, highlight the merge field cell and the cell above it.
2. Overwrite the value in the Name box with whatever name you want to give this range, a simple name is best as this will be referenced later.

 It is recommended to create Named Range(s) for the cell(s) where calculations are done so that data can be used in further calculations

The Named Range(s) can now be referenced directly within Excel formulas regardless of the number of rows of data added during the merge.

## HTML Email Templates

You can merge data from Salesforce into an HTML email by using Conga Email Templates.

Select one of the following topics for more information:

- [HTML Email Template Basics](#)
- [Advanced HTML Email Templates](#)

## HTML Email Template Basics

Basic information about creating HTML Email templates for Composer.

## HTML Email Template Guide

### Learn HTML Email Template Basics

- [Before You Get Started With HTML Email Templates](#)
- [About Conga Email Templates](#)

### Create HTML Email Templates

- [Build a Conga Email Template Using the Template Builder](#)
- [Create Merge Fields in a Conga Email Template](#)
- [Format Merge Fields in a Conga Email Template](#)

### Pre-Selecting a Conga Email Template in Composer

- [Create a Conga Email Template](#)

### The Conga Email User Interface

- [Conga Email Template Special Merge Fields](#)

- [Row Number Field in HTML Email Template](#)

## Troubleshoot HTML Email Templates

### [Conga Email Template FAQs](#)

## Before You Get Started with HTML Email Templates

You'll need to be able to build a Conga Composer button. You can find the details on how to do this in Conga Composer QuickStart 1 - Creating Your First Conga Composer Solution.

Due to the potentially complicated nature of HTML source code, our support staff can only assist with the creation of Conga Email Templates edited using the non-Source view mode in the Conga Advanced Editor. Editing the template through the Source view in the Conga Advanced Editor constitutes an unsupported, "use-at-your-own-risk" configuration.

Conga highly recommends excluding the "Use attachment for template body" checkbox field from the Conga Email Template page layout. Use of this field is only suggested if your Conga Email Template is more than 32,000 characters, in which case the body of the Conga Email Template can be added as an attachment to the Conga Email Template record.

## About Conga Email Templates

Conga Composer can merge Salesforce data into an HTML email document.

With Conga Email, you create a template record in the new custom object included when you install Conga from the AppExchange. Then, you build a merge template in the record's Rich Text field using the Conga Advanced Editor.

Conga Composer collects the information from Salesforce and assembles the HTML rich-text email but does not send it. Emails are sent through the Salesforce email service or SendGrid, if you have it enabled.

Salesforce limits the number of emails that can be sent through the API each 24 hours. Contact Salesforce Support for details on the limitation of your instance.

Once the daily API email limit is exceeded, the Composer/Conga Batch/Conga Trigger processes return the following error: "*Problem: ERROR: Email delivery failed: SINGLE\_EMAIL\_LIMIT\_EXCEEDED.Failed to send email.*"

If your email volume exceeds the Salesforce limit, you can integrate SendGrid email delivery service in your Conga Composer solutions. For more information, contact [SendGrid](#).

A Conga HTML email template can:

- Merge data fields from the Master, Org, and User datasets.

- Merge report and query fields using table- and paragraph-based detail regions.
- Hide tables based on the absence of rows on a report or query using TableHide.
- Calculate the sum of a table column using SUM(ABOVE).
- Display a comma-separated list of report or query values using TableList.
- Merge images into image fields.
- Generate hyperlink fields.
- Replace merge data values using the REPLACE: prefix.
- When merging rich text fields in Conga HTML Email Templates, Conga Composer will automatically read the HTML code behind a rich text field and render the rich text. Therefore, the "HTML:" prefix is not necessary on the merge field for a rich text field.

A Conga HTML email template may be combined with one or more document templates. Outgoing emails may be delivered via Salesforce email as well as through Conga Composer's integration with [SendGrid](#).

If you plan to use Conga HTML email templates with custom objects in Salesforce, you must ensure the object has Allow Activities enabled in the Custom Object definition.

## Create a Conga Email Template

You can create a Conga Email Template (CET) manually. You can also create CETs within Solution Manager. For more information, see [Add a New Conga Email Template to a Solution using Solution Manager](#).

To create a Conga Email Template:

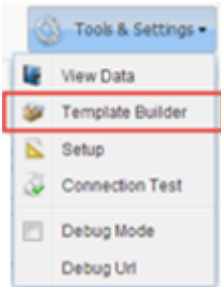
1. Click the **Conga Email Templates** tab.
2. Click **New** to create a record.
3. Enter a description and basic information about the new template and save the record. Do not create the email body here. Instead, use the Advanced Editor.
4. Click **Advanced Editor** to create the body of the template.
5. Copy and paste merge fields from the Template Builder into the Advanced Editor. The template builder is located under tools and settings in the button/solution.

Hi {{CONTACT_FIRSTNAME}},		
Please complete payment for the attached invoice. Here's a summary of what's included:		
PRODUCTS		
PRODUCT NAME	QUANTITY	TOTAL PRICE
{{TableStart-OpplLineItems}} {{OPPORTUNITY_LINEITEM_PRODUCTNAME}}	{{OPPORTUNITY_LINEITEM_QUANTITY}} # Number0}}	{{OPPORTUNITY_LINEITEM_TOTALPRICE}} # Currency}} {{TableEnd-OpplLineItems}}
	<b>TOTAL:</b>	{{=SUM(Above) /# Currency}}
Please see the attachment for complete details. We appreciate the opportunity to be of service.		
Kind regards,		
{{USER_FIRSTNAME}}		
{{USER_NAME}}		
{{USER_PHONE}}		
{{USER_COMPANYNAME}}		
<small>Powered by Conga Composer®</small>		

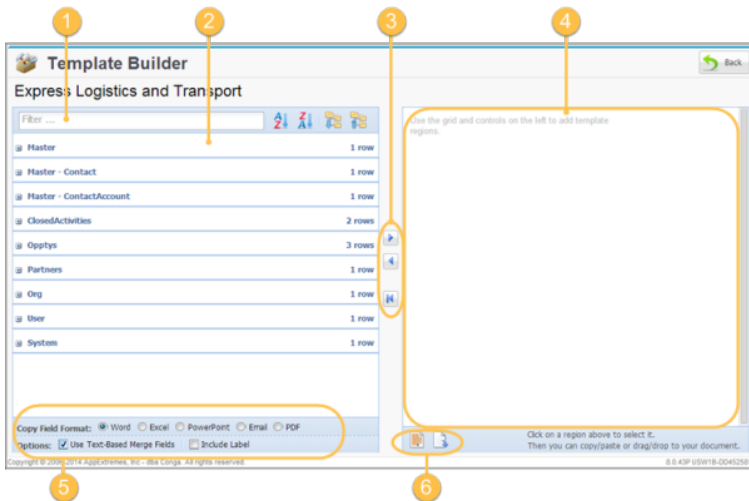
6. Click **Save**.

## Build a Conga Email Template using the Template Builder

An essential tool in the development of Conga HTML Email Templates is the Template Builder.



The Template Builder is located in the Tools & Settings drop-down menu in the upper-right corner of the Conga Composer page.



The Template Builder exposes the data retrieved by Conga Composer from the Master Object and from any Salesforce Reports or Conga Queries referenced in the Conga Composer URL.

The field names listed in the left pane indicate the field names you must use in your templates. The field names displayed here will be used to create merge fields on your Word, Excel, PowerPoint, Email or PDF template which will populate the corresponding data from Salesforce.

1	Use the Filter box to quickly locate data fields that match your search terms.
2	Click the plus sign next to a data set to expand it to see and select data fields.

3	Use arrows to move selected data to and from the template work area.
4	Stage data fields and tables of repeating data in the template work area before adding the fields to the template.
5	Choose the type of template you are building, as well as whether to use text-based merge fields and labels as data fields are moved to the template work area.
6	Select fields in the template work area and then use the buttons to copy and paste fields to your template or let Template Builder create and populate a new template. You can also copy and paste or drag and drop fields from the template work area to an open template document.

View Data (available from the Tools & Settings menu on the Conga Composer window) will download the available data fields into an Excel workbook.

## Format Merge Fields in a Conga Email Template

Similar to merge fields in Microsoft Word templates, merge fields in HTML email templates provide several formatting switches (called “picture switches”) that affect the appearance of your merged data – this is invaluable, particularly for numbers and dates. You can use the Template Builder to add these picture switches to your merge fields so the merged data is correctly formatted in your email. Alternately, you can manually add a picture switch to your field by simply typing it into your merge field in the HTML email template.

To manually add a picture switch to a merge field:

1. Click after the merge field name, but inside the double curly braces.
2. Type a space and picture switch.

Example: `{{ACCOUNT_CREATEDDATE \@ MM/dd/yy}}`

You can format the following types of fields:

### Date Fields

Parameter	Description		
Month (M)	Displays the month.		
	<table border="1"> <tr> <td data-bbox="676 392 863 575">M</td> <td data-bbox="863 392 1422 575">Displays the month as a number without a leading zero for single-digit months. February is shown as 2.</td> </tr> </table>	M	Displays the month as a number without a leading zero for single-digit months. February is shown as 2.
	M	Displays the month as a number without a leading zero for single-digit months. February is shown as 2.	
	<table border="1"> <tr> <td data-bbox="676 593 863 777">MM</td> <td data-bbox="863 593 1422 777">Displays the month as a number with a leading zero for single-digit months. February is shown as 02.</td> </tr> </table>	MM	Displays the month as a number with a leading zero for single-digit months. February is shown as 02.
	MM	Displays the month as a number with a leading zero for single-digit months. February is shown as 02.	
<table border="1"> <tr> <td data-bbox="676 788 863 972">MMM</td> <td data-bbox="863 788 1422 972">Displays the month as a three-letter abbreviation. February is shown as Feb.</td> </tr> </table>	MMM	Displays the month as a three-letter abbreviation. February is shown as Feb.	
MMM	Displays the month as a three-letter abbreviation. February is shown as Feb.		
<table border="1"> <tr> <td data-bbox="676 983 863 1126">MMMM</td> <td data-bbox="863 983 1422 1126">Displays the full name of the month. February.</td> </tr> </table>	MMMM	Displays the full name of the month. February.	
MMMM	Displays the full name of the month. February.		
The letter M must be uppercase to distinguish months from minutes.			



Parameter	Description								
Day (d)	<p data-bbox="676 331 1366 365">Displays the day of the month or the day of the week.</p> <table border="1" data-bbox="676 387 1426 1249"> <tbody> <tr> <td data-bbox="676 387 855 622">d</td> <td data-bbox="855 387 1426 622"> <p data-bbox="871 405 1406 517">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="871 539 1398 573">The sixth day of the month is shown as 6.</p> </td> </tr> <tr> <td data-bbox="676 622 855 857">dd</td> <td data-bbox="855 622 1426 857"> <p data-bbox="871 640 1406 752">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="871 775 1414 808">The sixth day of the month is shown as 06.</p> </td> </tr> <tr> <td data-bbox="676 857 855 1048">ddd</td> <td data-bbox="855 857 1426 1048"> <p data-bbox="871 875 1406 943">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="871 965 1190 999">Tuesday is shown as Tue.</p> </td> </tr> <tr> <td data-bbox="676 1048 855 1249">dddd</td> <td data-bbox="855 1048 1426 1249"> <p data-bbox="871 1066 1382 1133">Displays the full name of the day of the week.</p> <p data-bbox="871 1155 983 1189">Tuesday.</p> </td> </tr> </tbody> </table> <p data-bbox="676 1256 1318 1290">The letter d can be either uppercase or lowercase.</p>	d	<p data-bbox="871 405 1406 517">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="871 539 1398 573">The sixth day of the month is shown as 6.</p>	dd	<p data-bbox="871 640 1406 752">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="871 775 1414 808">The sixth day of the month is shown as 06.</p>	ddd	<p data-bbox="871 875 1406 943">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="871 965 1190 999">Tuesday is shown as Tue.</p>	dddd	<p data-bbox="871 1066 1382 1133">Displays the full name of the day of the week.</p> <p data-bbox="871 1155 983 1189">Tuesday.</p>
d	<p data-bbox="871 405 1406 517">Displays the day of the week or month as a number without a leading zero for single-digit days.</p> <p data-bbox="871 539 1398 573">The sixth day of the month is shown as 6.</p>								
dd	<p data-bbox="871 640 1406 752">Displays the day of the week or month as a number with a leading zero for single-digit days.</p> <p data-bbox="871 775 1414 808">The sixth day of the month is shown as 06.</p>								
ddd	<p data-bbox="871 875 1406 943">Displays the day of the week or month as a three-letter abbreviation.</p> <p data-bbox="871 965 1190 999">Tuesday is shown as Tue.</p>								
dddd	<p data-bbox="871 1066 1382 1133">Displays the full name of the day of the week.</p> <p data-bbox="871 1155 983 1189">Tuesday.</p>								
Year (y)	<p data-bbox="676 1339 1174 1373">Displays the year as two or four digits.</p> <table border="1" data-bbox="676 1395 1426 1787"> <tbody> <tr> <td data-bbox="676 1395 855 1630">yy</td> <td data-bbox="855 1395 1426 1630"> <p data-bbox="871 1413 1366 1480">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="871 1503 1414 1570">1999 is shown as 99; 2006 is displayed as 06.</p> </td> </tr> <tr> <td data-bbox="676 1630 855 1787">yyyy</td> <td data-bbox="855 1630 1426 1787"> <p data-bbox="871 1648 1286 1682">Displays the year as four digits.</p> <p data-bbox="871 1704 1046 1738">1999 or 2006</p> </td> </tr> </tbody> </table> <p data-bbox="676 1794 1318 1827">The letter y can be either uppercase or lowercase.</p>	yy	<p data-bbox="871 1413 1366 1480">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="871 1503 1414 1570">1999 is shown as 99; 2006 is displayed as 06.</p>	yyyy	<p data-bbox="871 1648 1286 1682">Displays the year as four digits.</p> <p data-bbox="871 1704 1046 1738">1999 or 2006</p>				
yy	<p data-bbox="871 1413 1366 1480">Displays the year as two digits with a leading zero for years 01 through 09.</p> <p data-bbox="871 1503 1414 1570">1999 is shown as 99; 2006 is displayed as 06.</p>								
yyyy	<p data-bbox="871 1648 1286 1682">Displays the year as four digits.</p> <p data-bbox="871 1704 1046 1738">1999 or 2006</p>								

Parameter	Description								
Hours (h)	<p>A lowercase h bases time on the 12-hour clock.</p> <p>An uppercase H bases time on the 24-hour clock.</p> <table border="1" data-bbox="676 443 1426 1227"> <tbody> <tr> <td data-bbox="676 443 772 640">h</td> <td data-bbox="772 443 1426 640">           Displays the hour without a leading zero for single-digit hours.            9 A.M. is shown as 9.         </td> </tr> <tr> <td data-bbox="676 640 772 837">H</td> <td data-bbox="772 640 1426 837">           Displays hours without a leading zero for single-digit hours on the 24-hour clock.            9AM is shown as 9. 9PM is shown as 21.         </td> </tr> <tr> <td data-bbox="676 837 772 1034">hh</td> <td data-bbox="772 837 1426 1034">           Displays the hour with a leading zero for single-digit hours.            9 PM. is shown as 09.         </td> </tr> <tr> <td data-bbox="676 1034 772 1227">HH</td> <td data-bbox="772 1034 1426 1227">           Displays the hour with a leading zero for single-digit hours on the 24-hour clock.            9 AM. is shown as 09. 9 PM is shown as 21.         </td> </tr> </tbody> </table>	h	Displays the hour without a leading zero for single-digit hours. 9 A.M. is shown as 9.	H	Displays hours without a leading zero for single-digit hours on the 24-hour clock. 9AM is shown as 9. 9PM is shown as 21.	hh	Displays the hour with a leading zero for single-digit hours. 9 PM. is shown as 09.	HH	Displays the hour with a leading zero for single-digit hours on the 24-hour clock. 9 AM. is shown as 09. 9 PM is shown as 21.
h	Displays the hour without a leading zero for single-digit hours. 9 A.M. is shown as 9.								
H	Displays hours without a leading zero for single-digit hours on the 24-hour clock. 9AM is shown as 9. 9PM is shown as 21.								
hh	Displays the hour with a leading zero for single-digit hours. 9 PM. is shown as 09.								
HH	Displays the hour with a leading zero for single-digit hours on the 24-hour clock. 9 AM. is shown as 09. 9 PM is shown as 21.								
Minutes (m)	<p>Displays minutes.</p> <table border="1" data-bbox="676 1330 1426 1720"> <tbody> <tr> <td data-bbox="676 1330 772 1527">m</td> <td data-bbox="772 1330 1426 1527">           Displays minutes without a leading zero for single-digit minutes.  <code>{{TIME \@ "m"}}</code> is shown as 2.         </td> </tr> <tr> <td data-bbox="676 1527 772 1720">mm</td> <td data-bbox="772 1527 1426 1720">           Displays minutes with a leading zero for single-digit minutes.  <code>{{TIME \@ "mm"}}</code> is shown as 02.         </td> </tr> </tbody> </table> <p>The letter m must be lowercase to distinguish minutes from months.</p>	m	Displays minutes without a leading zero for single-digit minutes. <code>{{TIME \@ "m"}}</code> is shown as 2.	mm	Displays minutes with a leading zero for single-digit minutes. <code>{{TIME \@ "mm"}}</code> is shown as 02.				
m	Displays minutes without a leading zero for single-digit minutes. <code>{{TIME \@ "m"}}</code> is shown as 2.								
mm	Displays minutes with a leading zero for single-digit minutes. <code>{{TIME \@ "mm"}}</code> is shown as 02.								

Parameter	Description	
A.M. and P.M. (AM/PM)	am/pm	Displays A.M. and P.M.
	AM/PM	Displays A.M. and P.M. as uppercase. {{NOW \@ "h AM/PM"}} and {{NOW \@ "h am/pm"}} are both shown as 9 AM.

### Number Fields

When Conga Composer populates a numeric value, the digit grouping symbol (the “thousands separator”) is defined according to your Locale settings (as defined in Salesforce’s Setup > Personal Information) or according to the &Culture or &CurrencyCulture Conga Composer parameters.

Despite the cultural settings that govern the output from Conga Composer, within an appearance parameter you must use a comma (",") for the digit grouping symbol (for thousands) and a period (".") as the decimal symbol.

Parameter	Description
0 (zero)	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a 0 (zero).  {{Points \# 00.00}} appears as 09.00.
#	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, nothing appears in that place.  {{Cost \# \$### } appears as \$15.
, (digit grouping symbol)	Separates a series of three digits.  The digit grouping symbol is always entered as a comma into the formatting code, regardless of the country’s cultural preference. Conga Composer will format the output according to the cultural preference.  {{= NetProfit \# \$#,###,###}} appears as \$2,456,800.

Parameter	Description
%, \$, *, and so on	Includes the specified character in the result.  {{= NetProfit \# "##%"}} appears as 33%.
[positive value];[negative value]	Specifies different number formats for results with positive or negative values.  {{Sales95 \# "\$#,##0.00;-#,##0.00"}} Positive values appear as \$1,245.65. Negative values appear as -\$1,245.65.
Currency	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol, and two decimal places  {{OPPORTUNITY_AMOUNT \# Currency}}
Currency0 Also available: Currency1, 2, 3, 4	Formats the value according to the Salesforce Locale, including a currency symbol, digit grouping symbol, and zero decimal places (or 1 decimal place, 2 decimal places, 3, 4, or 5 if using Currency1, Currency2, Currency3, etc.)  {{OPPORTUNITY_AMOUNT \# Currency0}}
Number	Formats the value according to the Salesforce Locale, excluding a currency symbol, but including a digit grouping symbol, and two decimal places.  {{OPPORTUNITY_AMOUNT \# Number}}
Number0	Formats the value according to the Salesforce Locale, including a digit grouping symbol, but excluding two decimal places and a currency symbol.  {{OPPORTUNITY_AMOUNT \# Number0}}  Can also be used with SUM fields.

Parameter	Description
Percent	<p>Formats the value according to the Salesforce Locale, by multiplying the value by 100 and including a digit grouping symbol, two decimal places, and a percent sign.</p> <p>{{OPPORTUNITY_PROBABILITY \# Percent}}</p> <p>Cannot be used with SUM fields.</p>

## Conga Email Template FAQs

This article lists frequently asked questions about using Conga HTML email templates.

### **Q: Is it possible to add a reusable logo to all of the outbound messages?**

**A:** Yes. Included with Composer is a Conga Email Letterhead which may be customized to include a logo, to modify colors, etc. Changes to this letterhead will be automatically included in the subsequent use of Conga Email Templates.

### **Q: Conga Email Templates are built with HTML. Do I need to know much about HTML?**

**A:** Although a little experience with HTML would be useful, you do not need to be an expert in HTML because of the Conga Advanced Editor. Using HTML templates makes sense for Conga Email Templates because they are easy to format and can include tables.

### **Q: What if the recipient cannot open HTML emails?**

**A:** Conga includes a plain-text version of the email along with the HTML version. Special formatting and tables, however, will not look good in the plain-text version.

### **Q: What best practices can you suggest for developing Conga Email Templates?**

**A:** We have several suggestions that should help streamline Conga Email Templates:

- Keep it simple. An HTML email is not a website.
- Use images sparingly and make sure they are stored in Salesforce or on your own server.
- Use simple tables to align data. A cascading style sheet (CSS) is useful for websites, but many email clients do not work well with CSS.
- If you do use some CSS, include it inline.

### **Q: Is it Possible to use report data or detail data merge fields in the subject line?**

**A:** Only master data merge fields will work in the email subject line of a Conga email template.

### **Q: Can I use IF Statement in Conga HTML email template?**

**A:** No. If statements do not work in a Conga HTML email template.

## Create Merge Fields in a Conga Email Template

This article describes how to create merge fields in HTML email templates.

Follow these steps:

1. Click **Template Builder** in the Conga Composer window.
2. Click the radio button for the Email template type.
3. Click the **Conga Email Templates** tab and create a record.
4. Enter a description and basic information about the new template.
5. (Optional) Save the record, then click **Advanced Editor** to further edit the body of the template.
6. Copy the desired merge fields from the Template Builder.
7. Save the record.

## Advanced HTML Email Templates

This section contains information on advanced template-building techniques and tricks.

### Conga Email Template Special Merge Fields

Conga Email Templates have a rich variety of tools, which are similar to the features available in Conga templates built in Microsoft Word.

The following features are available:

- Merge data fields from the Master, Org and User datasets. For more information, see [Create a Conga Email Template](#).
- Merge Report and Query fields using Table- and Paragraph-based Detail Regions.
  - For example Word templates, create a table and include a `{{TableStart:DataSet}}` and a `{{TableEnd:DataSet}}` field to indicate where detail data should be located. Insert merge fields from the detail region in the cells between the TableStart and TableEnd.
  - You cannot add detail data to the Subject line of a Conga Email Template.
- Hide tables based on the absence of rows on a Report or Query using TableHide. Within a table, include a `{{TableHide:DataSet}}` field to indicate that you want the table removed if no data is available in the dataset.
- Calculate the sum of a table column using SUM(ABOVE). Find the sum of a column of numbers by appending a blank row to a table, and creating a SUM(ABOVE) field with `{ =SUM(Above) \# Currency}`

- Display a comma-separated list of Report or Query values using TableList. You can display a list of detail values with `{{TableList:DataSetName:FieldName:Punctuation}}`
- Merge images into Image fields. Insert images using a `{{IMAGE:MergeFieldName}}` merge field.
- Generate Hyperlink fields.
  - Create dynamic hyperlinks by including a `{{HYPERLINK:MergeFieldName}}` field. The field from Salesforce must contain an “absolute” (complete) URL. Optionally, you can include a colon (:) after the merge field name, followed by “friendly” text to display as the clickable link, rather than simply merging the URL. For example: `{{HYPERLINK:ACCOUNT_WEBSITE:Click Here}}`
  - To merge images within a Conga Email Template in Lightning, such as a company logo, users must upload the image as a Document in Salesforce Classic and check the Externally Available Image field for that Document record.

## Row Number Field in HTML Email Template

If you want to number table rows in an email, you can use the ROW\_NUMBER field. This field numbers each line item in a detail dataset and is included by default in every report or query dataset.

Insert the ROW\_NUMBER field into any detail region in the template to number the results in the merged output file.

In the below example, the `{{ROW_NUMBER}}` merge field is placed in the far left column:

The merged output file looks like this:

Item Number	Product Name	Quantity	Total Price
1	GenWatt Gasoline 300kW (Products)	1	\$35,000.00
2	GenWatt Gasoline 750kW (Products)	1	\$75,000.00
3	Installation: Industrial - Low (Installation)	1	\$20,000.00
4	SLA: Gold (Services)	1	\$30,000.00

## PDF Templates

You can merge data from Salesforce into a formatted Portable Document Format (PDF) document that can contain images, text, and tables.

Select one of the following topics for more information:

- [PDF Template Basics](#)
- [Advanced PDF Templates](#)

## PDF Template Basics

Basic information for creating PDF templates.

## PDF Template Guide

Overview of Composer PDF templates

## Learn PDF Template Basics

- [Designing PDF Merge Templates](#)

## Create PDF Templates


- [Create Merge Fields in PDF Templates](#)
- [Creating a Checkbox or Radio Button Field in PDF Templates](#)
- [Merging Detail Data in PDF Templates](#)
- [Refine Merge Fields with Parameters in PDF Templates](#)
- [Formatting Numeric Values in PDF Templates](#)
- [Formatting Date Values in PDF](#)
- [Row Number Field in PDF Templates](#)
- [Use DocuSign Anchor Tags with PDF Forms](#)
- [Watermark Special Merge Field in PDF Templates](#)

## Troubleshoot PDF Templates

- [PDF Template FAQs](#)
- [Product Data Limits](#)

## Designing PDF Merge Templates

A PDF file that contains one or more merge fields is a PDF template. Conga Composer can merge data into a PDF template.

 PDF templates only work with Conga Composer, not Conga Mail Merge.



## What Exactly Is a PDF Template?

It's a PDF file that contains one or more merge fields. It's important to note that the term "PDF template" does not mean taking a Word or Excel template, merging data to it and then converting it to PDF (which Conga Composer does quite nicely).

No, we're talking about starting with a PDF – usually a form designed by a graphic artist or a government agency – to which a layer of PDF fields are added. Common uses for this kind of merge template are US Government W-9 Employment forms and bank loan application forms.

So, if you have a PDF to which you'd like to add fields and so you can merge data, you're in the right place. We'll examine how to create and edit PDF fields.

If you're using Adobe eSign to get your documents signed, you should know you can use PDF merge templates to gather information from your signers, determine the position of signatures in the document and more. See [Adobe eSign integration parameters](#) for more information.

## Before You Get Started

You also need software to edit PDF templates such as Adobe Acrobat Standard 9.

Adobe LiveCycle PDFs are not supported because LiveCycle modifies merge field names.

While rights can be granted to Adobe Reader users (or are standard rights for Adobe Acrobat users) to allow for saving of data in fillable or non-fillable PDF forms, these capabilities are not supported by Conga Composer. Please see this article, "Enable Reader users to save form data," on the Adobe support site for further details on these Adobe rights. Users will not be able to edit the PDF form in any way once it has been generated via the Conga Composer merge process.

## Create Merge Fields in PDF Templates

How to add merge fields to your PDF template.

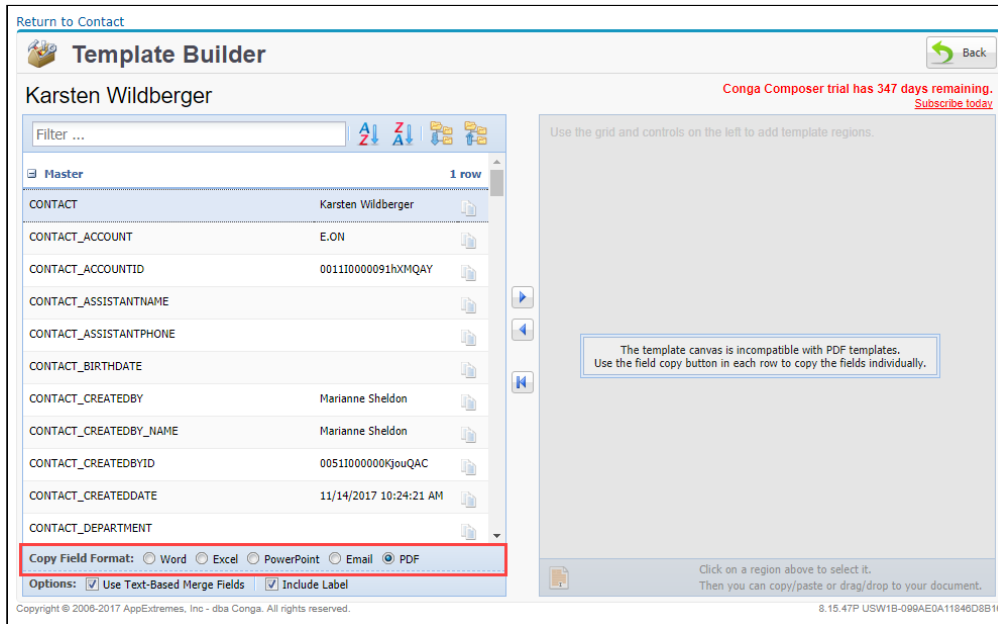
A PDF merge field is a Text field, Checkbox field, or Radio Button field that exists as a layer of fields "on top" of a PDF. The field name consists of the dataset name, a period (full stop) and the field name, as exemplified below.

Example: `Master.CONTACT_FULLNAME`

To add merge fields to a PDF template:

1. Open the [Template Builder](#).

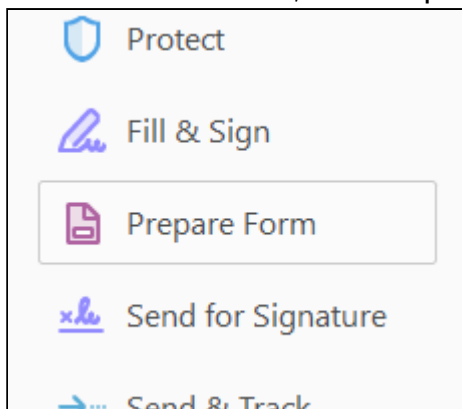
2. In the Copy Field Format area, select PDF.



3. Open a PDF to use as your template. A fictional registration form in Adobe Acrobat is used as an example below.



4. From the Tools menu, click Prepare Form.

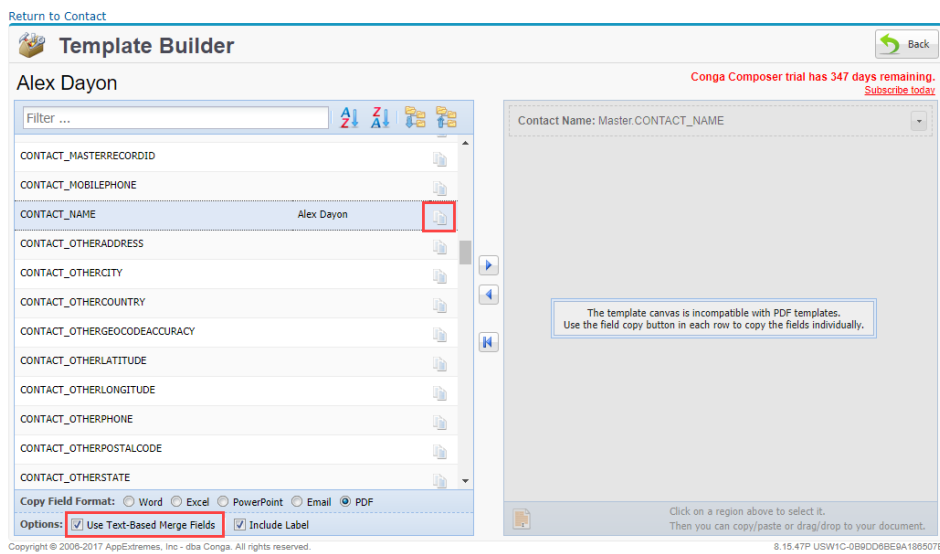


5. Select your file or scan a document, then click **Start**.

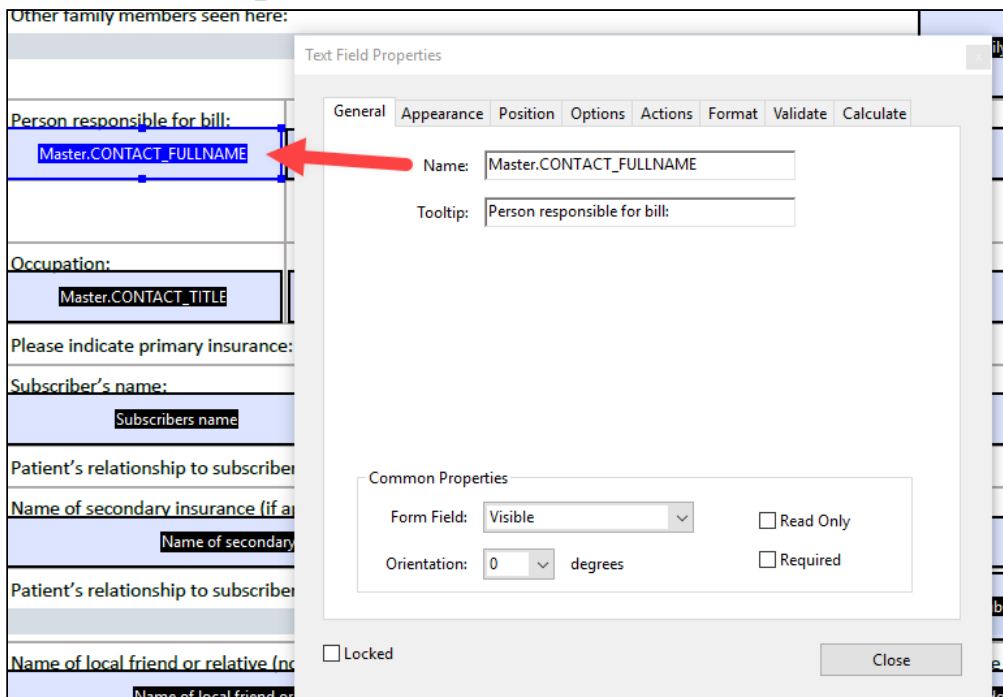
- Once the document is finished processing, you may need to edit the size and shape of the form fields:

- The next step is copying the field names from the Template Builder and replacing the form field labels in our PDF.
6. Return to the Template Builder and select the field you want to merge into the document. The Use Text-Based Merge Fields option has no bearing on PDF so may remain checked or unchecked.
- The copy tool may not work in your browser when PDF is selected for Copy Field Format. Workarounds include manually entering the merge field syntax in your PDF template or selecting Excel for Copy Field Format and removing the

prepended "&=" syntax when pasting the syntax into the PDF template.



7. Double-click the text field in the PDF template to open the Text Field Properties dialog and, in a Name field, paste the CONTACT\_NAME field name copied from the Template Builder.
  - Notice that Conga Composer creates the field with the correct naming convention for PDF merge fields (example: DatasetName.FieldName) - Master.CONTACT\_NAME.



8. Repeat steps 6 and 7 above to add additional fields.

(Please give your insurance information)		
Person responsible for bill: <code>Master.CONTACT_FULLNAME</code>	Birth date: <code>Master.CONTACT_BIRTHDATE</code>	Address (if different): <code>Master.CONTACT_ADDRESS</code>
Occupation: <code>Master.CONTACT_TITLE</code>	Employer: <code>Master.CONTACT_ACCOUNT</code>	Employer address: <code>Master.CONTACT_EMPLOYER_ADDRESS</code>
Please indicate primary insurance: [Choose an item]   Other: <code>Master.CONTACT_OTHER_INSURANCE</code>		


9. Polish the look of your form by resizing the fields to a consistent size and aligning fields.
- Acrobat allows you to select multiple fields by ctrl-clicking each field.
  - Resize and alignment options are available by right-clicking on any of the selected fields.
10. Save the template and return to the Composer User Interface.
11. Run a test merge with the local template file to check that your fields were populated correctly.



Person responsible for bill: <b>Alex Dayon</b>	Birth date: 10/18/1974	Address (if different): 1310 S. Pitkin Ave
---	---------------------------	---


- ✔ It is recommended to use the same page orientation for all of your pages within your PDF template. If the template contains a mix landscape and portrait orientation, the merge fields can merge sideways and display incorrectly when printing the output document.

## Formatting Date Values in PDF

The following date formatting codes are used in conjunction with the "Date" parameter.

Parameter	Description
Month (M)	<p data-bbox="676 338 932 367">Displays the month.</p> <p data-bbox="676 376 919 405"><b>Formatting Codes:</b></p> <ul data-bbox="730 434 1385 1039" style="list-style-type: none"><li data-bbox="730 434 1385 584">• M<ul data-bbox="815 472 1385 584" style="list-style-type: none"><li data-bbox="815 472 1385 546">• Displays the month as a number without a leading zero for single-digit months.</li><li data-bbox="815 555 1129 584">• February is shown as 2</li></ul></li><li data-bbox="730 593 1385 743">• MM<ul data-bbox="815 631 1385 743" style="list-style-type: none"><li data-bbox="815 631 1385 705">• Displays the month as a number with a leading zero for single-digit months.</li><li data-bbox="815 714 1150 743">• February is shown as 02.</li></ul></li><li data-bbox="730 752 1385 902">• MMM<ul data-bbox="815 790 1385 902" style="list-style-type: none"><li data-bbox="815 790 1385 864">• Displays the month as a three-letter abbreviation.</li><li data-bbox="815 873 1166 902">• February is shown as Feb.</li></ul></li><li data-bbox="730 925 1385 1039">• MMMM<ul data-bbox="815 963 1385 1039" style="list-style-type: none"><li data-bbox="815 963 1305 992">• Displays the full name of the month.</li><li data-bbox="815 1001 1230 1039">• February is shown as February</li></ul></li></ul> <div data-bbox="676 1106 1426 1227" style="border: 1px solid #f9c77c; padding: 10px;"><p data-bbox="699 1133 1337 1207"> The letter M must be uppercase to distinguish months from minutes.</p></div>

Parameter	Description
Day (d)	<p>Displays the day of the month or the day of the week.</p> <p><b>Formatting Codes:</b></p> <ul style="list-style-type: none"> <li>• d <ul style="list-style-type: none"> <li>• Displays the day of the week or month as a number without a leading zero for single-digit days.</li> <li>• The sixth day of the month is shown as 6.</li> </ul> </li> <li>• dd <ul style="list-style-type: none"> <li>• Displays the day of the week or month as a number with a leading zero for single-digit days.</li> <li>• The sixth day of the month is shown as 06.</li> </ul> </li> <li>• ddd <ul style="list-style-type: none"> <li>• Displays the day of the week or month as a three-letter abbreviation.</li> <li>• Tuesday is shown as Tue.</li> </ul> </li> <li>• dddd <ul style="list-style-type: none"> <li>• Displays the full name of the day of the week.</li> <li>• Tuesday is shown as Tuesday.</li> </ul> </li> </ul> <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p> The letter d can be either uppercase or lowercase.</p> </div>
Year (y)	<p>Displays the year as two or four digits.</p> <p><b>Formatting Codes:</b></p> <ul style="list-style-type: none"> <li>• yy <ul style="list-style-type: none"> <li>• Displays the year as two digits with a leading zero for years 01 through 09.</li> <li>• 1999 is shown as 99; 2006 is displayed as 06.</li> </ul> </li> <li>• yyyy <ul style="list-style-type: none"> <li>• Displays the year as four digits.</li> <li>• Years are shown in full, such as 1999 or 2006</li> </ul> </li> </ul> <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p> The letter y can be either uppercase or lowercase.</p> </div>

Parameter	Description
Hours (h)	<p>A lowercase h bases time on the 12-hour clock.</p> <p>An uppercase H bases time on the 24-hour clock.</p> <p><b>Formatting Codes:</b></p> <ul style="list-style-type: none"> <li>• h <ul style="list-style-type: none"> <li>• Displays the hour without a leading zero for single-digit hours.</li> <li>• 9 A.M. is shown as 9.</li> </ul> </li> <li>• H <ul style="list-style-type: none"> <li>• Displays hours without a leading zero for single-digit hours on the 24-hour clock.</li> <li>• 9 AM is shown as 9. 9 PM is shown as 21.</li> </ul> </li> <li>• hh <ul style="list-style-type: none"> <li>• Displays the hour with a leading zero for single-digit hours.</li> <li>• 9 PM. is shown as 09.</li> </ul> </li> <li>• HH <ul style="list-style-type: none"> <li>• Displays the hour with a leading zero for single-digit hours on the 24-hour clock.</li> <li>• 9 AM. is shown as 09. 9 PM is shown as 21.</li> </ul> </li> </ul>
Minutes (m)	<p>Displays minutes.</p> <p><b>Formatting Code</b></p> <ul style="list-style-type: none"> <li>• m <ul style="list-style-type: none"> <li>• Displays minutes without a leading zero for single-digit minutes.</li> <li>• <code>{{TIME \@ "m"}}</code> is shown as 2.</li> </ul> </li> <li>• mm <ul style="list-style-type: none"> <li>• Displays minutes with a leading zero for single-digit minutes.</li> <li>• <code>{{TIME \@ "mm"}}</code> is shown as 02.</li> </ul> </li> </ul> <div style="border: 1px solid #f0e68c; padding: 10px; margin-top: 10px;"> <p> The letter m must be lowercase to distinguish minutes from months.</p> </div>



Parameter	Description
A.M. and P.M. (AM/PM)	<p><b>Formatting Codes:</b></p> <ul style="list-style-type: none"> <li>• am/pm <ul style="list-style-type: none"> <li>• Displays A.M. and P.M.</li> </ul> </li> <li>• AM/PM <ul style="list-style-type: none"> <li>• Displays A.M. and P.M. as uppercase.</li> <li>• <code>{{NOW \@ "h AM/PM"}}</code> and <code>{{NOW \@ "h am/pm"}}</code> are both shown as 9 AM</li> </ul> </li> </ul>

Here are several common examples of the date appearance parameter. Let's say the field we're formatting is OPPORTUNITY\_CLOSEDATE and that it contains the date 03/03/2008.

Example	Output
<code>Master.OPPORTUNITY_CLOSEDATE(Date MM-dd-yyyy)</code>	03-03-2008
<code>Master.OPPORTUNITY_CLOSEDATE(Date dd/MM/yyyy)</code>	03/03/2008
<code>Master.OPPORTUNITY_CLOSEDATE(Date "MMM dd, yyyy")</code> The format code is enclosed in quotes because it contains spaces	Mar 03, 2008
<code>Master.OPPORTUNITY_CLOSEDATE(Date "MMMM d, yyyy")</code> The format code is enclosed in quotes because it contains spaces	March 3, 2008

### Today's Date:

By creating a merge field with the name Date. Today, Conga Composer will insert the current date. You may format the date using a date-formatting parameter.

## Formatting Numeric Values in PDF Templates

The following numeric formatting codes are used in conjunction with the "Numeric" parameter.

When Conga Composer populates a numeric value, the digit grouping symbol (the "thousands separator") is defined according to your Locale settings (as defined in **Salesforce's Setup → Personal Information**) or according to the Culture or CurrencyCulture Conga Composer parameter.

Despite the cultural settings that govern the output from Conga Composer, within an appearance parameter, you must use a comma (",") for the digit grouping symbol (for thousands) and a period (".") as the decimal symbol.

Parameter	Description
0 (zero)	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a 0 (zero).  <code>Master.OPPORTUNITY_AMOUNT(Numeric 0.00)</code> displays "5280.25".
#	Specifies the requisite numeric places to display in the result. If the result does not include a digit in that place, Word displays a space.  <code>Master.OPPORTUNITY_AMOUNT(Numeric ###0)</code> displays "5280 ".
, (digit grouping symbol)	Separates a series of three digits.  The digit grouping symbol is always entered as a comma into the formatting code, regardless of the country's cultural preference. Conga Composer will format the output according the cultural preference.  <code>Master.OPPORTUNITY_AMOUNT(Numeric #,##0.00)</code> displays "5,280.25".
%, \$, *, and so on	Includes the specified character in the result.
"positive; negative"	Specifies different number formats for positive and negative results.

Here are several common examples of the number appearance parameter. Let's say the field we're formatting is OPPORTUNITY\_AMOUNT and that it contains 5280.25 and the Salesforce Locale is "English (United States)".

Example	Output
<code>Master.OPPORTUNITY_AMOUNT(Numeric #,##0)</code>	5,280
<code>Master.OPPORTUNITY_AMOUNT(Numeric \$#,##0)</code>	\$5,280
<code>Master.OPPORTUNITY_AMOUNT(Numeric \$#,##0.00)</code>	\$5,280.25
<code>Master.OPPORTUNITY_AMOUNT(Currency)</code>	\$5,280.25
<code>Master.ACCOUNT_PHONE(Numeric (###) ###-####)</code>  The data must be entered without additional formatting like spaces or dashes (for example: 6517331110) in the View Data workbook.	(651) 733-1110

Let's change the scenario a bit. Let's assume we're formatting the same field value as above, however, the Salesforce Locale is "German (Germany)".

Example	Output
<code>Master.OPPORTUNITY_AMOUNT(Numeric #,##0)</code>	1. 280
<code>Master.OPPORTUNITY_AMOUNT(Numeric "#,##0 €")</code> The formatting code is enclosed in straight quotes because it contains spaces.	1. 280 €
<code>Master.OPPORTUNITY_AMOUNT(Numeric #,##0.00)</code>	1. 280,25
<code>Master.OPPORTUNITY_AMOUNT(Currency)</code>	1. 280,25 €

## Refine Merge Fields with Parameters in PDF Templates

You can tune Conga Composer's behavior when merging to PDF with field parameters. One of the following parameters may be appended to the end of a merge field within parentheses.

Parameter	Description
Date	<p>Formats a field value as a date, if possible.</p> <ul style="list-style-type: none"> <li>• <b>Requirements:</b> A <a href="#">date formatting code</a>.</li> <li>• <b>Examples:</b> <ul style="list-style-type: none"> <li>• <code>Master.LastModifiedDate(Date MM/dd/yyyy)</code> If the Last Modified Date field contained 03/03/2008 09:05 AM, this parameter would format the date as 03/03/2008</li> <li>• <code>Master.LastModifiedDate(Date "MMMM d, yyyy")</code> If the Last Modified Date field contained 03/03/2008 09:05 AM, this parameter would format the date as March 3, 2008</li> </ul> </li> </ul> <p>The format code in Example 2 is enclosed in quotes because it contains spaces</p>
Currency	<p>Formats a field value as currency according to the Salesforce Locale (or Conga Composer parameter CurrencyCulture), including currency symbol, digit grouping ("thousands separator") and two decimal places.</p> <ul style="list-style-type: none"> <li>• <b>Example:</b> <ul style="list-style-type: none"> <li>• <code>Master.AMOUNT(Currency)</code></li> </ul> </li> </ul> <p>If the Amount field contained 5280.25 and the Locale is set to "English (United States)", this parameter would format the value as \$5,280.25</p>

Parameter	Description
Numeric	<p>Formats a field value as a number, if possible.</p> <ul style="list-style-type: none"> <li>• <b>Requirements:</b> A numeric formatting code (see <a href="#">Formatting Numeric Values in PDF Templates</a>)</li> <li>• <b>Examples:</b> <ul style="list-style-type: none"> <li>• <code>Master.NumberOfParticipants(Numeric #,##0)</code> If the Number of Participants field contained 5280, this parameter would format the value as 5,280</li> <li>• <code>Master.Height (Numeric "#,##0 feet")</code> If the Height field contained 5280, this parameter would format the value as 5,280 feet</li> </ul> </li> </ul> <p>The format code is enclosed in quotes because it contains spaces. For percentage in PDF ensure the merge field is (Numeric ##) and that the format option in PDF is 'percentage'.</p>

- You must enter a space between the parameter name and the format code when using the Date or Numeric parameter.
- Don't enter a comma between the parameter name and the format code.

## PDF Template FAQs

This article lists frequently asked questions about using Conga PDF templates.

### How can I lock the output PDF to prevent it from being printed?

You can use any of the PDF output parameters to enable encryption, printing, copying, and filling in forms.

### Can I merge images from Salesforce into a PDF template?

No. If you need to merge images into a Conga template, use a Word, PPTX, or HTML Email template.

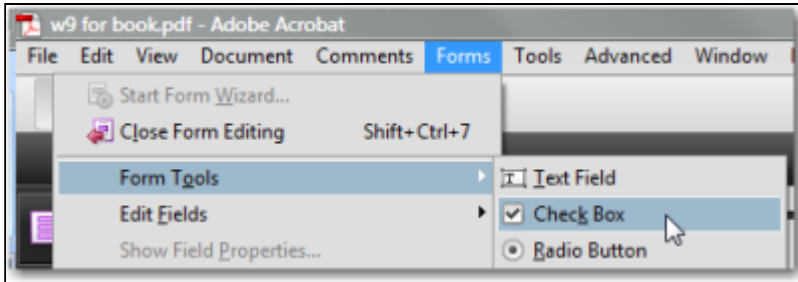
## Advanced PDF Templates

This section contains information on advanced template-building techniques and tricks.

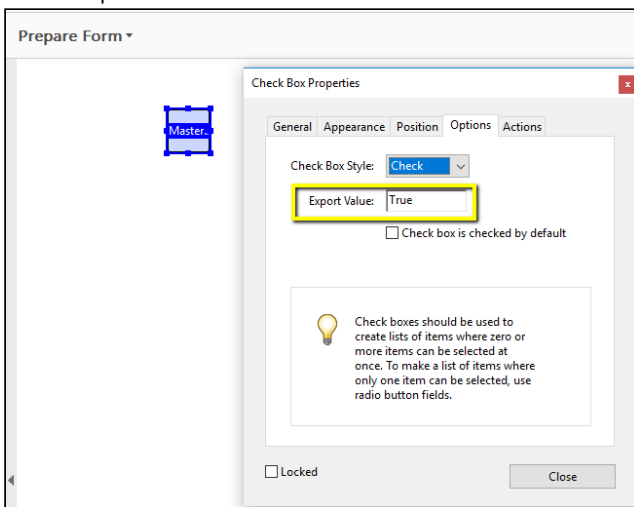
## Creating a Checkbox or Radio Button Field in PDF Templates

In addition to creating Text fields, you can create Checkbox or Radio Button fields in PDF templates. To create Checkbox or Radio Button fields in PDF templates:

1. In a PDF, click **Forms**.
2. Click **Form Tools**.
3. Select **Check Box**. Use the standard field naming conventions.



4. In the Check Box Properties, enter the Field Name from the Template Builder on the General tab.
5. In the Check Box Properties, define the True or False value that indicates when the checkbox field should appear selected in the Export Value field on the Options tab. The Export Value for Salesforce Checkbox fields must be either True or False.



You must have at least 2 radio button fields (both referencing the same merge field name) in order for the value to appropriately merge to the output.

For Mac users: You must have a form that Acrobat recognizes as a fillable form to add a radio button or checkbox. Go to **Prepare Form**. Drag or right-mouse click in the area where you want checkboxes and select checkbox.

## Merging Detail Data in PDF Templates

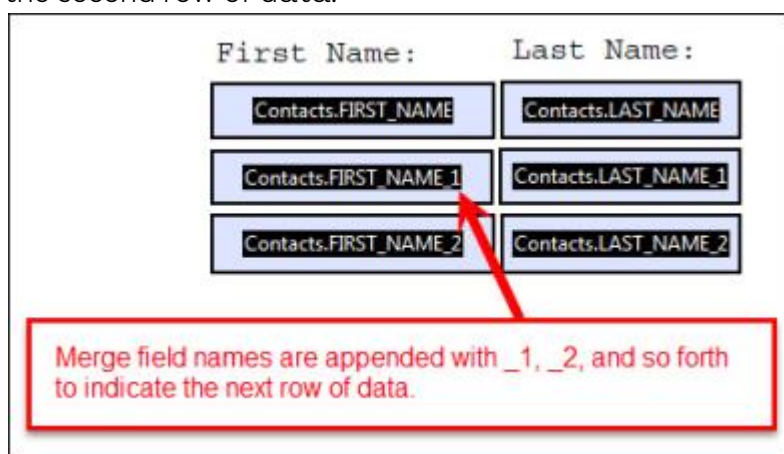
Conga Composer has the unique ability to use data from a Salesforce Report or SOQL query as detail data (e.g., invoice line items for each invoice).

PDF templates, however, have no way to dynamically expand to accommodate an unlimited number of detail rows. (If you need that flexibility in handling detail data, consider using a Word-, Excel-, PowerPoint-, or HTML Email-based template which support dynamic detail data.)

With PDF templates, you may lay out a fixed number of rows to accommodate detail data by creating fields in the format `DatasetName.FieldName`, `DatasetName.FieldName_1`, `DatasetName.FieldName_2`, and so on.

To merge Salesforce data into a PDF template:

1. Open the Template Builder window and copy the field to the clipboard.
2. Decide how many rows of data you want to accommodate in your PDF template. Because PDF templates cannot dynamically add rows of detail data as Word, Excel, PowerPoint, and HTML Email templates do, you'll need to build a row of merge fields to accommodate each result/record in your report or query.
3. Switch to your PDF program and create a merge field for the field you copied to the clipboard in Step 1. This field will represent the field from the first row of data in your report/query.
4. Create another merge field, perhaps on the next line down on your template. This field will represent the field from the second row of data. The name of the field must be the same as the previous field, except with an "\_1" appended to the end to indicate the second row of data.



5. Repeat the creation of merge fields, each time incrementing the suffix number, e.g. "\_2", then "\_3", and so on.

Consider this data from a Conga Composer View Data workbook. Here, our Contacts report (note the Contacts dataset worksheet) has retrieved three Contacts related to our Account:

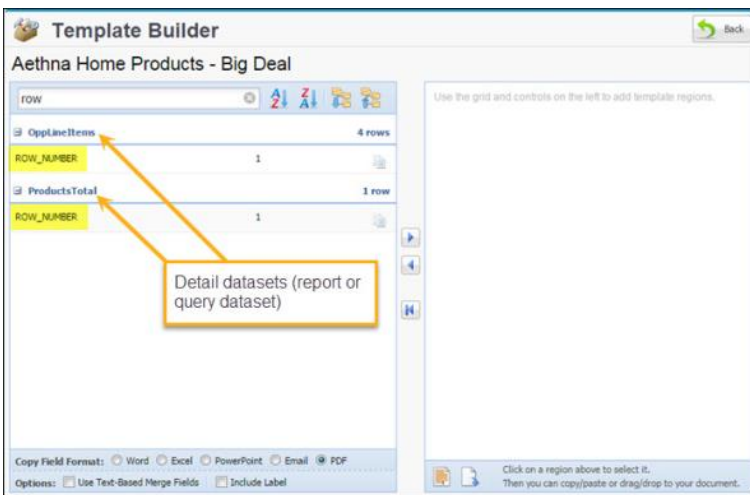
Salutation	First Name	Last Name	Title	Mailing Street	Mailing City	Mailing State
Ms.	Rose	Gonzalez	SVP, Procurement	313 Constitution Place	Austin	TX
Mr.	Sean	Forbes	CFO	312 Constitution Place	Austin	TX
Mr.	Elias	Anderson	SVP	312 Constitution Place	Austin	TX

The final output, after merging to the template created above, would look like this:

First Name:	Last Name:
Rose	Gonzalez
Sean	Forbes
Elias	Anderson

## Row Number Field in PDF Templates

The Row Number field numbers each line item in a detailed dataset and is included by default in every report or query dataset. The field name is ROW\_NUMBER.



Insert this field into any detailed region in the template to number the results in the merged output file. In the below example, the `{{ROW_NUMBER}}` merge field is placed in the far left column:



Item Number	Product Name	Quantity	Total Price
OppLineItems.ROW_NUMBER	OppLineItems.OPPORTUNITY_LINEITEM_PROD UCTNAME	OppLineItems.OPPORTUNITY_LINEITEM_QUANTITY	OppLineItems.OPPORTUNITY_LINEITEM_TOTALPRICE
OppLineItems.ROW_NUMBER_1	OppLineItems.OPPORTUNITY_LINEITEM_PROD UCTNAME_1	OppLineItems.OPPORTUNITY_LINEITEM_QUANTITY_1	OppLineItems.OPPORTUNITY_LINEITEM_TOTALPRICE_1
OppLineItems.ROW_NUMBER_2	OppLineItems.OPPORTUNITY_LINEITEM_PROD UCTNAME_2	OppLineItems.OPPORTUNITY_LINEITEM_QUANTITY_2	OppLineItems.OPPORTUNITY_LINEITEM_TOTALPRICE_2

## Disable {{ROW\_NUMBER}}

To disable the {{ROW\_NUMBER}} field in all detail datasets use the parameter `&DisableRowNums=1`. This parameter prevents the {{ROW\_NUMBER}} field from displaying in the report, query, Opportunity Line Items, or Quote Line Items datasets.

The merged output file looks like this:

Item Number	Product Name	Quantity	Total Price
1	GenWatt Diesel 10kW	1	5000
2	GenWatt Propane 100kW	2	30000
3	GenWatt Diesel 200kW	5	125000
4	GenWatt Gasoline 300kW	3	105000

## Use Conga Sign Signature Tags with PDF Files

This article describes how to add Conga Sign signature tags to Conga Sign PDF file. Use this method to send a PDF file for signature with Conga Sign. These instructions are specific to Adobe Acrobat X Standard.

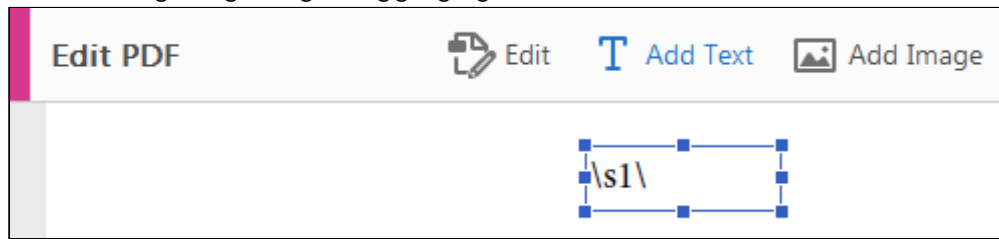
Add Conga Sign tags to a PDF file for use with Conga Sign and Conga Sign-integrated Composer solutions. This method allows users to utilize pre-tagged PDF files as Conga Sign documents or Conga Composer Templates.

**Both Conga Composer merge fields and Conga Sign automatic anchor tags populate when the template is sent for signature using this method. These instructions are specific to Adobe Acrobat X Standard.**

To add Conga Sign signature tags to a PDF template (using Adobe Acrobat X Standard):

1. Create a PDF file or open an existing PDF file.
  - (Optional) Add Composer merge fields to the PDF document.
    - For more information on creating a Composer PDF template, see [PDF forms reference guide](#).
2. Open the PDF file in Adobe Acrobat, click **Tools**, and then click **Edit**.
3. Click **Add Text** and select an area in the PDF template to place a Conga Sign tag.

4. Add a Conga Sign tag's tagging syntax to the text field.



- For a list of all Conga Sign tags and their respective tagging syntax, see [List of Conga Sign signature tags](#).
5. (Optional) Repeat Steps 3 and 4 to additional Conga Sign tags.
  6. Save the PDF template.

## Use DocuSign Anchor Tags with PDF Forms


This article will describe the process of adding DocuSign automatic anchor tags to a PDF form template for use with Conga Composer. Using this method, both Conga merge fields and DocuSign automatic anchor tags will populate when the template is sent for signature. These instructions specific to Adobe Acrobat X Standard.

You can add DocuSign automatic anchor tags to a PDF form template for use with Conga Composer. Using this method, both Conga merge fields and DocuSign automatic anchor tags populate when the template is sent for signature. These instructions are specific to Adobe Acrobat X Standard.

To add DocuSign signature tags to a PDF form (using Adobe Acrobat X Standard):

1. Create a PDF form template with Conga Composer merge fields. Detailed information about how to use PDF forms with Conga Composer is available in the [PDF forms reference guide](#). When the PDF form template with Composer merge fields is created, you can add DocuSign anchor tags.
2. Open your PDF form in Adobe Acrobat and select **View** → **Tools** → **Forms**.
3. Click **Edit** in the list of options in the right column.
4. Determine the place in your form where you want to add a DocuSign anchor tag, then select **Add New Field**. Choose a Text Field.
5. Enter a placeholder value for Field Name. The value can be whatever you like. Unlike Composer merge fields, the field name does not map a Salesforce field to the form field. The value is a placeholder to name the form field.
6. Now add your DocuSign anchor tag. Click **Preview** in the top menu to return to the standard view of your PDF form.
7. Place your cursor in your new form field and manually type your DocuSign anchor text string.

8. Return to Edit mode (same place as the Preview button). Select your form field, click **All Properties**, then select the **Appearance tab**. Change the text color for this form field to white. Changing the color to white ensures your DocuSign anchor text string is not shown to your eventual signer and they are only presented with the relevant DocuSign tag. Repeat this for every DocuSign anchor tag in your template.
9. Save your form and upload your template to Salesforce.

 In your Conga Composer button URL, add the `&FlattenPDFForms=1` parameter. This is **REQUIRED** in order for anchor tags on a PDF form to be recognized by DocuSign.

## Watermark Special Merge Field in PDF Templates

You can add a watermark to PDF documents by using a special merge field. Here are the specifications.

Depending upon the length and complexity of the Watermark, this functionality may increase the file size of the output file by up to 50%.

### Watermark Merge Field Format

To specify custom values, use this format:

`WATERMARK:DATASET.FIELDNAME:FONT:FONTSIZE:ROTATION:OPACITY:COLOR`

To use default values (see table below), use this format: `WATERMARK:DATASET.FIELDNAME`

A total of three lines of text are available in a PDF Watermark. To add a line break to your Watermark, use the web encoded syntax for a line break: `"%0D%0A"`

Field	Default Value
Font	Helvetica
Font Size	72
Rotation	52 Value must be between -90 and 90.
Opactiy	70% Value must be between 1 (fully transparent) and 100 (opaque).

Color	<p>Gray</p> <p>Value can be a hex value or a color from <a href="#">this Known Color Enumeration list</a>.</p>
-------	--

### Allowed Fonts:

Courier	Helvetica-BoldOblique
Courier-Bold	Helvetica-Oblique
Courier-BoldOblique	Times-Roman
Courier-Oblique	Times-Bold
Helvetica	Times-BoldItalic
Helvetica-Bold	Times-Italic

## Conga Composer Parameter Guide

You can customize the Conga Composer Solution by adding parameters to the button URL. This guide contains information on how to work with parameters, parameter categories, and parameters that you can use for customization.

### Parameter Basics

- [A Few Rules About Parameters](#)
- [Adding Parameters to a Solution](#)
- [How Parameters are Formatted](#)

### Parameter Categories

- [Activity Logging Parameters](#)
- [Adobe PDF Template Parameters](#)
- [Adobe eSign Integration Parameters](#)
- [Advanced Control Parameters](#)

- [Advanced Data Formatting Parameters \(Locale and Culture Parameters\)](#)
- [Automation-Background Mode Parameters](#)
- [Bigtincan Integration Parameters](#)
- [Box Integration Parameters](#)
- [Branding Parameters](#)
- [Chatter Integration Parameters](#)
- [Conga Appearance Parameters](#)
- [Conga Contracts Parameters](#)
- [Conga Email Template Parameters](#)
- [Conga Sign Integration Parameters](#)
- [Data Gathering Parameters](#)
- [DocuSign Integration Parameters](#)
- [Email Parameters](#)
- [Excel Parameters](#)
- [Google Drive Parameters](#)
- [OneDrive Parameters](#)
- [Output Parameters](#)
- [Powerpoint Parameters](#)
- [Salesforce Content Integration Parameters](#)
- [Salesforce1 Integration Parameters](#)
- [Sertifi Integration Parameters](#)
- [SpringCM Parameters](#)
- [Template Selection Parameters](#)
- [Word Parameters](#)

## A Few Rules About Parameters

The Conga Composer Solution can be customized by adding parameters to the button URL. Parameters are added using the following convention:

`&ParameterName=Value`

To learn more about how to add parameters to the button URL, see [Adding Parameters to a Solution](#). For the complete list of parameters available to add to a solution, see the [Parameter Guide](#).

## Tips and Tricks about Parameters

Below are a few best practices to adhere to when customizing your solution with parameters:

- Spaces are not permitted anywhere in URLs, including parameter/value pairs. If the parameter value requires spaces, replace each space with a plus sign (+).
- Quote marks (""") are not permitted anywhere in URLs.
- Parameter names are not case-sensitive but values may be, so the best practice is to observe the case closely.
- Parameters may be appended to the URL in any order.
- Some parameters require other parameters in order to function. (For example, to have the solution automatically log activity as it runs, you must include &ACO=1 to turn on activity logging, and then use other activity-logging parameters to specify what should be logged where.)
- The entire URL cannot exceed 1750 characters.

## Adding Parameters to a Solution

Add parameters to any solution:

- Using the Solution Manager
- Using the Document Automation Wizard
- Manually entering parameters and parameter values to the solution's button URL

## Using Solution Manager to Add Parameters

You can add multiple parameters to a solution within the Solution Manager. The Parameter Picker within Solution Manager presents all available parameters and provides detailed information about the accepted parameter values and each parameter's functionality.

For more information, see the [Conga Solution Parameter Screen](#).

## Manually Adding Parameters to a Button URL

- For solutions not created in or managed by Solution Manager, you can enter parameter value pairs directly into the button URL.
- For more information about parameter guidelines, see [Before You Start: A Few Rules About Parameters](#).
- For formatting examples, see [How Parameters are Formatted](#).

# How Parameters are Formatted

Format Parameters in a Conga Solution.

## When Using Solution Manager

When you add parameters using the Customize with Parameters button, the Solution Manager automatically formats the parameters and value pairs as it builds the button URL. In some cases, it also selects default values.

## When Building or Editing Button URLs Manually

When you build button URLs manually, you must add parameters as a parameter-value pair in this format:

```
&Parameter=value
```

## Why Is the Ampersand (&) Necessary?

The button URL is essentially a set of instructions. The ampersand tells that what follows is an instruction set and text after the equals sign tells what value to apply to it. For instance, the parameter value pair `&AC0=1` tells Composer that activity logging (AC0) should be enabled (1).

Part of the power is that you can add multiple parameters to instruct to perform multiple tasks within the same solution. Each parameter value pair begins with an ampersand, letting know that each is a new instruction set.

You can add several parameters to your solution's button URL. Additional parameters within a URL are included without spaces between them. You can add paragraph returns after each parameter to make it easier for you to read

```
&Parameter1=value1  
&Parameter2=value2  
&Parameter3=value3
```

When submits the button URL to run your solution, it removes all paragraphs to format the URL into one string.

```
&Parameter1=value1&Parameter2=value2&Parameter3=value3
```

Throughout this Support website, parameter value tables follow the format below, although some parameter entries may not include prerequisites, notes, or examples. If only a specific range of values is allowed, the entry includes an acceptable values table.

Activity Logging Parameter ✔ Release 8 ✘ Release 7

---

3 **AC2**

---

4 Parameter Value: [0 or 1]

5 \* Requires: &AC0=1

**USAGE**

---

6 \* &AC0=1  
&AC2=1

---

7 **Save Text**  
Determines whether to save the text of a Word document (post-merge) to the body of the Activity (Task).  
📌 This feature works only with Microsoft Word templates.

Value	Definition
0	Save text is set to none.
1	Save text is set to Document Text (Word only). (Default)

8



1. The parameter category, which is simply a way to group parameters by function.
2. Shows whether you can use the parameter with a specific release.
3. The parameter.
4. Acceptable parameter values.
5. Requirements. Some parameters require other parameters or settings to function.
6. An example of the parameter and value pair in use. If a parameter requires another to function, the required parameter is shown in the example. If you can also use dynamic field content, that example is shown in the Definition area.
7. The parameter definition, containing the name of the parameter and a description of what it does or how to use it. Includes any notes and warnings, as well as additional examples.
8. If the parameter accepts specific values (such as numbers or IDs), a value table appears with the values and description.

## Activity Logging Parameters

The following parameters affect the Log Activity Options of the dialog box.

### Description

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

### Parameters

- [AC0](#) enables automatic activity logging.
- [AC1](#) sets the activity logging subject.
- [AC2](#) saves the text of a Word document into the body of the activity.
- [AC3](#) sets a task follow-up date.
- [AC5](#) saves HTML email markup to an Activity related list on the Salesforce Object without the need to send an outbound email.
- [AttachmentParentID](#) specifies a record in Salesforce that will contain the output file.
- [DocFolderId](#) specifies the Salesforce Documents folder in which the merged output file is stored.
- [DS3](#) enables or disables the Activity Logging area.
- [LG0](#) determines whether to store the text of the document (post-merge) to the body of the Task.
- [LG1](#) sets the Subject to the text specified.
- [LG2](#) sets the follow-up date by using the integer to add days to today's date.
- [LG3](#) sets the file attachment options.
- [LG4](#) specifies whether to automatically log an activity after merging.
- [LG5](#) sets the Assigned To field in the activity.

- [LG6](#) sets the Assigned To field in the follow-up activity.
- [LG7](#) sets the Activity Subject in the follow-up activity.
- [LGAttachOption](#) sets the Salesforce object to which attachments generated during Activity Logging are stored.
- [LGParentID](#) overrides the default record on which the task is created.
- [LGWhoID](#) specifies the WhoID.
- [MFTS0](#) the API Field Name of the field to be set as part of the activity logging.
- [MFTS1](#) the API Field Name of the field to be set as part of the activity logging.
- [MFTS2](#) the API Field Name of the field to be set as part of the activity logging.
- [MFTSId0](#) sets a field on a record other than the master object.
- [MFTSId1](#) sets a field on a record other than the master object.
- [MFTSId2](#) sets a field on a record other than the master object.
- [MFTSValue0](#) specifies a value for a Master Field.
- [MFTSValue1](#) specifies a value for a Master Field.
- [MFTSValue2](#) specifies a value for a Master Field.
- [SC0](#) saves a copy of the merged output file.
- [SC1](#) specifies where to save a copy of the merged output file.
- [UF0](#) updates a defined Master Field to Set field.

## AC0

Automatic Activity Logging. Specifies whether to automatically log an activity after merging.

**Category:** Activity Logging Parameter

**Compatible with:** Conga Composer 8

**Parameter Name:** AC0

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** AC0 provides Automatic Activity Logging. This parameter specifies whether to automatically log an activity after merging.

Parameter Value	Definition
0	Automatic Logging is disabled. (Default)
1	Automatic Logging is enabled.

**Example:**

```
&AC0=1
```

## AC1

Sets the Subject to the text specified. Also used as the subject of a Follow-Up Task, if one is created.

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** AC1

**Parameter Value:** [text]

**Parameter Requirements:** [&AC0=1](#)

**Parameter Description:** AC1 provides an Activity Subject.

The parameter sets the Subject to the text specified. Also used as the subject of a Follow-Up Task, if one is created. If not specified and the Activity Subject is blank, the output file name is used for the Activity Subject.

You can also use dynamic field content for the parameter value: `&AC1={!Opportunity.EmailSubject}`

 Do not use the less than (<) or greater than (>) symbol with this parameter.

**Example:**

```
&AC0=1
```

```
&AC1=Quote+Generated
```

## AC2

Determines whether to save the text of a Word document (post-merge) to the body of the Activity (Task).

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** AC2

**Parameter Value:** [0 or 1]

**Parameter Requirements:** [&AC0=1](#)

**Parameter Description:** AC2 saves the text of a merged Word document to the body of an Activity (Task)

 This parameter works only with Microsoft Word templates.

Parameter Value	Definition
0	Save text is set to none.
1	Save text is set to Document Text (Word only). (Default)

**Example:**

```
&AC0=1
&AC2=1
```

## AC3

Follow-Up Date. Indicates the number of days in the future in which a Follow-Up Task should be created by using the integer to add days to the current date

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** AC3

**Parameter Value:** [integer]

**Parameter Requirements:** [&AC0=1](#)

**Parameter Description:** AC3 sets the Follow-Up Date on a Task. The parameter indicates the number of days in the future in which a Follow-Up Task should be created by using the integer to add days to the current date.

**Example:**

```
&AC0=1
&AC3=14
```

## AC5

Save HTML email markup to an Activity related list on the Salesforce Object without the need to send an outbound email.

**Category:** Activity Logging Parameter


**Compatible with:** Composer 8

**Parameter Name:** AC5


**Parameter Value:** (0 or 1)

**Parameter Requirements:** None

**Parameter Description:** AC5 saves HTML email markup to an Activity related list on the Salesforce Object without the need to send an outbound email.

 This parameter only works if you do not send an outbound email.

Parameter Value	Definition
0	Does not save HTML markup to Activity relates list on an Object.
1	Saves HTML markup to Activity related list on an Object. (default)

 AC5 is not compatible with &DS7=2. When using these two parameters together, the following error occurs: *There was an issue getting the Master object record details.* To accomplish email activity logging in background mode, this parameter must be used: DS7=12. To avoid this error, use &DS7=12 instead of &DS7=2.

**Example:**

&AC5=1

## Adobe PDF Template Parameters

### Description

The following parameters affect the behavior of Adobe PDF templates.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

### Parameters

- [FlattenPDFForms](#) specifies whether to flatten PDF form fields.
- [FPO](#) locks the output file format.
- [PDFEF](#) specifies whether to embed fonts in the PDF document.
- [PDFIQ](#) specifies the image quality of the PDF document.

- [PDFQR](#) specifies the rendering speed of the PDF document.
- [PS0](#) specifies whether the PDF document is encrypted.
- [PS1](#) specifies whether printing is allowed for a PDF document.
- [PS2](#) specifies whether a reader can copy content on the PDF.
- [PS3](#) specifies whether a document allows readers to complete forms.
- [PS4](#) specifies whether screen readers can access the PDF document.
- [PSPW](#) specifies the password to use for a locked PDF.

## Adobe Sign Integration Parameters


### Description

Each of these parameters affects the field specified on an Adobe eSign Agreement record.

Adobe Sign delivers documents through email, much like Salesforce itself, but adds the ability to acquire, track, and file signatures.

Adobe Sign allows you to use PDF forms to collect data from your signers, position signature fields in your documents, collect two-party information from both sender and signer and more. You can add Adobe Sign-specific form fields to PDF merge templates to enable those features.

Adobe Sign is available separately on the AppExchange. You must install the Adobe Sign package to use these integration parameters.

 After installing Adobe Sign from the AppExchange, you must modify the Adobe Sign application by disabling the default value of the Signature Order field. For more information on disabling the Signature Order field, see [Modify the Adobe eSign Application to Disable the Signature Order Field](#).

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

Each of the following parameters affects the field specified on an Adobe Sign Agreement record. This integration requires Adobe Sign for Salesforce, version 6.0 or higher.

### Parameters

- [ESAccountID](#) specifies the Adobe eSign Agreement account ID.
- [ESAdditionalTo](#) sets the value of the Additional Recipients field.
- [ESAgreementName](#) specifies the Adobe Sign agreement name.
- [ESAgreementTemplateID](#) specifies the Adobe Sign Agreement Template to use.

- [ESContactId](#) specifies the value of the Contact field.
- [ESContactId1](#) specifies the value of the Contact field.
- [ESContactId2](#) specifies the value of the Contact field.
- [ESContactId3](#) specifies the value of the Contact field.
- [ESContractId Adobe Sign Integration Parameters](#) specifies the value of the Adobe Sign Contract.
- [ESCopyTo](#) specifies the Copy To email address.
- [ESCustomField](#) specifies the name of the custom field on the Adobe Sign agreement record.
- [ESCustomFieldValue](#) specifies the value of the custom field.
- [ESDataMappingId](#) specifies the Adobe Sign Data Mapping record.
- [ESExpireInDays](#) specifies the number of days before the email expires.
- [ESLanguage](#) To set the Language field.
- [ESLeadId](#) To set the value of the Lead field.
- [ESLeadId1](#) To set the value of the Lead field.
- [ESLeadId2](#) To set the value of the Lead field.
- [ESLeadId3](#) To set the value of the Lead field.
- [ESMessage](#) To set the value of the Message field.
- [ESOpportunityId](#) specifies the value of the Opportunity ID field.
- [ESPreview](#) To activate the Adobe Sign preview field.
- [ESRecipient1](#) specifies Adobe Sign recipients.
- [ESRecipient1Role](#) specifies roles for Adobe Sign recipients.
- [ESRemindRecipient](#) To remind an Adobe Sign recipient to sign.
- [ESRequirePWSign](#) To require an Adobe Sign recipient to enter a password to sign.
- [ESRequirePWView](#) To require a password to view the email.
- [ESRequireSelfSign](#) specifies that the sender must also sign.
- [ESSignatureOrder](#) specifies the order in which recipients sign.
- [ESSignatureType](#) specifies the signature type.
- [ESSignatureGroup](#) specifies the signature group.
- [ESSVBA](#) sends an Adobe Sign agreement as a background action.
- [ESVisible](#) enables you to use Adobe Sign functionality.

## Advanced Control Parameter

### Description

These Advanced Control Parameters allow you to enable and disable Conga for certain profiles and solutions.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution, A Few Rules About Parameters, and How Parameters are Formatted](#).

## Parameters

- [DC](#) disables the execution of a solution if the contents do not meet the criteria you set.
- [DCL](#) specifies the text to display when disabling Conga execution.
- [EC](#) enables the execution of a Conga solution providing specific criteria is met.
- [ECL](#) specifies the text to display when enabling Conga execution.
- [OCNR](#) specifies whether the Salesforce record is refreshed when Closed is clicked.
- [QueryStringField](#) allows you to reference a Salesforce formula field in the merge process.
- [QueryStringField1](#) allows you to reference a second Salesforce formula field in the merge process.
- [QueryStringField2](#) allows you to reference a third Salesforce formula field in the merge process.
- [ReturnPath](#) specifies where to refresh after merging.
- [ContentNetworkID](#) specifies the Network Id for the appropriate community to ensure the file record is created and shared within Salesforce when saving a Salesforce file as a community user.

# AmazonS3Visible

Enables Amazon S3 as an output option on the Conga Composer user interface.

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** AmazonS3Visible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &SC1=AmazonS3

**Parameter Description:** AmazonS3Visible enabled Amazon S3 as an output option on the Conga Composer user interface.



- You need to go to Conga Composer Setup, click the Amazon S3 tab, and enter your credentials for this integration to work. For more information, click [here](#).



- Before sending an email using Amazon SES, Conga recommends validating a domain or email address that you use to send or receive emails. For more information on how to validate your Amazon SES configuration, see [Sending test emails in Amazon SES](#).

Parameter Value	Definition
0	Hides Amazon S3 as an available output option.

**Example:**

```
&sc1=amazons3
&amazons3visible=1
```

## APDF

Assemble PDF Documents. Specifies whether merged PDF documents will be saved as individual PDF documents or consolidated into one document.

**Category:** Output Parameter


**Compatible with:** Composer 8

**Parameter Name:** APDF

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Multiple Template IDs specified (ex: `&TemplateId=a01268Hc74Vb9Mh, a0149mJh85ht78Cxj`).

**Parameter Description:** APDF stands for Assemble PDF Documents. The parameter specifies whether merged PDF documents are saved as individual PDF documents or consolidated into one document

 `&APDF=0` does not apply to output files emailed through Salesforce. In that case, the output PDFs will be zipped.

Parameter Value	Definition
0	Individual PDFs will not be assembled, but rather left as individual PDFs.
1	Individual PDFs will be assembled into one consolidated PDF file. (default)

**Example:**

```
&TemplateId=a01268Hc74Vb9Mh,a0149mJh85ht78Cxj
&APDF=1
```

## APF

Assembles the output PPTX presentation files before the file is either converted to PDF or delivered to the specified output location.

**Category:** PowerPoint Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** APF

**Parameter Value:** [0 or 1]

**Parameter Requirements:** All templates must be PowerPoint PPTX.

**Parameter Description:** Assembles the output PPTX presentation files before the file is either converted to PDF or delivered to the specified output location. The benefit is a unified slide numbering and a seamless output document.

Value	Definition
0	PPTX output files are not assembled, resulting in a ZIP file output. (default)
1	PPTX output files are assembled.

**Example:**

```
&APF=1
```

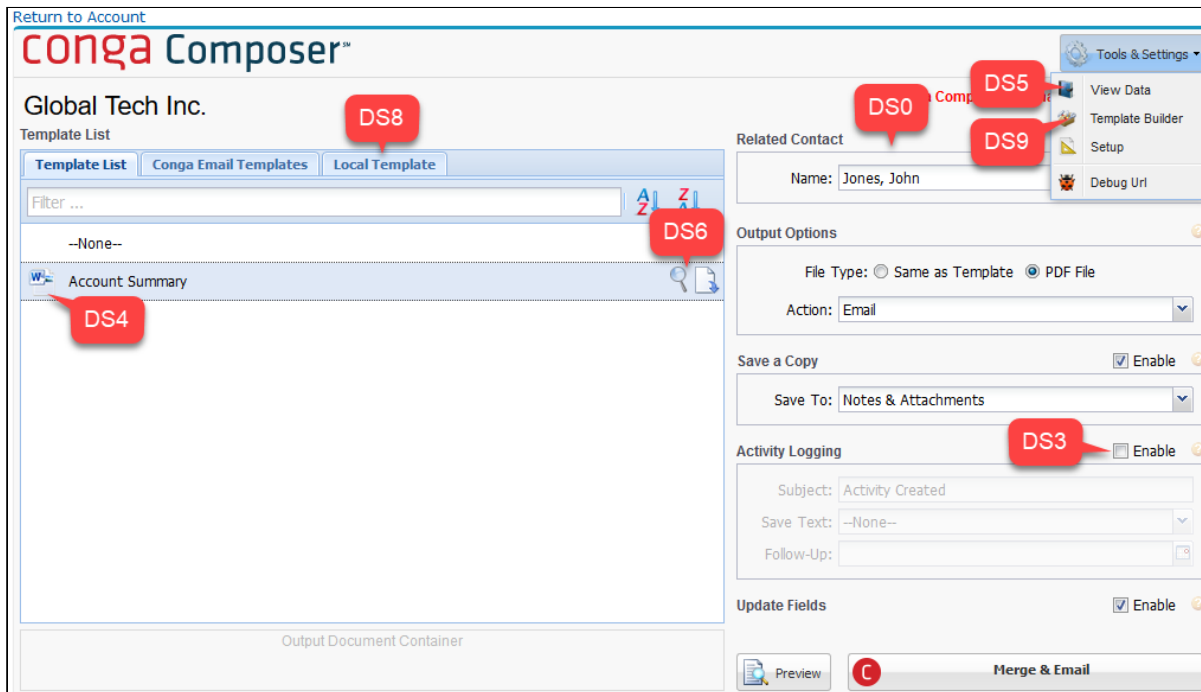
## Appearance Parameters

It may sometimes be desirable to preset certain user interface settings on the Conga Composer dialog and restrict the user's ability to alter the settings.

**Description**

You can preset certain user interface settings on the Conga Composer dialog and restrict the user's ability to alter the settings.

⚠ This is a navigation and convenience feature, not a security feature. Though the parameters are deliberately cryptic to obscure their purpose, a determined user could potentially alter them in a URL and override the intended behavior.



For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

## Parameters

- [DS0](#) specifies whether to disable Conga Composer dialog box controls.
- [DS1](#) disables the Merge button.
- [DS2](#) disables the Email button.
- [DS3](#) disables the Activity Logging options.
- [DS4](#) locks the list of template choices so that a user cannot select a different template.
- [DS5](#) disables View Data options.
- [DS6](#) disables View Template options.
- [DS7](#) bypasses the Conga Composer dialog box and runs a solution in background mode.
- [DS8](#) displays the Local Template tab.
- [DS9](#) disables Template Builder options.
- [DS10](#) disables the Attach File button in the Conga Email Editor.

## AssembleChatter

When generating a multi-file output, this parameter attaches multiple files to a single Chatter post.

**Category:** Salesforce Chatter Parameters


**Compatible with:** Composer 8

**Parameter Name:** AssembleChatter

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** The AssembleChatter parameters assembles Chatter Files. When generating a multi-file output, this parameter will attach multiple files to a single Chatter post.

 Omitting this parameter will result in the default behavior which creates one Chatter post per document.

Value	Definition
0	Default chatter behavior (one Chatter post per document).
1	Multiple files will be attached to a single chatter post.

**Example:**

&AssembleChatter=1

## AttachmentID

Specifies the Salesforce IDs of up to 150 attachments that may be appended to a merge process. Specify multiple attachments by separating the Salesforce IDs with commas.

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** AttachmentID


**Parameter Value:** [Salesforce ID of an object that allows attachments]

**Parameter Requirements:** None

**Parameter Description:** AttachmentID specifies the Salesforce IDs of up to 150 static attachments to be included in a single merge process. Additional [product data limits](#) apply. Specify multiple attachments by separating the Salesforce IDs with commas.

When you use this parameter, ignores the AttachmentSourceId and SelectAttachments parameters.

Acceptable file types include Excel, PDF, PowerPoint, .PNG, .JPG, and Word.

 Use a [Qvar](#) to specify multiple Attachment IDs to avoid exceeding URL length limitations. While up to 150 static attachment IDs can be specified, only five appear as selected in the UI Attachments tab. `&AttachmentID={Qvar0}` can be used as a method to dynamically obtain that ID. The Local Template tab will not work when using this parameter. You must use the TemplateId parameter to specify and select templates.

**Example:**

```
&AttachmentId=00P000000004yHQ,00P000000005Ljm
```

## AttachmentParentId

Specifies the record in Salesforce to which the output file is attached, when requested.

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** AttachmentParentId

**Parameter Value:** [Salesforce ID of an object that allows attachments]

**Parameter Requirements:** SC0 and SC1

**Parameter Description:** AttachmentParentID specifies the record in Salesforce to which the output file is attached when requested.

This parameter also works with Conga Batch ([Qmode=2](#)) to control the behavior of the merge operation's primary output.

 When used with [SCI=Chatter](#), this parameter may be set to a Chatter User ID or Chatter Group Id.

**Example:**

&SC0=1

&SC1=Attachments

&AttachmentParentID=00361000000u4eh or &AttachmentParentID={!SBQQ\_\_Quote\_\_c.SBQQ\_\_Opportunity2Id\_\_c}

## AttachmentSourceId

Displays the 20 most recent attachments or Salesforce files from the record specified by this ID instead of the master object.

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** AttachmentSourceId

**Parameter Value:** [Salesforce ID of an object that allows attachments]

**Parameter Requirements:** &SelectAttachments=1

**Parameter Description:** AttachmentSourceId displays the 20 most recent attachments or Salesforce files from the record specified by this ID instead of the master object.

**Example:**

&SelectAttachments=1 or &AttachmentSourceId=001500000ju78xK or AttachmentSourceId

## AWD

Assembles the output Word documents before the file is either converted to PDF or delivered to the specified output location. The benefit is a unified page numbering and a seamless output document.

**Category:** Word Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** AWD

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** All templates must be Microsoft Word.

**Parameter Description:** AWD stands for Assemble Word Documents.

The parameter assembles the output Word documents before the file is either converted to PDF or delivered to the specified output location. The benefit is a unified page numbering and a seamless output document

⚠ Embedded fonts are supported only in the first template, so the use of embedded fonts is not recommended. Also, the output will only be a consolidated, single Word document when using Composer.

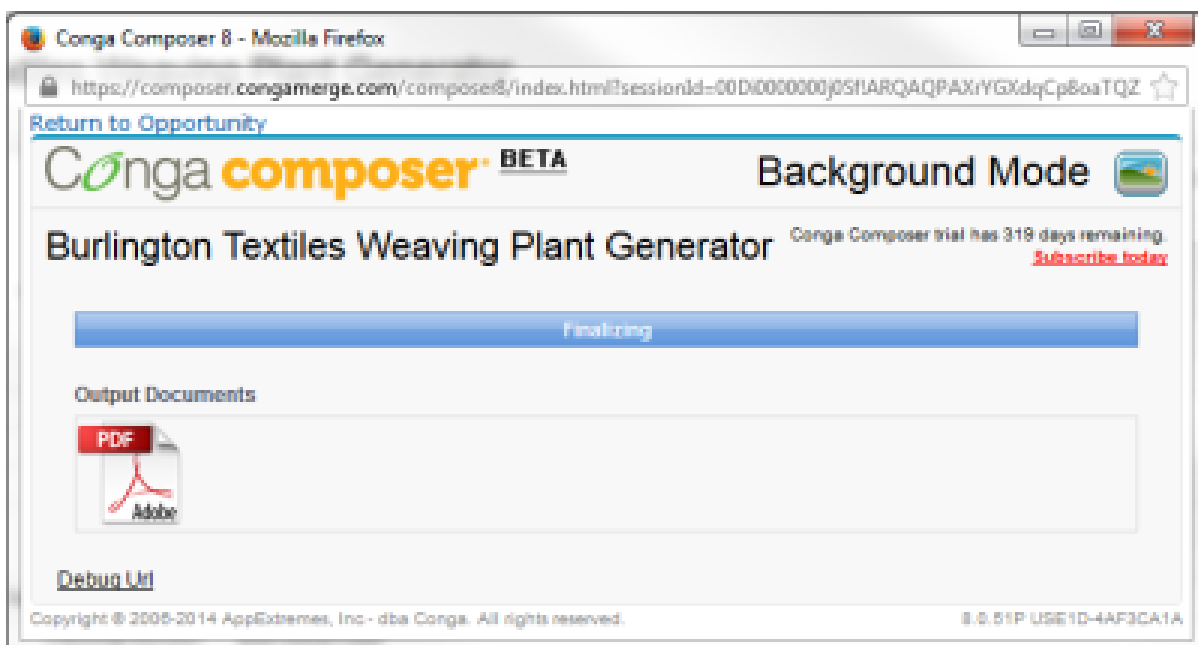
Value	Definition
0	Word output files are not assembled, resulting in a ZIP file output. (default)
1	Word output files are assembled with a New Page Section Break between documents.
2	Word output files are assembled with a Continuous Section Break between documents.

Example:

&AWD=1

## Background Mode Parameters

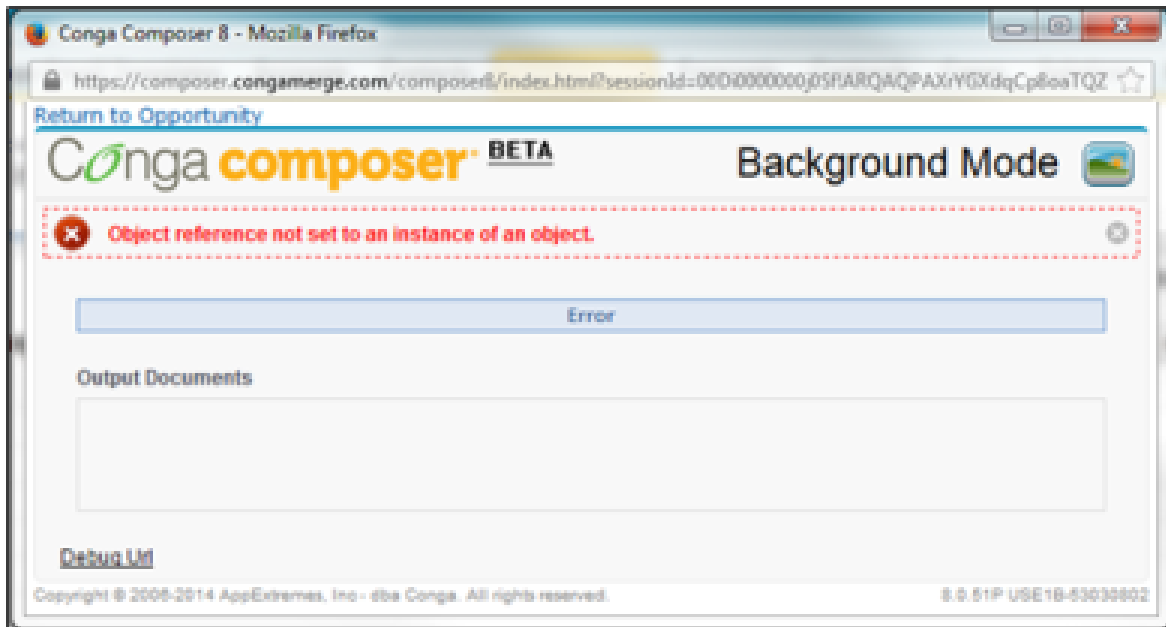
Background Mode parameters let you bypass the classic Conga Composer dialog box.



For more information on background mode, see [Automating Solutions using Background Mode Parameters](#). For more information about parameters in general and how to use them,

see [Adding Parameters to a Solution, A Few Rules About Parameters, and How Parameters are Formatted.](#)

- ✓ Conga Composer error messages continue to appear on the dialog box, even when Background Mode is enabled.



If you use a pop-up window for a background mode dialog box, we recommend enabling the Resizable property so the window can be stretched in case an error message is displayed.

### Parameters

- [BML](#) displays a message on the Background Mode dialog box.
- [DS7](#) activates Conga Composer in background mode.
- [DS7Preview](#) enables Background Mode Preview.
- [QMode](#) parameters are required when working with Conga Batch® or Conga Trigger solutions.

## Bigtincan Integration Parameters

These parameters enable Conga applications to interact with bigtincan.

### Description

The following parameter enables Conga applications to interact with bigtincan as an output file destination.



 A user must have a valid email address to ensure the correct account is used.

## Parameters

- `bigtincanVisible` specifies whether bigtincan is visible to the end-user as an output file destination.
- `btccuser` is a required parameter that corresponds to the email address of the bigtincan user who publishes the document.
- `DS7` enables merge operations to occur in the background and the document is sent to bigtincan.
- `OFP` specifies the bigtincan Story where the merged output files can be saved.
- `SCI` defines where a copy of the merged output file will be stored.

## bigtincanVisible

This parameter determines if bigtincan integration is visible or not for output file distribution.

**Category:** Bigtincan Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** bigtincanVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:**

- The `&OFP` parameter with a value matching the bigtincan root folder. The `OFP` parameter must be two segments only. The two segments must represent the channel/story. The channel must be created through the bigtincan hub. The story can be created dynamically.
- A valid user email is needed to ensure the user is in the correct account.

**Parameter Description:** bigtincanVisible determines if Bigtincan integration is visible or now. This parameter requires a user to authenticate through bigtincan.

Value	Definition
0	Bigtincan visibility is disabled.
1	Bigtincan visibility is enabled.

**Example:**

`&OFP=Composer+Content/{!Account.Name}` or `&bigtincanVisible=1` or `bigtincanvisible`

## BML

Displays a message on the Background Mode dialog box. Displays up to 50 characters.

**Category:** Automation-Background Mode Parameter

**Compatible with:** Composer 8

**Parameter Name:** BML

**Parameter Value:** [text]

**Parameter Requirements:** DS7 (Background mode must be enabled)

**Parameter Description:** BML stands for Background Mode Label. The parameter displays a message on the Background Mode dialog box. It displays up to 50 characters.

**Example:**

```
&DS7=1 or &BML=Account+Brief+for+{!Account.Name}
```

## Box Integration Parameter

### Description

The following parameter enables Conga apps to interact with Box.

### Parameters

- [BoxVisible](#) specifies whether Box integrations are visible to the end-user.
- [DS7](#) enables merge operations to occur in the background.
- [OFP](#) specifies the file path where the merged output files can be saved.
- [QMode](#) for use with Conga Batch or Conga Trigger solutions.
- [SCI](#) defines where a copy of the merged output file is stored.

## BoxVisible

This parameter determines if Box integration is visible or not.

**Category:** Box Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** BoxVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** The &OFP parameter with a value matching the Box root folder.

**Parameter Description:** The BoxVisible parameter determines if Box integration is visible or not. This parameter will require a user to authenticate through Box.

Value	Definition
0	Box visibility is disabled.
1	Box visibility is enabled.

**Example:**

&OFP=Composer+Documents/Opportunities/{Opportunity.Name} or &BoxVisible=1

## BrandingCode

The BrandingCode parameter is used to white-label Conga Composer for Enterprise Edition customers.

**Category:** Branding


**Compatible with:** Composer 8

**Parameter Name:** BrandingCode

**Parameter Value:** [*yourcode*]

**Parameter Requirements:** Composer Enterprise Edition.

**Parameter Description:** BrandingCode is used to white-label Conga Composer for Enterprise Edition customers. After submitting your logo to Conga you will be provided with a Branding Code to be entered with this parameter.

 Contact your Account Manager for more information about Conga Composer Enterprise Edition. If you want to add a Branding code to your Salesforce org, log a support case and provide the publicly accessible image URL.

**Example:**

&BrandingCode=yourcode

## btouser

This parameter determines if bigtincan integration is visible or not for output file distribution.

**Category:** Bigtincan Integration Parameter


**Compatible with:** Composer 8

**Parameter Name:** btcuser

**Parameter Value:** Valid email address within your bigtincan hub. This can be a static or dynamic value.

**Parameter Requirements:** A valid user email is needed to ensure the user is in the correct account and can publish documents. The email address on the User's profile in Salesforce must be the same as the User email address in Bigtincan.

**Parameter Description:** Btcuser determines a valid email address for the user who can publish documents.

 This parameter requires a user to authenticate through bigtincan. The email must have write access to the channel Composer is accessing.

**Example:**

```
&btcUser={!User.Email}
```

## CETFilter

Specifies the where clause to filter the list of Conga Email Templates that display in the user interface. Typically, a custom field is added to the Conga Email Template object and this parameter is used in conjunction with that field.

**Category:** Email Parameter

**Compatible with:** Composer 8

**Parameter Name:** CETFilter

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** CETFilter specifies the WHERE clause to filter the list of Conga Email Templates that displays in the user interface. Typically, a custom field is added to the Conga Email Template object and this parameter is used in conjunction with that field.

In the example below, the System Administrator created and referenced a custom field, Language\_\_c.

**Example:**

```
&CETFilter=Language__c INCLUDES ('English')
```

## ChatterCommunityId

When attaching an output file to a Chatter Group as a Community User, you must specify the Chatter Community (via its Salesforce ID) to which the file will be visible.

**Category:** Chatter Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ChatterCommunityId

**Parameter Value:** [ChatterCommunityId]

**Parameter Requirements:** None

**Parameter Description:** ChatterCommunityID specifies a Chatter Community ID. When attaching an output file to a Chatter Group as a Community User, you must specify the Chatter Community (via its Salesforce ID) to which the file will be visible.

Note: Chatter Community IDs begin with ODB

**Example:**

```
&ChatterCommunityId=0DB0459az7hJk8s
```

## Conga Contracts Parameters

You can use these parameters when using Conga Composer as the document generation engine for Conga Contracts.

### Description

Use these parameters when using Conga Composer as the document generation engine for Conga Contracts.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

### Parameters

- [DataTagEnable](#) creates data tags in the output document that allows changes to the document to be tracked and synced into Salesforce when negotiation is done.
- [RM](#) allows merge fields within merge fields. Recursive merge is useful when merging clauses from the Clause Library that contain merge fields.
- [DS7](#) with a value of 111 and 112 allows you to run Composer in Background Mode and send the document for negotiation. Additionally, this allows you to use a Conga Email Template.

## CongaEmailTemplateGroup

This parameter filters the list of Conga Email Templates that appear in the Email Templates list on the Composer user interface to those that match the specified criteria. The value may include one or more values, separated by commas. This parameter may be populated via a QVAR query.

**Parameter Category:** Email Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** CongaEmailTemplateGroup

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** CongaEmailTemplateGroup filters the list of Conga Email Templates that appear in the Email Templates list on the Composer user interface to those that match the specified criteria. The value may include one or more values, separated by commas. This parameter may be populated via a QVAR query.

 You can also use this abbreviate this parameter name and specify multiple values: *&CETG=USA, Contact*

**Example:**

`&CongaEmailTemplateGroup=Quotes`

## CongaEmailTemplateld

This parameter pre-selects the Conga Email Template specified in the Email Templates list on the Composer user interface. You may specify only one Conga Email Template ID. This parameter may be populated via a QVAR query.

**Category:** Email Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** CongaEmailTemplateld

**Parameter Value:** [ID]

**Parameter Requirements:** None

**Parameter Description:** CongaEmailTemplateID pre-selects the Conga Email Template specified in the Email Templates list on the Composer user interface. You may specify only one Conga Email Template ID. This parameter may be populated via a QVAR query.

 This parameter can also be abbreviated as &CETID=. Conga Email Templates are not compatible with Composer solutions integrated with Conga Sign.

#### Examples:


&CongaEmailTemplateId=a0WG0000006PNrS or &CETId=a0WG0000006PNrS

## Conga Email Template Parameters

### Description

These parameters control the behavior of Conga Email Templates.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

 Track Activities is a required Salesforce setting for the object you want to send an email from.

### Parameters

- [CETFilter](#) specifies the where clause to filter the list of Conga Email Templates that display in the user interface.
- [CongaEmailTemplateGroup](#) filters the list of Conga Email Templates that appear in the Email Templates list on the Composer user interface to those that match the specified criteria.
- [CongaEmailTemplateId](#) pre-selects the Conga Email Template specified in the Email Templates list on the Composer user interface.
- [DS10](#) disables the Attach File button in the Conga Email Editor.
- [EmailAdditionalTo](#) sets the value of the Additional To field.
- [EmailBCC](#) sets the value of the BCC field.
- [EmailCC](#) sets the value of the CC field.
- [EmailFromId](#) specifies an Organization-Wide Address as the From identity of the email.
- [EmailLetterheadId](#) specifies which Salesforce email letterhead is used in conjunction with a Conga Email Template.
- [EmailRelatedToId](#) sets the value of the Related To field.

- [EmailReplyToId](#) changes the apparent sender of an email by setting the message ReplyTo and DisplayName properties to those of the specified user.
- [EmailSubject](#) sets the value of the Subject field.
- [EmailTemplateAttachments](#) indicates to preserve attachments stored with the Salesforce email template.
- [EmailTemplateId](#) loads the subject and body fields from a Salesforce email template.
- [EmailToId](#) sets the value of the To field.
- [EmailUseSignature](#) indicates whether to include or exclude the signature of the sending user from the email body.

## Conga Sign Integration Parameters

Each of these parameters affects the field specified on a Conga Sign Agreement record.


### Description

Conga Sign delivers Conga Composer documents through email, much like Salesforce itself, but adds the ability to add, track and file signatures.

Conga Sign allows you to use PDF and Word documents to collect data from your signers, position signature fields in your documents, collect two-party information from both sender and signer and more. You can add Conga Sign-specific form fields to Conga Composer PDF merge templates to enable those features.

Conga Sign is available separately on the AppExchange. You must install the Conga Sign package to use these integration parameters.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

 Conga Email Templates are not compatible with Composer solutions integrated with Conga Sign.

### Parameters

- [CSVisible](#) Specifies Conga Sign as the delivery and output methods in Composer.
- [CSRecipient\[1-10\]](#) ID can be a Salesforce Contact, User, or Lead.
- [CSRole\[1-10\]](#) Specifies the default Recipient role of Send Final.
- [CSEmail\[1-10\]](#) Specifies Conga Sign Recipient's email address.
- [CSFirst\[1-10\]](#) Specifies Conga Sign Recipient's first name.
- [CSLast\[1-10\]](#) Specifies Conga Sign Recipient's last name.
- [CSName\[1-10\]](#) Specifies Conga Sign Recipient's first and last name.



- [CSRoutingType](#) Specifies the routing recipient type and defaults to PARALLEL if not specified.
- [CSEmailMessage](#) Sends an email message and defaults to a generic message if not specified.
- [CSEmailSubject](#) Specifies the subject line of an email.
- [CSExpiration](#) Specifies the date and time for eSignature expiration.
- [CSExpirationReminder](#) Specifies an expiration date reminder on an eSignature request.
- [CSRequestReminder](#) Specifies a request reminder in the number of days.
- [CSReassignAllowed](#) Specifies whether the recipient of a transaction can reassign the signer from the received transaction.
- [CSMobile\[1-10\]](#) Specifies a recipient's mobile number that the SMS authentication passcode is sent to.
- [CSAuth\[1-10\]](#) Specifies the authentication type for a recipient in a Conga Sign Transaction.
- [CSLang\[1-10\]](#) Specifies a recipient's language in a Conga Sign Transaction.
- [CScurl\[1-10\]](#) Specifies the URL that a signer can choose to be taken to after successfully completing the signing.
- [CSBusinessUnit](#) Specifies the Business Unit selected for a Conga Sign Transaction.
- [DS7](#) Specifies the send and execute actions for the Conga Sign Transaction in background mode.
- [QMode](#) Specifies full automation delivery with Conga Batch or Conga Trigger.
- [CSSenderReturnUrl](#) Specifies return URL that the sender is returned to after successfully sending a transaction.
- [CSMultiDocs](#) Specifies whether you can send multiple documents to Conga Sign.

## ContactId

Defines the related Contact drop-down menu in the user interface.

**Category:** Data Gathering Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** ContactId

**Parameter Value:** [ID]

**Parameter Requirements:** The Salesforce ID of a Contact record or a field that contains such an ID.

**Parameter Description:** ContactID defines the related Contact drop-down menu in the user interface. The ContactID parameter also sets the default email and eSignature recipient,

populates the Master-Contact dataset, and defines which Contact activities will be logged against.

 For Salesforce standard objects with related Contacts (Account, Case, Opportunity, Quote), will automatically populate the related Contact drop-down menu, either from the look-up field on the Master Object or via the related Contact Roles.

**Example:**

```
&ContactId={!Referral.Contact__c}
```

## ContentCustomFieldName

Sets the name of a custom field to set on the Salesforce Content Record created through the merge process. Use the Salesforce API name of the field on the Salesforce Content Object.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ContentCustomFieldName

**Parameter Value:** [value]

**Parameter Requirements:** ContentWorkspaceId and ContentCustomFieldValue.

**Parameter Description:** ContentCustomFieldName sets the name of a custom field to set on the Salesforce Content Record created through the merge process. Use the Salesforce API name of the field on the Salesforce Content Object.

**Example:**

```
&ContentWorkspaceId=058a123456789ab or &ContentCustomFieldValue=001ABCEDFGHIJL or  
&ContentCustomFieldName=Account__c
```

## ContentCustomFieldValue

Sets the value of the corresponding custom field.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ContentCustomFieldValue

**Parameter Value:** [value]

**Parameter Requirements:** ContentCustomFieldName

**Parameter Description:** ContentCustomFieldValue sets the value of the corresponding custom field.

**Example:**

&ContentCustomFieldName=Account\_\_c or &ContentCustomFieldValue=001ABCEDFGHIJL

## ContentDocumentId

Versioning with Salesforce Files - Specifies the Salesforce ID of the Salesforce File record ID where you want to use Salesforce Files versioning features.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ContentDocumentID

**Parameter Value:** [Salesforce ID of the Salesforce File ID]

**Parameter Requirements:** SC0 and SC1 or DS7=11, 111

**Parameter Description:** ContentDocumentID specifies the Salesforce ID of the Salesforce File record ID in which you want to use Salesforce Files' versioning features.

Typically, you use a QVar to bring the value of this Salesforce File record ID when a first version is merged already:

```
&QVar0ID=a031t000004INZr
```

```
&QVar0format=10000
```

```
&ContentDocumentID={QVar0}
```

And the corresponding SOQL Query is:

```
SELECT ContentDocumentId
```

```
FROM ContentDocumentLink
```

```
WHERE LinkedEntityId = '{pv0}'
```

**Example:**

&SC0=1 or &SC1=SalesforceFile or &ContentDocumentID={QVar0}

## ContentNetworkId

Specifies the Network Id for the appropriate community to ensure the file record is created and shared within Salesforce when saving a Salesforce file as a community user.

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8

**Parameter Name:** ContentNetworkId

**Parameter Value:** [Unique Community Network ID]

**Parameter Requirements:** None

**Parameter Description:** ContentNetworkID specifies a Communities' Networks ID. When saving a Salesforce File as a Community user, the Network Id for the appropriate community must be specified to ensure the file record is created and shared appropriately within Salesforce.

**Example:**

&ContentNetworkId=0DB0459az7hJk8s

## contentShareType

The contentShareType parameter determines the permission type granted to the user for a shared file.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** contentShareType

**Parameter Value:** [V, C, I]

**Parameter Requirements:** None

**Parameter Description:** ContentShareType determines the permission type granted to the user for a shared file.

- ✔ The contentShareType parameter is useful for determining Salesforce File permissions for Community users.

It is highly recommended to use both the *contentShareType* parameter and the [contentVisibility](#) parameter in the same Composer button URL to determine Salesforce File visibility and permissions.

Value	Definition
V	Assigns the Viewer permission. The user can explicitly view but not edit the shared file.
C	Assigns the Collaborator permission. The user can explicitly view and edit the shared file.
I	Assigns the Inferred permission. The user's permission is determined by the related record. For shares with a library, this is defined by the permissions the user has in that library.

**Example:**

&contentShareType=I

## contentVisibility

The contentVisibility parameter determines whether a file is available to all users, internal users, or shared users.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** contentVisibility

**Parameter Values:** [AllUsers, InternalUsers, SharedUsers]


**Parameter Requirements:** None

**Parameter Description:** The contentVisibility determines whether a file is available to all users, internal users, or shared users.

- ✓ The contentVisibility parameter is useful for giving Community users access to Salesforce Files generated with Conga Composer. To give Community users access to Composer-generated files, use &contentVisibility=AllUsers.

It is highly recommended to use both the *contentVisibility* parameter and the [contentShareType](#) parameter in the same Composer button URL to determine Salesforce File visibility and permissions.

Value	Definition
AllUsers	The file is available to all users who have permission to see the file.
InternalUsers	The file is available only to internal users who have permission to see the file.
SharedUsers	The file is available to all users who can see the feed to which the file is posted.

 SharedUsers is used only for files shared with users, and is available only when an org has private org-wide sharing on by default. The SharedUsers value is available in API version 32.0 and later

**Example:**

&contentVisibility=AllUsers

## ContentVisible

Enables Salesforce Content as an output option on the Conga Composer user interface.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ContentVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** A Salesforce ID of a Content Workspace. The ID must begin with the prefix 058.

**Parameter Description:** ContentVisible enables Salesforce Content as an output option on the Conga Composer user interface.

Value	Definition
0	Salesforce Content is not the designated Action. (default)
1	The Action is set to Salesforce Content.

**Example:**

&ContentWorkspaceld=058a123456789ab or &ContentVisible=1

## ContentWorkspaceld

Sets the Salesforce ID of the Content Workspace under which attachments will be stored thus providing versioning with Salesforce Files.

**Category:** Salesforce Content Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ContentWorkspaceld

**Parameter Value:** [ID]

**Parameter Requirements:** A Salesforce ID of a Content Workspace. The ID must begin with the prefix 058.

**Parameter Description:** Sets the Salesforce ID of the Content Workspace under which attachments will be stored.

### Important

Do not use this ID on the Conga Batch Record. Ensure that users have permission to view the content workspace.

### Example:

&ContentWorkspaceld=058a123456789ab

## CSAuth[1-10]

This parameter determines the authentication type for a recipient in a Conga Sign transaction.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSAuth[1-10]

**Parameter Value:** [SMS or NONE]

**Parameter Requirements:** CSRecipient[1-10]

**Parameter Description:** CSAuth[1-10] determines the authentication type for a recipient in a Conga Sign transaction. If the CSAuth[1-10] parameter is not defined, the authentication for a recipient defaults to NONE.

For more information on SMS authentication with Conga Sign, see [SMS Authentication](#).

Recipients assigned SMS authentication must enter a one-time passcode delivered through SMS, prior to viewing a Conga Sign document. The phone number that recipients receive the passcode to is determined by the [CSMobile\[1-10\]](#). If the [CSMobile\[1-10\]](#) parameter is not present, the passcode is sent to the phone number on the Mobile field of the recipient's Salesforce (Contact, User, or Lead) record.

**Example:**

```
&CSRecipient1=0031100000BbPlu  
&CSAuth1=SMS  
&CSMobile1=13038675308
```

## CSBusinessUnit

This parameter determines the Business Unit selected for a Conga Sign transaction.

**Important**

Coming soon! Business Units are not available for use until the Conga Sign 1.49.0 release.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSBusinessUnit

**Parameter Value:** [Business Unit API Name]

**Parameter Requirements:** None

**Parameter Description:** CSBusinessUnit determines the [Business Unit](#) selected for a Conga Sign Transaction. The parameter can be used only once in a Composer Solution, as individual transactions are restricted to one Business Unit.

The parameter value must be the API Name of an active Business Unit. The Business Unit API Name cannot contain any spaces.

**Example:**

```
&CSBusinessUnit=Global_Tech_Inc
```



## CScurl[1-10]

This parameter determines the URL that a signer can choose to be taken to after successfully completing the signing.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CScurl[1-10]

**Parameter Value:** [An active website URL beginning with **http://** or **https://**]

**Parameter Requirements:** None

**Parameter Description:** CScurl[1-10] determines the URL that a recipient can choose to be taken to after successfully completing the signing. With this parameter, recipients can click Continue after completing signing and are taken to the URL defined with the CScurl[1-10] parameter. You must `urlencode` the URL that is used in the CScurl[1-10] parameter to prevent any subsequent parameters from dropping characters. This can be done using the Salesforce function `URLENCODE()` within the button.

If the recipient is a standard SIGNER and clicks Continue after completing the signing, they are taken to the URL in the same browser tab. If the recipient is an IN\_PERSON\_SIGNER and clicks Continue after completing the signing, they are taken to the URL in a new browser tab.

### Important

The format of the URL must start with either **http://** or **https://**. The **www.** portion of a URL is optional.

Example CScurl[1-10] parameter values:

- `http://www.conga.com`
- `http://conga.com`
- `https://www.conga.com`
- `https://conga.com`

**Example:**

```
&CSRecipient1=0031100000BbPlu  
&CSRole1=SIGNER  
&CScurl1={!URLENCODE('https://conga.com')}
```

## CSEmail[1-10]

**Category:** Conga Sign

**Compatible with:** Conga Composer 8

**Parameter Name:** CSEmail[1-10]


**Parameter Value:** [static email address, or a Salesforce merge field]

**Parameter Requirements:** [CSFirst\[1-10\]](#) and [CSLast\[1-10\]](#), or [CSName\[1-10\]](#)

**Parameter Description:** CSEmail[1-10] determines a new Conga Sign Recipient's email address in a transaction. The email address must be in the following format:

<username>@<domain>.com.

Use the CSEmail[1-10], [CSFirst\[1-10\]](#), and [CSLast\[1-10\]](#) parameters (or [CSName\[1-10\]](#)) to create a new Conga Sign Recipient that does not already exist as a Contact, Lead, or User record in Salesforce.

 Creating and adding a new Conga Sign Recipient to a transaction does not create a Contact, Lead, or User record in Salesforce. It creates a Conga Sign Recipient (custom object) record specific to that transaction only. New Conga Sign Recipients created with Conga Sign integration parameters cannot be added or re-used in other transactions.

The CSEmail[1-10] parameter value can either be a static value, such as [john.doe@gmail.com](mailto:john.doe@gmail.com), or a Salesforce merge field that returns an email address (ex: `{!CustomObject.EmailAddress}`).

### **Important**

You must use the CSEmail[1-10] parameter with either the [CSFirst\[1-10\]](#) and [CSLast\[1-10\]](#) parameters, or use it separately with the [CSName\[1-10\]](#) parameter. You cannot use [CSFirst\[1-10\]](#), [CSLast\[1-10\]](#), and [CSName\[1-10\]](#) together to define a new Conga Sign Recipient.

### **Example:**

&CSEmail1=[john.doe@gmail.com](mailto:john.doe@gmail.com)

&CSFirst1=John

&CSLast1=Doe

```
&CSEmail2={!CustomObject.Email}
```

```
&CSFirst2=Jane
```

```
&CSLast2=Doe
```

## CSEmailMessage

Sets the message text for the recipients, defaults to a generic message if not specified.

**Category:** Conga Sign


**Compatible with:** Composer 8

**Parameter Name:** CSEmailMessage

**Parameter Value:** [Text]

**Parameter Requirements:** None

**Parameter Description:** CSEmailMessage sets the message text for the recipients, defaults to a generic message if not specified.

 Because this value is transmitted through the window URL, we recommend that this value be less than 40 characters.

**Example:**

```
&CSEmailMessage=[text]
```

## Add Line Breaks in Email Message

You can include line breaks in the email message when using the CSEmailMessage parameter with either DS7=1141 or DS7=1142. Follow the instructions below to include line breaks:

- If the button begins with "https://composer.congamerge.com", insert %250a in the location(s) where you want the line to break.
- If the button begins with "/apex/APXTConga4\_\_Conga\_Composer?SolMgr=1", insert %25250a in the location(s) where you want the line to break.

**Example:**

```
&CSEmailMessage=This is a message.%250alt has a line break.
```

The message is displayed as:

This is a message.

It has a line break.

## CSEmailSubject

Sets the subject line of an email.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSEmailSubject

**Parameter Value:** [Text]

**Parameter Requirements:** None

**Parameter Description:** CSEmailSubject sets the subject line of an email.

**Example:**

`&CSEmailSubject=[Text]`

## CSExpiration

Specifies the expiration date and time for recipients of eSignature.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSExpiration

**Parameter Value:** [Date/Time]

**Parameter Requirements:** None

**Parameter Description:** CSExpiration specifies the expiration date and time for recipients of eSignature.

The format as yyyy-MM-dd'T'HH:mm:ss'Z for example: `&csexpiration=2018-01-22T20:00:01Z`

You can also dynamically set the expiration date using a Salesforce formula (text) field. For example, to set the expiration date thirty days from today's date, use the following syntax in the formula (text) field and then use the formula field's merge field as the value of the csExpiration parameter: `TEXT(TODAY()+30) & "T" & TEXT(TIMEVALUE(NOW()+0.04167)) & "Z"`

**Example:**

`&CSExpiration=[yyyy-MM-dd'T'HH:mm:ss.SSS'Z]`

## CSExpirationReminder

Specifies the reminder date and time of expiration for recipients of eSignature.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSExpirationReminder

**Parameter Value:** [Date/Time]

**Parameter Requirements:** None

**Parameter Description:** CSExpirationReminder specifies the reminder date and time of expiration for recipients of eSignature.

The format as yyyy-MM-dd'T'HH:mm:ss'Z for  
example: `&csexpirationreminder=2018-01-22T20:00:01Z`.

You can also dynamically set the reminder date using a Salesforce formula field. For example, to set the reminder date thirty days from today's date, use the following syntax in the formula field and then use the formula field's merge field as the value of the csExpiration parameter: `TEXT(TODAY()+30) & "T" & TEXT(TIMEVALUE(NOW()+0.04167)) & "Z"`.

**Example:**

`&CSExpirationReminder=[yyyy-MM-dd'T'HH:mm:ss.SSS'Z]`

## CSFirst[1-10]

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSFirst[1-10]

**Parameter Value:** [static first name, or a Salesforce merge field]

**Parameter Requirements:** [CSLast\[1-10\]](#) and [CSEmail\[1-10\]](#)

**Parameter Description:** CSFirst[1-10] determines a new Conga Sign Recipient's first name in a transaction. Use the [CSEmail\[1-10\]](#), [CSFirst\[1-10\]](#), and [CSLast\[1-10\]](#) parameters (or [CSName\[1-10\]](#)) to create a new Conga Sign Recipient that does not already exist as a Contact, Lead, or User record in Salesforce.

**⚠** Creating and adding a new Conga Sign Recipient to a transaction does not create a Contact, Lead, or User record in Salesforce. It creates a Conga Sign Recipient (custom object) record specific to that transaction only. New Conga Sign Recipients created with Conga Sign integration parameters cannot be added or re-used in other transactions.

The CSFirst[1-10] parameter value can either be a static value, such as John, or a Salesforce merge field that returns a first name (ex: {!CustomObject.FirstName}).

**📌 Important**

You must use the [CSEmail\[1-10\]](#) parameter with either the CSFirst[1-10] and [CSLast\[1-10\]](#) parameters, or use it separately with the [CSName\[1-10\]](#) parameter. You cannot use CSFirst[1-10], [CSLast\[1-10\]](#), and CSName[1-10] together to define a new Conga Sign Recipient.

**Example:**

```
&CSEmail1=john.doe@gmail.com
&CSFirst1=John
&CSLast1=Doe
&CSEmail2=jane.doe@gmail.com
&CSFirst2={!CustomObject.FirstName}
&CSLast2={!CustomObject.LastName}
```

## CSLang[1-10]

This parameter allows you to specify a recipient's language in a Conga Sign transaction.

**Category:** Conga Sign

**Compatible with:** Composer 8


**Parameter Name:** CSLang[1-10]

**Parameter Value:** [de, fr, en-US, es, pt-PT, pt-BR, it, el, nl-NL, hr, pl, uk, ro, ka, ru, sh.]

**Parameter Requirements:** None

**Parameter Description:** CSLang[1-10] allows you to specify the language that a recipient completes a Conga Sign transaction in.

Supported languages include English (US), French, German, Spanish (Spain), Portuguese (Portugal), Portuguese (Brazil), Italian, Greek, Dutch (Netherlands), Croatian, Polish, Ukrainian, Romanian, Georgian, Russian, and Serbian (Latin).

 English (US) is the default language for the Conga Sign transaction if a language is not specified.

Use the language codes below to assign a language to a Conga Sign recipient.

- German: de
- French: fr
- Spanish (Spain): es
- Portuguese (Portugal): pt-PT
- Portuguese (Brazil): pt-BR
- Italian: it
- Greek: el
- Dutch (Netherlands): nl-NL
- Croatian: hr
- Polish: pl
- Ukrainian: uk
- Romanian: ro
- Georgian: ka
- Russian: ru
- Serbian (Latin): sh

For more information, see [Supported Languages in Conga Sign](#).

#### Example:

```
&CSRecipient1=0031100000BbPlu
&CSLang1=fr
```

```
&CSRecipient2=0031100000CfDla
&CSLang2=de
```

## CSLast[1-10]

**Category:** Conga Sign

**Compatible with:** Conga Composer 8

**Parameter Name:** CSLast[1-10]

**Parameter Value:** [static last name, or a Salesforce merge field]

**Parameter Requirements:** [CSFirst\[1-10\]](#) and [CSEmail\[1-10\]](#)

**Parameter Description:** CSLast[1-10] determines a new Conga Sign Recipient's last name in a transaction. Use the [CSEmail\[1-10\]](#), [CSFirst\[1-10\]](#), and CSLast[1-10] parameters (or

**CSName[1-10]**) to create a new Conga Sign Recipient that does not already exist as a Contact, Lead, or User record in Salesforce.

**⚠** Creating and adding a new Conga Sign Recipient to a transaction does not create a Contact, Lead, or User record in Salesforce. It creates a Conga Sign Recipient (custom object) record specific to that transaction only. New Conga Sign Recipients created with Conga Sign integration parameters cannot be added or re-used in other transactions.

The **CSLast[1-10]** parameter value can either be a static value, such as Doe, or a Salesforce merge field that returns the last name (ex: `{!CustomObject.Name}`).

**📌 Important**

You must use the **CSEmail[1-10]** parameter with either the **CSFirst[1-10]** and **CSLast[1-10]** parameters, or use it separately with the **CSName[1-10]** parameter. You cannot use **CSFirst[1-10]**, **CSLast[1-10]**, and **CSName[1-10]** together to define a new Conga Sign Recipient.

**Example:**

```
&CSEmail1=john.doe@gmail.com
&CSFirst1=John
&CSLast1=Doe
&CSEmail2=jane.doe@gmail.com
&CSFirst2={!CustomObject.FirstName}
&CSLast2={!CustomObject.LastName}
```

## CSMobile[1-10]

This parameter defines a recipient's mobile number that the SMS authentication passcode is sent to.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSMobile[1-10]

**Parameter Value:** [mobile phone number including the country code]

**Parameter Requirements:** CSAuth[1-10]=SMS



**Parameter Description:** CSMobile[1-10] defines a recipient's mobile number that the SMS authentication passcode is sent to. The number must be all numeric and contain the appropriate country code (ex: for the US, the country code is 1).

The country code must be included in the phone number used for this parameter value, or it will result in an error message.

No special characters or text are allowed in the mobile number used for the CSMobile[1-10] parameter value.

- ✓ Do not add the + symbol for a country code if you are using a static numeric value for the CSMobile[1-10] parameter. Using the + symbol in that scenario will result in an error.

If the mobile number is not provided in the CSMobile[1-10] parameter, it is automatically retrieved from the recipient's Salesforce record (Contact, User, or Lead). Specifically, the number is retrieved from the standard Mobile field in Salesforce.

The Mobile field's value must be in the following format:

- Full E.164 format, which includes the '+' and the country code (ex: US mobile number: +13036669999)

**Example:**

```
&CSRecipient1=0031100000BbPlu  
&CSAuth1=SMS  
&CSMobile1=13038675309
```

## CSMultiDocs

This parameter allows you to send multiple documents to Conga Sign.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSMultiDocs

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &APDF=0, templateId with multiple values

**Parameter Description:** This parameter allows you to send multiple documents to Conga Sign. You must provide the CSMultiDocs=1 to send multiple documents for eSignature in a single transaction using Conga Composer. This parameter must be used with the templateId parameter and APDF=0.

**Example:**

```
&csmultidocs=1  
&APDF=0  
&templateid=a035000002abcde,a035000002ABCdE
```

## CSName[1-10]

**Category:** Conga Sign


**Compatible with:** Composer 8

**Parameter Name:** CSName[1-10]

**Parameter Value:** [static first and last name, or a Salesforce merge field]

**Parameter Requirements:** [CSEmail\[1-10\]](#)

**Parameter Description:** CSName[1-10] determines a new Conga Sign Recipient's first and last name in a transaction. Use the [CSEmail\[1-10\]](#) and CSName[1-10] parameters (or the [CSFirst\[1-10\]](#), and [CSLast\[1-10\]](#) parameters) to create a new Conga Sign Recipient that does not already exist as a Contact, Lead, or User record in Salesforce.

 Creating and adding a new Conga Sign Recipient to a transaction does not create a Contact, Lead, or User record in Salesforce. It creates a Conga Sign Recipient (custom object) record specific to that transaction only. New Conga Sign Recipients created with Conga Sign integration parameters cannot be added or re-used in other transactions.

The CSName[1-10] parameter value can either be a static value, such as John Doe, or a Salesforce merge field that returns a first and last name (ex: `{!CustomObject.Name}`). Use a space or + symbol to separate first and last name in the parameter value for CSName[1-10].

### Important

You must use the [CSEmail\[1-10\]](#) parameter with either the [CSFirst\[1-10\]](#) and [CSLast\[1-10\]](#) parameters, or use it separately with the CSName[1-10] parameter. You cannot use [CSFirst\[1-10\]](#), [CSLast\[1-10\]](#), and [CSName\[1-10\]](#) together to define a new Conga Sign Recipient.

### Example:

```
&CSEmail1=john.doe@gmail.com  
&CSName1=John Doe  
&CSEmail2=jane.doe@gmail.com  
&CSName2={!CustomObject.Name}
```

## CSReassignAllowed

This parameter determines the authentication type for a recipient in a Conga Sign transaction.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSReassignAllowed

**Parameter Value:** True or False

**Parameter Requirements:** None

**Parameter Description:** CSReassignAllowed determines if the recipient of a Conga Sign transaction is allowed to reassign the signer of the transaction.

For more information on Reassigning the Signer, see [Remove Reassign Signer](#).

Recipients of a Conga Sign Transaction are given the option to reassign the signer to another recipient of their choice from the signer email.

**Example:**

```
&CSReassignAllowed=true
```

## CSRecipient[1-10]

Allows you to specify the recipients of a Conga Sign Agreement.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSRecipient[1-10]

**Parameter Value:** [ID]

**Parameter Requirements:** None

**Parameter Description:** CSRecipient[1-10] allows you to specify the recipients of a Conga Sign Agreement. Each recipient must be the ID (not email address) of a Contact, User, or Lead. If you specify the CSRoutingType as SERIAL, the document is routed to the recipients in their order specified.

You can also use a dynamic, custom Salesforce field, for example: *&csRecipient1={!Opportunity.Primary\_ContactId\_\_c}*.

 You can specify up to 10 Conga Sign recipients by incrementing the number in this parameter.

 **Important**

At this time, you cannot use a [QVar](#) to populate the value of the CSRecipient parameter when merging from the Lightning Component.

**Example:**

```
&CSRecipient[1-10]=[ID]
```

or

```
&CSRecipient1=0031100000BbPlu
```

or

```
&CSRecipient2=0034100000z6aqO
```

## CSRequestReminder

Specifies how much time (in days) a reminder should be set on a request.

**Category:** Conga Sign

**Compatible with:** 8

**Parameter Name:** CSRequestReminder

**Parameter Value:** [Integer]

**Parameter Requirements:** None

**Parameter Description:** CSRequestReminder specifies how much time (in days) a reminder should be set on a request.

**Example:**

```
&csRequestReminder=5
```

## CSRole[1-10]

**Category:** Conga Sign Integration Parameters

**Compatible with:** Composer 8

**Parameter Name:** csroleX

**Parameter Value:** [IN\_PERSON\_SIGNER, SIGNER, or CC]

**Parameter Requirements:** None

**Parameter Description:**

- In UI Mode: allows the role field on the UI to be specified.
- In Automated Mode: enables writing a script specifying that some recipients are CC only.

This parameter can be used 10 times by changing the number after csrole. For example, *csrole2*, *csrole3*, *csrole4*, ... *csrole10*.

 If the csRole[1-10] is not specified, the default role is set to Signer.

**Example:**

&csRole1=CC

## CSRoutingType

Specifies the type of routing order set up for recipients.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSRoutingType

**Parameter Value:** [PARALLEL OR SERIAL]

**Parameter Requirements:** None

**Parameter Description:** CSRoutingType specifies the type of routing order set up for recipients. The routing order is in the order of recipients specified, for example, CSRecipient1, CSRecipient2, and so on.


Value	Definition
PARALLEL	All recipients are routed at the same level. (default)
SERIAL	Controls the routing order of recipients.

**Example:**

&CSRoutingType=PARALLEL or &CSRoutingType=SERIAL

## CSSenderReturnUrl

Set a string to determine a page that the sender is returned to after successfully sending a transaction.

 CSSenderReturnUrl parameter is not compatible in background mode (ds7=1142, apimode=3, qmode=congasign).

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSSenderReturnUrl

**Parameter Value:** http/https site or relative url.

**Parameter Requirements:** None

**Parameter Description:** This parameter allows the administrator to configure a return URL within the Conga Composer solution for senders to be returned to after successfully sending a transaction. You must urlEncode the URL that is used in the CSSenderReturnUrl parameter to prevent any subsequent parameters from dropping characters. This can be done using the Salesforce function URLENCODE() within the button.

### Example

- By using the below example, you are redirected to the Conga Support website after successfully sending a transaction.

#### Example

```
&cssenderreturnurl={!URLENCODE('https://documentation.conga.com/')}
```

- By using the below example, you are redirected to the Account record associated with the Opportunity record after successfully sending a transaction.

#### Example

```
&cssenderreturnurl={!URLENCODE(Opportunity.AccountId)}
```

## CSVisible

This parameter determines if Conga Sign integration is visible or not. The merge button becomes Merge & Conga Sign when &CSVisible=1.

**Category:** Conga Sign

**Compatible with:** Composer 8

**Parameter Name:** CSVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** CSVisible determines if Conga Sign integration is visible or not. The merge button becomes *Merge & CongaSign* when &csvisible=1

You must drag and drop the new button to the appropriate record.

Value	Definition
0	Conga Sign visibility is disabled.
1	Conga Sign visibility is enabled.

**Example:**

&CSVisible=1

## Culture

Overrides the default cultural settings for currency and date formatting. Useful in Word templates that format currency fields with \# Currency picture switch.

**Category:** Advanced Data Formatting Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** Culture

**Parameter Value:**[5-character language/country code]


**Parameter Requirements:** None

**Parameter Description:** The Culture parameter specifies cultural settings for currency and date formatting. It overrides the default cultural settings for currency and date formatting.

Culture is useful in Word templates that format currency fields with \# Currency picture switch.

This parameter is similar to &CurrencyCulture, but also formats the date as well as currency.

Valid options to use as the value of the parameter are available here: [language/country culture codes](#).

 The Culture parameter will not overwrite the date format of the Company's locale, only if the date field is being populated via a query. Master data is unaffected. A workaround would be to create a formula field on the same object the date field is on. You don't have to add the field on your layout, just bring it in via the query. Here is the formula field's (return type text) syntax: `TEXT(Month(Date_Test__c)) + "/" + LPAD(TEXT(Day(Date_Test__c)),2,"0") + "/" + LPAD(TEXT(YEAR(Date_Test__c)),2,"0")`. This formula field takes the highlighted field and populates data in the same order the Month, Date, and Year appears in.

**Example:**

&Culture=zh-HK

## CurrencyCulture

Overrides the default cultural settings for currency formatting. Useful in Word templates that format currency fields with the \# Currency picture switch.

**Category:** Advanced Data Formatting Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** CurrencyCulture

**Parameter Value:** [5-character language/country code]

**Parameter Requirements:** None

**Parameter Description:** CurrencyCulture Specifies cultural settings for currency formatting. This parameter overrides the default cultural settings for currency formatting. Useful in Word templates that format currency fields with the \# Currency picture switch.

A [language/country culture code](#) is composed of 2 lowercase letters, a hyphen, followed by 2 uppercase letters. The first two letters indicate the language; the last two indicate the country.



⚠ In the example below, de-DE specifies that the culture is for German/Germany. For Word merge fields that use the \# Currency picture switch, the currency values would automatically be formatted as Euros.

You can also use dynamic field content for the parameter value: `&CurrencyCulture={!Contact.Culture}`.

**Example:**

`&CurrencyCulture=de-DE`

## Data Gathering Parameters

### Description

These parameters help you gather data.

For more information about parameters in general and how to use them, see [Adding Parameters to a Solution](#), [A Few Rules About Parameters](#), and [How Parameters are Formatted](#).

### Parameters

- [ContactID](#) defines the related Contact drop-down menu in the Conga Composer Interface.
- [ID](#) sets the master object ID.
- [QueryID](#) retrieves and executes the SOQL query stored in the Salesforce record and makes the results available in the Template Builder window.
- [ReportID](#) retrieves the contents of the specified report and makes the report contents available in the Template Builder window.
- [DVO](#) inserts a value into the Master data set without requiring a custom field in Salesforce.
- [QVarOID](#) executes a Querystring Variable and uses the result as a value for another parameter.
- [QVar0Format](#) specifies the format of a QVar query.
- [ReportVar](#) specifies a placeholder variable for one or more reports.
- [XMD](#) excludes Master object data.

## DataTagEnable

**Category:** Conga Contracts

**Compatible with:** Composer 8

**Parameter Name:** DataTagEnable

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None


**Parameter Description:** Data Tag Enable is meant specifically for Composer solutions used with Conga Contracts. When enabled, it creates data tags in the output document that allows changes to the document to be tracked and synced to Salesforce using the True-Up feature.

When enabling this parameter, the Query Aliases in the Composer button become case-sensitive in the template document.

You must include the ID field for each relevant object included in the SELECT statement of your query to properly create the data tags in your document. For example, the below query retrieves the ID and Account. ID fields along with the data to be merged:

*IDAccount.IDFROM OpportunityWHERE Opportunity.ID = '{pv0}'*

Value	Definition
0	Data Tag disabled.
1	Data Tag enabled with bookends.

 The DataTagEnable parameter currently does not work with custom lookup fields or the TableGroup Word function.

**Example:**

&DataTagEnable=1

## DC

**Category:** Advanced Control Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DC

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** The DC parameter stands for Disable Conga (Execution). This parameter is usually set as part of a Salesforce formula to evaluate the contents of a field and, in turn, prevent the execution of Conga if the field does not meet some criteria.

 The best way to accomplish this is with a formula field in Salesforce that returns a numeric value of "1" or "0".

Value	Definition
0	Allows Composer to run (default).
1	Does not allow Composer to run.

**Example:**

```
&DC={!IF(ISPICKVAL(Opportunity.StageName, "Closed Won"), "0", "1")} or &DC={!IF(Field = value, 1, 0)}
```

## DCL

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DCL

**Parameter Value:** [text]

**Parameter Requirements:** &DC=1 (Disable Conga Execution).

**Parameter Description:** DCL stands for Disable Conga Execution Label. This parameter displays a message on the Conga Composer dialog box when Disable Conga Execution is enabled. Displays up to 50 characters.

**Example:**

```
&DC=1 or &DCL=Required+fields+have+not+been+set
```

## DefaultLocal

**Category:** Template Selection Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DefaultLocal

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Setup options configured to allow Local Templates.

**Parameter Description:** DefaultLocal sets the default source for a template to the local computer.

Value	Definition
0	Default template source is set to Choose a template from <a href="#">Salesforce.com</a> .
1	Default template source is Choose a template from your local computer.

**Example:**

&DefaultLocal=1

## DefaultPane

**Category:** Template Selection Parameter

**Compatible with:**

**Parameter Name:** DefaultPane

**Parameter Value:** [0 or 1]

**Parameter Requirements:** SelectTemplates

**Parameter Description:** DefaultPane sets Salesforce Content as the default option in the Select Templates view of the Composer dialog box.

Value	Definition
0	Conga Templates is the default option. (Default)
1	Salesforce Content & Documents is the default option, regardless of what the <a href="#">HideContent</a> parameter is set to.

**Example:**

&SelectTemplates=1 or &DefaultPane=1

## DefaultPDF

**Category:** Output Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** DefaultPDF

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DefaultPDF sets the default output type to Adobe Acrobat.

The user may override this choice unless the Lock Output to PDF parameter (**FPO=1**) is enabled.

 On computers and tablets using Apple's operating system, PDF files may appear empty when opened using Preview. To avoid this, open PDF files on Apple computers using Adobe Reader.

Value	Definition
0	Output Format is Same as template. (default)
1	Output Format is Adobe Acrobat.

**Example:**

&DefaultPDF=1

## DisableRowNums

**Category:** Word, Excel, PPTX and PDF Parameter

**Compatible with:** Composer 8

**Parameter Name:** DisableRowNums

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DisableRowNums disables the ROW\_NUMBER field in all detailed datasets. If this parameter is enabled, the ROW\_NUMBER field (used to number the results from report, query, OppLineItems or QuoteLineItems datasets) will not display in any detail datasets.

Value	Definition
0	The ROW_NUMBER field appears in all detailed datasets. (default)

Value	Definition
1	The ROW_NUMBER field does not display in any detail datasets.

**Example:**

&DisableRowNums=1

## DocFolderId

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** DocFolderId

**Parameter Value:** [Salesforce Documents Folder ID]

**Parameter Requirements:** None

**Parameter Description:** DocFolderId specifies the location where documents will be saved. More specifically, this parameter specifies the Salesforce Documents folder in which the merged output file is stored.

When utilizing Save a Copy in Composer and specifying Salesforce Documents as the Save a Copy location, you may optionally include this parameter to specify in which Documents folder the merged output file is saved. In the absence of this parameter when saving a copy to Documents, the merged output file is saved to the My Personal Documents folder.

 This parameter is required when using the `&QMode=Documents` when using Conga Batch or Conga Trigger.

**Example:**

&DocFolderID=00lj0000001EUwF

## DocuSign7

**Category:** DocuSign Integration Parameter


**Compatible with:** Composer 8

**Parameter Name:** DocuSign7

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DocuSign7 determines the DocuSign for Salesforce version used with a Composer solution.

 This parameter is primarily used for Composer solutions integrated with DocuSign for Salesforce in Salesforce Communities. It is not required for solutions in Salesforce.

Value	Definition
0	All other versions of DocuSign for Salesforce preceding DocuSign for Salesforce 7
1	DocuSign for Salesforce 7

**Example:**

&DocuSign7=1

## DocuSignAuthoritativeCopy

**Parameter:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignAuthoritativeCopy

**Parameter Value:** [1 or 0]

**Parameter Requirements:** None

**Parameter Description:** DocuSignAuthoritativeCopy enables the eOriginal eVault feature to store the Conga document in the eVault location with DocuSign. If using eOriginal service with DocuSign Connected Service (publishes an event to eOriginal, informing the eOriginal system that a transaction that needs to be vaulted is complete). This parameter specifies that this document goes into the eVault.

**Example:**

&DocuSignAuthoritativeCopy=1

## DocuSignBrandName

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignBrandName

**Parameter Value:** [BrandName]

**Parameter Requirements:** &DS7=17 or &QMode=7 (to enable Automatic DocuSign delivery via the API) and &DocuSignEndpoint.

**Parameter Description:** DocuSignBrandName specifies the name of the brand (as defined in DocuSign) that a DocuSign Envelope should display in outbound emails.

⚠ Ensure that you reference the correct DocuSign Endpoint value in the [DocuSignEndpoint parameter](#). Branding will not appear if the wrong DocuSign endpoint is referenced. If branding does not show up with &DocuSignEndpoint=Prod, try using NA2 or NA3 as the parameter value. It is also recommended to contact DocuSign support to obtain the correct endpoint.

⚠ Due to evolving integration challenges with Conga and DocuSign, the Conga Support Team is no longer able to recreate full DocuSign functionality in test Salesforce environments. This limits the Conga Support Team's capability to test and achieve desired results for this parameter.

**Example:**

&DS7=17 or &DocuSignEndpoint=Prod or &DocuSignBrandName=Conga

## DocuSignCustomField

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignCustomField


**Parameter Value:** [ApiFieldName]

**Parameter Requirements:** DocuSignCustomFieldValue

**Parameter Description:** DocuSignCustomField sets the name of a custom field to set on the DocuSign Envelope Record created through the merge process. Use the Salesforce API name of the field on the DocuSign Envelope Custom Object. This parameter is designed to be used for those DocuSign Envelope fields that do not already have a corresponding Conga Composer parameter.



You can use a total of 12 DocuSign custom fields and values with the format DocuSignCustomField, DocuSignCustomField1, DocuSignCustomField2, ... DocuSignCustomField10, DocuSignCustomField11.

 The DocuSignCustomField parameter does not update or write back to custom fields in DocuSign. This parameter only updates field values for custom fields on the DocuSign Envelope Object in Salesforce.

**Example:**

```
&DocuSignCustomField=Account__c  
&DocuSignCustomFieldValue=001ABCEDFGHIJL
```

## DocuSignCustomFieldValue

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignCustomFieldValue

**Parameter Value:** [value]

**Parameter Requirements:** DocuSignCustomField

**Parameter Description:** DocuSignCustomFieldValue sets the value of the corresponding custom field.

You can use a total of 12 DocuSign custom fields and values with the format DocuSignCustomFieldValue, DocuSignCustomFieldValue1, DocuSignCustomFieldValue2, ... DocuSignCustomFieldValue10, DocuSignCustomFieldValue11.

This parameter updates a field value on the DocuSign Envelope record in Salesforce. It does not update any fields in the DocuSign for Salesforce UI.

**Example:**

```
&DocuSignCustomField=Account__c or &DocuSignCustomFieldValue=001ABCEDFGHIJL
```

## DocuSignEmailBody

**Category:** DocuSign Integration Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignEmailBody

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** DocuSignEmailBody populates the body of the DocuSign email.

 Because this value is transmitted through the Conga Composer window URL, we recommend that this value be less than 40 characters.

**Example:**

`&DocuSignEmailBody=Please+sign+this+document.`

## DocuSignEmailSubject

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignEmailSubject

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** DocuSignEmailSubject assigns a custom email subject. By default, the email subject sent through DocuSign is pulled from the DocuSign Admin Settings. With this parameter, you can override the Admin Settings and establish a custom value for the email subject.

**Example:**

`&DocuSignEmailSubject=Contract`

## DocuSignEnableWetSign

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignEnableWetSign

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DocuSignEnableWetSign enables the Print & Sign option on the DocuSign envelope. The Print & Sign functionality allows the recipient of the document to print and sign the document and then upload or fax the document back into DocuSign.

Value	Definition
0	DocuSign Print & Sign is not enabled. (default)
1	DocuSign Print & Sign is enabled.

**Example:**

```
&DocuSignEnableWetSign=1
```

## DocuSignEndpoint

**Category:** DocuSign Integration Parameter

**Compatible with:** Conga Composer 8 and Conga Composer 7


**Parameter Name:** DocuSignEndpoint


**Parameter Value:** [Demo, Test, Preview, EU, NA2, Prod, AU]

**Parameter Requirements:** None

**Parameter Description:** DocuSignEndpoint configures and directs a Composer button to a specific DocuSign Endpoint (or subdomain) other than the DocuSign production endpoint.

This value is pulled from the DocuSign URL. For example, for a DocuSign URL of `https://demo.docusign.net`, the parameter value would be a `demo`.

 This advanced parameter is typically used only by DocuSign account engineers.

 This parameter is required when processes run completely in background mode, for example when using `DS7=17` or `QMode=DocuSign`. If you are not using these parameters, only use this parameter if your endpoint is something other than Production, such as a DocuSign developer account.

**Example:**

```
&DocuSignEndpoint=Demo
```

## DocuSignExpireIn

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignExpireIn

**Parameter Value:** [0 to 999]

**Parameter Requirements:** None

**Parameter Description:** DocuSignExpireIn indicates the number of days until the transaction is expired

**Example:**

&DocuSignExpireIn=30

## DocuSignExpireVoidEnv

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignExpireVoidEnv

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DocuSignExpireVoidEnv indicates whether the envelope automatically expires.

Value	Definition
0	The envelope will not automatically expire.
1	The envelope will automatically expire. (default)

**Example:**

&DocuSignExpireVoidEnv=0

## DocuSignIgnoreOCR

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignIgnoreOCR

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** The DocuSignIgnoreOCR ignores Opportunity Contact Roles.

When Composer creates a DocuSign Envelope from an Opportunity, it automatically scans the list of Opportunity Contact Roles and attempts to match DocuSign Role Names with the Contact Roles. If they match, the Contact is automatically included as a signer. When this parameter is enabled, this behavior is skipped and Conga Composer ignores the Opportunity Contact Roles.

In using this parameter, you must explicitly populate the DocuSign Recipient parameters to pre-address the Envelope. At this time, this feature only works in earlier versions of DocuSign for Salesforce (versions 6.9 and earlier). This feature currently does not work with DocuSign for Salesforce versions 7 and greater.

Value	Definition
0	Opportunity Contact Roles will automatically be attempted to be matched with DocuSign Roles. (default)
1	Opportunity Contact Roles will be ignored and therefore not matched to DocuSign Roles.

**Example:**

&DocuSignIgnoreOCR=1

## DocuSign Integration Parameters

DocuSign delivers Conga Composer documents via email, much like Salesforce itself, but adds the ability to acquire, track, and file signatures using [automatic anchor text and tags](#).

DocuSign lets you send documents electronically to collect electronic signatures from a recipient and update relevant data in your Salesforce account. Your recipient's sign, enter data, initial, and date your documents online and can update Salesforce automatically with signed documents and collected data. For more information visit <http://sites.force.com/appexchange/listingDetail?listingId=a0N30000001taX4EAI>

DocuSign is available separately on the AppExchange. You must subscribe to that service to use these integration parameters.

When using Conga products and DocuSign in a fully automated fashion (such as using the background parameter/value pair &DS7=17, Conga Trigger, or Conga Batch) the DocuSign Username and Salesforce User Email must match. If not, you will receive this error: Error: CreateEnvelope in DocuSignAPI. The UserID does not have a valid membership in this Account.

## Parameters

- [DocuSign7](#) specifies the version of DocuSign used with Composer.
- [DocuSignBrandName](#) specifies the brand name to use in DocuSign emails.
- [DocuSignCustomField](#) specifies the name of a DocuSign custom field.
- [DocuSignCustomFieldValue](#) specifies the value for a DocuSign custom field.
- [DocuSignEmailSubject](#) specifies the email subject.
- [DocuSignEmailBody](#) specifies text to appear in the body of the email.
- [DocuSignEnableWetSign](#) enables the Print & Sign option on the DocuSign envelope.
- [DocuSignEndpoint](#) enables a specific DocuSign Endpoint account to connect to the Composer button.
- [DocuSignExpireIn](#) specifies the number of days until the transaction expires.
- [DocuSignExpireVoidEnv](#) specifies whether the email envelope will automatically expire.
- [DocuSignIgnoreOCR](#) specifies whether to correlate Opportunity Contact Roles with DocuSign Roles.
- [DocuSignR1AccessCode](#) specifies a recipient's access code.
- [DocuSignR1Email](#) enables the Quick Add Recipient functionality with Conga.
- [DocuSignR1EmailBlurb](#) specifies the email Blurb for each specified recipient.
- [DocuSignR1EmailSubject](#) specifies the email subject for each of the recipients.
- [DocuSignR1ID](#) specifies the email recipient.
- [DocuSignR1IdCheck](#) initiates the DocuSign ID Check functionality in Conga.
- [DocuSignR1IdCheckType](#) specifies the authentication method to use for the DocuSign ID Check functionality in Conga.
- [DocuSignR1IdCheckSMSPhone](#) specifies a mobile phone number for an SMS authentication using the DocuSign ID Check functionality.
- [DocuSignR1Language](#) specifies the email language for each specified recipient.
- [DocuSignR1Name](#) enables the Quick Add Recipient functionality with Conga.
- [DocuSignR1Role](#) specifies the role for each specified recipient.
- [DocuSignR1RoutingOrder](#) specifies the routing order for each specified recipient.
- [DocuSignR1SignNow](#) enables the DocuSign Sign Now functionality with Conga.
- [DocuSignR1SigningGroup](#) enables the DocuSign Signing Group functionality with Conga.
- [DocuSignR1SIPName](#) specifies the Sign In-Person name on the envelope.
- [DocuSignR1Type](#) specifies the recipient type.
- [DocuSignRelatedAccountID](#) specifies the Account ID field on the DocuSign envelope.
- [DocuSignRemindAgainIn](#) specifies the reminder repeat frequency.
- [DocuSignRemindIn](#) specifies the number of days until the first reminder email is sent.

- [DocuSignSendAsUserID](#) specifies the email sender's ID.
- [DocuSignSendReminders](#) specifies whether to send email reminders.
- [DocuSignTag](#) redirects the browser to the DocuSign screen after a completed merge.
- [DocuSignVisible](#) includes DocuSign in the Composer dialog box Action picklist.
- [DocuSignWarnIn](#) specifies the number of days prior to expiration that Composer should send a reminder email.

## DocuSignR1AccessCode

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1AccessCode

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1AccessCode sets the DocuSign Access Code for Recipient 1

This parameter can be used `{{DocuSignUsers}}` times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR`{{DocuSignUsers}}`.

**Example:**

```
&DocuSignR1AccessCode=ABC123
```

## DocuSignR1Email

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1Email

**Parameter Value:** [text]

**Parameter Requirements:** DS7=17 or DocuSignTag=1 and DocusignEndpoint=Prod

**Parameter Description:** DocuSignR1Email (DocuSign Quick Add Recipient) enables the Quick Add Recipient functionality with Conga. This allows users to add an email address (and a name) of a recipient who is not a User, Contact, or Lead in their Salesforce org.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

Both &DocuSignR1Email and &DocuSignR1Name are required to use Quick Add Recipient. Not applicable to DocuSign for Salesforce (DfS). To use these parameters, contact DocuSign to enable these features in your production Salesforce Org. Additional fees for these DocuSign features may be required; contact DocuSign to learn more.

**Example:**

```
&DS7=17 &DocuSignR1Name=[FirstName+LastName] &DocuSignR1Email=[text]
```

## DocuSignR1EmailBlurb

**Category:** DocuSign Integration Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1EmailBlurb

**Parameter Value:** [text]

**Parameter Requirements:** Automatic DocuSign delivery via the API using &DS7=17 or &QMode=7.

**Parameter Description:** DocuSignR1EmailBlurb (DocuSign Email Blurb for Recipient 1) specifies the DocuSign Email Blurb for each DocuSign Recipient.

 Warning: Short pieces of text can be passed directly through the parameter, but because this value is transmitted through the Conga Composer window URL, we recommend using &QueryStringField rather than passing a large block of text. This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

**Example:**

```
&DS7=17 &DocuSignR1EmailBlurb=Please+sign+by+March+15
```

## DocuSignR1EmailSubject

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1EmailSubject

**Parameter Value:** [text]



**Parameter Requirements:** Automatic DocuSign delivery via the API using &DS7=17 or &QMode=7.

**Parameter Description:** DocuSignR1EmailSubject (DocuSign Email Subject for Recipient 1) specifies the Email Subject for the DocuSign Recipient. You can specify the email subject for each DocuSign Recipient that you identified.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

This value is passed through the Conga Composer window URL and should be less than 40 characters.

**Example:**

&DS7=17 &DocuSignR1EmailSubject=Contract

## DocuSignR1ID

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** DocuSignR1ID

**Parameter Value:** [Salesforce Contact, Lead, or User ID]

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1ID sets the recipient of the document to be delivered.

This parameter can be used 24 times by changing the number after R. For example, DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

In the case of an Opportunity (as the master object) with Contact Roles, the contacts whose Contact Role values match the DocuSign Roles will automatically be included as document recipients unless the [&DocuSignIgnoreOCR](#) parameter is enabled.

Account ID works in Lightning as Lightning Account ID and Contact ID. If you use the DS7=7 parameter in a DocuSign-integrated Composer button, the routing order defined by DocuSignR1ID parameter appears out of order on the DocuSign envelope and custom anchor tags are not assigned to the correct recipients. This is a result of a DocuSign API limitation. To resolve the incorrect ordering on the DocuSign envelope, drag the first row slightly upward using the dotted area. This issue does not occur when DS7=17 is used to bypass the DocuSign user interface.

**Example:**

&DocuSignR1ID=003ABCD12345

## DocuSignR1IDCheck

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1IDCheck

**Parameter Value:** [ID]

**Parameter Requirements:** DocuSignR1IdCheck and DocuSignR1IdCheckType are both required to use the ID check feature. If DocuSignR1IdCheckType is SMS, then DocuSignR1IdCheckSMSPhone is also required. This parameter also requires DocuSignTag or a fully automated solution (DS7=17, Conga Trigger, or Conga Batch). Available for international numbers see the example.

**Parameter Description:** DocuSignR1IDCheck enables the DocuSignIdCheck functionality with Conga.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

DocuSign ID Check Parameters provide an additional layer of security in confirming that the recipient of the DocuSign Envelope is the correct person.

To use these parameters, contact DocuSign to enable these features in your production Salesforce Org. Additional fees for these DocuSign features can be required; contact DocuSign to learn more.

Value	Definition
0	DocuSign ID Check functionality is disabled. (Default)
1	DocuSign ID Check functionality is enabled.

**Example:**

```
&DS7=17
&DocuSignR1IdCheck=1
&DocuSignR1IdCheckType=RSA
```

## DocuSignR1IDCheckSMSPhone

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignR1IDCheckSMSPhone

**Parameter Value:** [mobile phone number]

**Parameter Requirements:** DocuSignR1IdCheck and DocuSignR1IdCheckType are both required to use the ID check feature. If DocuSignR1IdCheckType is SMS, then DocuSignR1IdCheckSMSPhone is also required. This parameter also requires DocuSignTag or a fully automated solution (DS7=17, Conga Trigger, or Conga Batch).

**Parameter Description:** Specifies a mobile phone number to complete an SMS Authentication using the DocuSignIdCheck functionality.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

DocuSign ID Check Parameters provide an additional layer of security in confirming that the recipient of the DocuSign Envelope is the correct person.

This parameter is available for international numbers - see the example below.

To use these parameters, contact DocuSign to enable these features in your production Salesforce Org. Additional fees for these DocuSign features may be required; contact DocuSign to learn more.

**Example:**

&DocuSignTag=1

&DocuSignR1IdCheck=1

&DocuSignR1IdCheckType=SMS

&DocuSignR1IdCheckSMSPhone=3035551212

**Or**

&DocuSignTag=1

&DocuSignR1IdCheck=1

&DocuSignR1IdCheckType=SMS

&DocuSignR1IdCheckSMSPhone=+(61)(448011069)

## DocuSignR1IDCheckType


**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1IDCheckType

**Parameter Value:** [RSA or SMS]

**Parameter Requirements:** DocuSignR1IdCheck and DocuSignR1IdCheckType are both required to use the ID check feature. If DocuSignR1IdCheckType is SMS, then DocuSignR1IdCheckSMSPhone is also required. This parameter also requires DocuSignTag or a fully automated solution (DS7=17, Conga Trigger, or Conga Batch).


 This parameter is available for use with international numbers, see the example below.

**Parameter Description:** DocuSignR1IdCheck (DocuSign Recipient ID Check Type for Recipient 1) provides ID Check or SMS text authentication for DocuSign documents.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

DocuSign ID Check Parameters provide an additional layer of security in confirming that the recipient of the DocuSign Envelope is the correct person.

To use these parameters contact DocuSign to enable these features in your production Salesforce Org.

 Additional fees for these DocuSign features might be required; contact DocuSign to learn more.

Value	Definition
RSA	Initiates the ID Check Authentication option, and the recipient is prompted to complete an identity check.  RSA corresponds to KBA in DocuSign.
SMS	Initiates the SMS Authentication, where the recipient is prompted to select a mobile phone number to use for authentication.  If SMS is used, a mobile phone number must be specified using DocuSignR1CheckSMSPhone .

**Example:**

&DocuSignR1IdCheck=1

&DocuSignR1IdCheckType=SMS

&DocuSignR1IdCheckSMSPhone=3035551212

## DocuSignR1Language

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignR1Language

**Parameter Value:** [2 or 5-character DocuSign language code]

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1Language specifies the email language for each DocuSign Recipient.

- ✓ Conga recommends using Composer without background mode (&DS7=17) for translation with DocuSign.

When using the DocuSignR1Language parameter with &DS7=17 (background mode), DocuSign does not provide the translation. Administrators must provide the translated text as the values for each recipient's DocuSignR1Email and DocuSignR1EmailBlurb parameters.

- Example of Translated Text when using DS7:

```
&DocuSignR1EmailSubject=Contrato &DocuSignR1EmailBlurb=Por+favor+firme
```

For non-background mode solutions, administrators must add each language used with the DocuSignR1Language parameter in their DocuSign account by navigating to DocuSign Admin → Settings → Email Settings, to successfully translate the Email Subject and Email Body fields.

- ⚠ To translate fields other than the Email Subject and Email Body, such as the Review Document field, recipients must have their browser set to the intended language. For more information, see the related [DocuSign customer help thread](#).


- ⚠ **Warning:** Due to an issue with DocuSign, you must select a different language and then re-select the intended language for each recipient on the DocuSign Envelope. This allows the Email Subject and Email Body fields to successfully translate.

For more information on parameters supported with DocuSign and &DS7=17, see [DS7](#) and [Using Conga Composer with DocuSign in Background Mode for Multiple Signers](#).

Acceptable parameter values for non-background mode Composer solutions:

- Arabic (ar)
- Bahasa Indonesia (id)
- Bahasa Melayu (ms)
- Bulgarian (bg)
- Czech (cs)
- Chinese Simplified (zh\_CN)
- Chinese Traditional (zh\_TW)
- Croatian (hr)
- Danish (da)
- Dutch (nl)
- English US (en)
- English UK (en\_GB)
- Estonian (et)
- Farsi (fa)
- Finnish (fi)
- French (fr)
- French Canada (fr\_CA)
- German (de)
- Greek (el)
- Hebrew (he)
- Hindi (hi)
- Hungarian (hu)
- Italian (it)
- Japanese (ja)
- Korean (ko)
- Latvian (lv)
- Lithuanian (lt)
- Norwegian (no)
- Polish (pl)
- Portuguese (pt)
- Portuguese Brazil (pt\_BR)
- Romanian (ro)
- Russian (ru)
- Serbian (sr)
- Slovak (sk)

- Slovenian (sl)
- Spanish (es)
- Spanish Latin America (es\_MX)
- Swedish (sv)
- Thai (th)
- Turkish (tr)
- Ukrainian (uk)
- Vietnamese (vi)

 This parameter has different a naming convention for Composer solutions &DS7=17. When using &DS7=17, use a hyphen instead of an underscore for country-specific language codes. For example, the naming convention for Spanish Latin America in a solution with &DS7=17 is `es-mx`, instead of `es_MX`.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

 You can specify a language parameter for each recipient you identified.

#### Example:

`&DocuSignR1Language=fr`

## DocuSignR1Name

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignR1Name

**Parameter Value:** [Firstname+Lastname]

**Parameter Requirements:** DS7=17 or DocuSignTag=1

**Parameter Description:** DocuSignR1Name (DocuSign Quick Add Recipient) enables the Quick Add Recipient functionality with Conga. This allows users to add a name (and email address) of a recipient who is not a User, Contact, or Lead in their Salesforce org.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

Both `&DocuSignR1Email` and `&DocuSignR1Name` are required to use Quick Add Recipient. Not applicable to DocuSign for Salesforce (DfS). To use these parameters contact DocuSign to

enable these features in your production Salesforce Org. Additional fees for these DocuSign features may be required; contact DocuSign to learn more.

**Example:**

```
DS7=17  
&DocuSignR1Email=[text]  
&DocuSignR1Name=[FirstName+LastName]
```

## DocuSignR1Role

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignR1Role

**Parameter Value:** [text]

**Parameter Requirements:** A valid DocuSign Role Name, as defined in Envelope Settings.

**Parameter Description:** DocuSignR1Role sets the role for the specified DocuSign recipient.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

- ✔ It is recommended to use this parameter when using Automatic Anchor tags or text in a template. If this parameter is not used the Automatic Anchor tags or text will not correctly populate when the document is sent.

**Example:**

```
&DocuSignR1Role=Signer+1
```

## DocuSignR1RoutingOrder

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1RoutingOrder


**Parameter Value:** [1, 2, 3, etc.]

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1RoutingOrder (DocuSign Routing Order for Recipient 1) sets the recipient routing order.



This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

 If you use the DS7=7 parameter in a DocuSign-integrated Composer button, the routing order defined by DocuSignR1ID parameter appears out of order on the DocuSign envelope and custom anchor tags are not assigned to the correct recipients. This is a result of a DocuSign API limitation. To resolve the incorrect ordering on the DocuSign envelope, drag the first row slightly upward using the dotted area. This issue does not occur when DS7=17 is used to bypass the DocuSign user interface.

**This issue does not occur in DocuSign for Salesforce version 7.3 or greater.**

#### Example:

&DocuSignR1RoutingOrder=3

## DocuSignR1SigningGroup

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8


**Parameter Name:** DocuSignR1SigningGroup

**Parameter Value:** Name of signing Group

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1SigningGroup enables the DocuSign Signing Group functionality with Conga. This allows users to be placed in signing groups.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

 Due to evolving integration challenges with Conga and DocuSign, the Conga Support Team is no longer able to recreate full DocuSign functionality in test Salesforce environments. This limits the Conga Support Team's capability to test and achieve desired results for this parameter.

The composer must be in API mode for this parameter to work. This means you need to use [&DS7=17](#) or send the DocuSign Agreement using Conga Batch or Conga Trigger ([QMode=DocuSign](#)).

This parameter can only be assigned to ONE of the intended recipients. Assigning it to more than one recipient in a group will not work. To use these parameters contact DocuSign to

enable these features in your production Salesforce Org. Additional fees for these DocuSign features may be required; contact DocuSign to learn more.

**Example:**

`&DocuSignRISigningGroup=test_SigningGroup`

## DocuSignRISignNow

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignRISignNow

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Enables the DocuSign Sign Now functionality with Conga. To be used in conjunction with the DocuSign Sign in Person feature (see [&DocuSignRISIPName](#)). When `&DocuSignRISignNow` is used, clicking Send Now redirects the user to the DocuSign signing experience.

 This parameter is only supported for the first signer/recipient.

Only applicable to DocuSign for Salesforce (DfS); not for use with DocuSign tag parameters or `&DS7=17`, `&QMode=DocuSign` (Conga Batch/Conga Trigger), or `&APIMode=17`. To use these parameters contact DocuSign to enable these features in your production Salesforce Org. Additional fees for these DocuSign features may be required; contact DocuSign to learn more.

Value	Definition
0	DocuSign Sign Now functionality is disabled. (Default)
1	DocuSign Sign Now functionality is enabled.

**Example:**

`&DocuSignRISignNow=1`

## DocuSignR1SIPName

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignR1SIPName

**Parameter Value:** [text]

**Parameter Requirements:** &DocuSignR1Type=Sign+in+person

**Parameter Description:** DocuSignR1SIPName sets the Sign In Person Name field on the Envelope Recipient.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

**Example:**

```
&DocuSignR1SIPName=Tony+Stark
```

```
&DocuSignR1Type=Sign+in+person
```

## DocuSignR1Type

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignR1Type

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** DocuSignR1 Type (DocuSign Recipient Type for Recipient 1) sets the recipient type.

This parameter can be used 24 times by changing the number after R. DocuSignR2, DocuSignR3, DocuSignR4, ... DocuSignR24.

When used with the DocuSignR1ID parameter, the DocuSignR1Type parameter is not required using earlier versions of DocuSign for Salesforce (versions 6.9 and earlier). The DocuSignR1Type parameter is required with DocuSign for Salesforce versions 7 and greater. If it is not used, there was an issue processing the Composer request error message is returned.

Value	Description
Signer	The recipient must sign, initial, date, or add data to tags on the documents in the envelope. The Signer must be capitalized.
Carbon+Copy	The recipient should get a copy of the envelope, but does not need to sign, initial, date, or add information to any of the documents.
Certified+Delivery	The recipient must receive the completed documents for the envelope to be completed but does not need to sign, initial, date, or add information to any of the documents.
Sign+in+person	<p>The recipient will sign the document in person with the DocuSign administrator. The DocuSign Envelope will be emailed to the DocuSign administrator specified with the corresponding DocuSignRxSIPName parameter Salesforce UserID value.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>Important</b></p> <p>The following parameter is required to use Sign+in+person as the value for DocuSignRIType: <b>DocuSignRxSIPName=[Text]</b></p> </div>
Address+recipients	The recipient can add name and email information for recipients who appear after the recipient in the Conga Trigger order.
Manage+envelope	The recipient has the same management and access rights for the envelope as the sender and can make changes to the envelope as if using the Advanced Correct feature. The recipient can add name and email information, add or change the routing order, and set authentication options for the remaining recipients. Additionally, this recipient can edit signature/initial tabs and data fields for the remaining recipients.

**Example:**

```
&DocuSignRIType=Signer
```

## DocuSignRelatedAccountId

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** DocuSignRelatedAccountId

**Parameter Value:** [Salesforce Account ID]

**Parameter Requirements:** Automatic DocuSign delivery via the API using &DS7=17 or &QMode=7.

**Parameter Description:** DocuSignRelatedAccountId sets the related Account field on the DocuSign Envelope. This parameter is used in establishing the relationship to the related Account on the DocuSign Status record.

**Example:**

```
&DS7=17  
&DocuSignRelatedAccountId=001i000000oJaIM
```

## DocuSignRemindAgainIn

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignRemindAgainIn

**Parameter Value:** [0 to 999]

**Parameter Requirements:** None

**Parameter Description:** DocuSign Remind Again In indicates how often (in days) the reminder should repeat until the transaction is complete.

**Example:**

```
&DocuSignRemindAgainIn=5
```

## DocuSignRemindIn

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignRemindIn

**Parameter Value:** [0 to 999]

**Parameter Requirements:** None

**Parameter Description:** DocuSignRemindIn indicates how many days should pass before the first reminder is sent.

**Example:**

&DocuSignRemindIn=2

## DocuSignSendAsUserId

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignSendAsUserId

**Parameter Value:** [UserId]

**Parameter Prerequisites:** You must enable the UserOverrideEnabled flag into your DocuSign Account. Click [here](#) for more information on how to enable this flag.

**Parameter Requirements:** Automatic DocuSign delivery through the API using &DS7=17 or &QMode=DocuSign

**Parameter Description:** DocuSignSendAsUserId specifies a User Id from which DocuSign Envelopes are sent. This setting only applies if an individual user does not have credentials specified on his or her User record. The user without credentials can use the Send On Behalf Of feature and use a different user's credentials.

**Example:**

&DS7=17  
&DocuSignSendAsUserId=00536000002MDP3

## To enable UserOverrideEnabled flag

1. Log In to your **DocuSign Account** on the DocuSign website.
2. Go to the **Settings** tab.
3. Click the **Reminders and Expiration** option.
4. Check **Allow senders to override account defaults** checkbox.
5. Click **Save**.

## DocuSignSendReminders

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignSendReminders

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DocuSignSendReminders sends reminder emails to recipients who have not yet completed their task.

Value	Definition
0	Email reminders are not sent. (default)
1	Email reminders are sent.

**Example:**

&DocuSignSendReminders=1

## DocuSignTag

**Category:** DocuSign Integration Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignTag

**Parameter Value:** [0 or 1]

**Parameter Requirements:** The Conga Template ID must be set using &TemplateID.

**Parameter Description:** DocuSignTag enables the DocuSign Tag. After the merge operation is complete, the browser is redirected to the DocuSign Tag screen.

 DocuSignTag=1 will not work with DS7=17 for automatic delivery. Also, the [&DocuSignEndpoint](#) parameter must be configured properly to use this parameter.

Value	Definition
0	Does not redirect to the DocuSign Tag screen. (default)
1	After the merge is complete, redirects to the DocuSign Tag screen.

**Example:**

&TemplateID=a0WG0000006PNrS  
&DocuSignTag=1

# DocuSignVisible

**Category:** DocuSign Integration Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** DocuSign Visible changes the Action picklist in the Composer dialog box to DocuSign and updates the Merge button to Merge and DocuSign. DocuSign is available separately on the AppExchange.

 You must subscribe to that service to use this integration parameter.

Value	Definition
0	DocuSign is not the designated Action. (default)
1	The Action is set to DocuSign.

**Example:**

&DocuSignVisible=1

# DocuSignWarnIn

**Category:** DocuSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DocuSignWarnIn

**Parameter Value:** [0 or 999]

**Parameter Requirements:** None

**Parameter Description:** DocuSignWarnIn indicates how many days prior to expiration you want an email message to alert recipients that a deadline approaches for completion.

**Example:**

&DocuSignWarnIn=3



## DS0

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS0

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disables controls within the Conga Composer dialog box.

Value	Definition
0	All controls in the Conga Composer dialog box (including buttons) are enabled. (default)
1	All controls in the Conga Composer dialog box (except the buttons) are disabled.

**Example:**

&DS0=1

## DS1

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS1

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disable Merge & Download Button. Enable or Disable the Download button.

The default name of the Merge button is Merge & Download but the name of it will change depending on file output options.

Value	Definition
0	The Merge & Download Button is enabled. (default)

Value	Definition
1	The Merge & Download Button is disabled. The document will still merge but will not download for the running user.

**Example:**

&DS1=1

## DS2

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS2

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Enable or disable the Email Action option in Output Options.

Value	Definition
0	The Email Action option is enabled. (default)
1	The Email Action option is disabled.

**Example:**

&DS2=1

## DS3

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS3

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Enables or disables the Activity Logging area.

Value	Definition
0	Activity Logging options are enabled. (default)
1	Activity Logging options are disabled.

Disabling Activity Logging will disable [MFTS](#).

**Example:**

&DS3=1

## DS4

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS4

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** &TemplatelD

**Parameter Description:** Locks the list of template choices to prevent end-users from selecting a different template.

In the absence of the DS4 parameter, the template indicated by the TemplatelD parameter is pre-selected in the list, but the user may select a different template. This parameter prevents the user from selecting a different template.

Value	Definition
0	The list of template choices is not locked (default).
1	The selected template choice is locked.
2	The selected template choice is locked and users can also select additional templates to merge. Using DS4=2 requires the SelectTemplates parameter.

**Example:**

&TemplatelD=a01268Hc74Vb9Mh  
&DS4=1

## DS5

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS5

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disables View Data

Value	Definition
0	View Data is enabled. (default)
1	View Data is disabled.

**Example:**

&DS5=1

## DS6

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS6

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disable View Template. View Template options include both the ability to open (download) the template file and to open the template record in Salesforce.

Value	Definition
0	View Template is enabled. (default)
1	View Template is disabled.

**Example:**


&amp;DS6=1



## DS7

**Category:** Automation-Background Mode Parameter**Compatible with:** Composer 8 and Composer 7**Parameter Name:** DS7**Parameter Value:** [value]**Parameter Requirements:** None**Parameter Description:** DS7 enables Background Mode, which initiates the Composer without user intervention. This parameter enables merge operations to occur in the background.


Value	Definition
0	Background Mode disabled. (Default)
1	Background Mode is enabled and the merged document is attached to the master object.
2	Background Mode is enabled and the merged document is attached to a Salesforce email.  If you use this parameter and want to log activities or follow-up tasks, you must also enable Automatic Logging (AC0=1).
3	Background Mode is enabled and the merged document is delivered to the browser to save or open.  If you use this parameter and want to log activities or follow-up tasks, you must also enable Automatic Logging (AC0=1).
4	Background Mode is enabled and the merged document is attached to a new Adobe eSign Agreement. (See Adobe eSign integration parameters)  If you use this parameter and want to log activities or follow-up tasks, you must also enable Automatic Logging (AC0=1).

Value	Definition
5	Background Mode is enabled and the merged document is attached to the master object (like DS7=1), except that the user can navigate away from the master object while the merge process occurs.
6	Background Mode is enabled and the merged document is attached to Salesforce Content.
7	<p>Background Mode is enabled and the merged document is attached to a DocuSign Envelope.</p> <p>If you use the DS7=7 parameter in a DocuSign-integrated Composer button, the routing order defined by the DocuSignRIID parameter appears out of order on the DocuSign envelope and custom anchor tags are not assigned to the correct recipients. This is a result of a DocuSign API limitation. To resolve the incorrect ordering on the DocuSign envelope, drag the first row slightly upward using the dotted area. This issue does not occur when DS7=17 is used to bypass the DocuSign user interface.</p>
8	Background Mode is enabled and the merged document is saved in a SpringCM folder.
9	Background Mode is enabled and the merged document (Microsoft Office Word, Excel, or PowerPoint) is saved to Google Drive.
11	Background Mode is enabled and the merged document is attached to Salesforce Files for the Master Object.
12	<p>Background Mode is enabled and the merged document is automatically delivered through email.</p> <p>Using DS7=12 with a Conga Email Template requires the CongaEmailTemplateID (CETID) parameter and at least one of the following the email recipient parameters: EmailToID, EmailAdditionalTo, EmailCC.</p> <p>Using DS7=12 with a Salesforce Email Template requires the EmailTemplateID parameter, as opposed to the CongaEmailTemplateID parameter, and the EmailToID parameter.</p> <p>No preview is available for this parameter. Please test your configuration thoroughly before using it.</p>

Value	Definition
13	<p>The parameter runs a merge in background mode, delivers the merged output to the user's browser, then returns the user to the Salesforce master record that initiated the merge.</p> <div data-bbox="432 477 1426 815" style="border: 1px solid red; padding: 10px;"> <p> <b>Warning</b></p> <p>This behavior is subject to browser restrictions and may not work consistently. After updating to Chrome Version 74.0.3729.108, users using a non-Oauth button URL are no longer prompted to download the merged document. The recommended workaround is to convert your button URL to an Oauth button to continue using &amp;DS7=13. Alternatively, you can switch to using &amp;DS7=3.</p> </div>
14	<p>Background Mode is enabled and the merged document is automatically delivered through Adobe eSign.</p> <p>No preview is available for this parameter. Please test your configuration thoroughly before using it.</p>
17	<p>Background Mode is enabled and the merged document is attached to a DocuSign Envelope and automatically delivered.</p> <p>When using Conga products and DocuSign in a fully automated fashion (such as using the background parameter/value pair &amp;DS7=17, Conga Trigger, or Conga Batch) the DocuSign Username and Salesforce User Email must match. If not, you receive the error: Error: CreateEnvelope in DocuSignAPI. The UserID does not have a valid membership in this Account.</p> <p>DocuSignTag=1 does not work with DS7=17 for automatic delivery.</p> <p>Prerequisites for using DS7=17</p> <ul style="list-style-type: none"> <li>• DocuSignEndPoint=Prod (or Demo)</li> <li>• DocuSignRIID</li> <li>• DocuSignRIRole</li> <li>• DocuSignRIType</li> <li>• DocuSign Credentials on the USER record</li> <li>• Same email address in Salesforce as in DocuSign</li> </ul>
19	<p>Background Mode is enabled and the merged document (Google Docs version) is saved to Google Drive.</p>

Value	Definition
24	<p>Background Mode is enabled and the merged document is automatically delivered through Adobe eSign. The browser is redirected to the Adobe eSign Confirmation Page. (Requires Adobe eSign version 9.5 or higher.)</p> <p>No preview is available for this parameter. Please test your configuration thoroughly before using it.</p>
111	<div data-bbox="432 591 1426 734" style="border: 1px solid #f9e79f; padding: 10px; margin-bottom: 10px;"> <p> <b>Important</b></p> <p>Specific value that integrates Composer and Contracts features.</p> </div> <p>Background mode is enabled and a Composer generated document is sent directly to Send for Negotiation email service in Contracts. The template is merged. The user is redirected to the email preview screen. The user can edit or verify the information before sending the negotiation email. A copy of the document is attached to the record as a Salesforce File.</p>
112	<div data-bbox="432 985 1426 1128" style="border: 1px solid #f9e79f; padding: 10px; margin-bottom: 10px;"> <p> <b>Important</b></p> <p>Specific value that integrates Composer and Contracts features.</p> </div> <p>Background mode is enabled and merges the template. The email is sent automatically without any additional user interaction. A copy of the document is attached to the record as a Salesforce File.</p> <p>Using DS7=112 requires the <i>congaemailtemplateid</i> and EmailTold parameters.</p>
301	<p>Background Mode is enabled and the merged document is automatically delivered through Sertifi. The browser is redirected to the Sertifi Econtract. (Requires Sertifi version 5.87 or higher.)</p> <p>No preview is available for this parameter. Please test your configuration thoroughly before using it.</p>
302	<p>Background Mode is enabled and the merged document is automatically delivered via Sertifi. The browser is redirected to the Master Object. (Requires Sertifi version 5.87 or higher.)</p> <p>No preview is available for this parameter. Please test your configuration thoroughly before using it.</p>



Value	Definition
401	Background Mode is enabled and the merged document is attached to Chatter.
501	Background Mode is enabled and the merged document is automatically sent to OneSpan Sign.
550	The background is enabled, Composer Link Service generates an online document link, and the user is taken to the email preview screen.
551	Background mode is enabled, Composer Link Service generates an online document link, and the email is sent with the online document link included.
552	Background mode is enabled, Composer Link Service generates an online document link, and no email is sent.
600	Background Mode is enabled and the merged document is automatically sent to Box.
700	Background Mode is enabled and the merged document is automatically sent to OneDrive.
900	Background mode is enabled and the merged document is automatically sent to BigTinCan.
1141	Background mode is enabled and the merged document is forwarded to Conga Sign at end of the merge, and a copy is saved under Salesforce Files.
1142	Background mode is enabled and the merged document is sent to the Conga Sign API, and a copy is saved under Salesforce Files.  <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">  This parameter value requires the &amp;CSRecipient parameter. </div>

**Example:**

```
&DS7=1
```

## DS7Preview

**Category:** Automation-Background Mode Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS7Preview

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** The [DS7 parameter](#), set to anything except 3 or 13.

**Parameter Description:** DS7Preview (Background Mode Preview) allows the end-user to preview the merged file before delivering the file according to the specific DS7 mode, preview each page of the merged output file, and choose options to download the file.

This parameter is not compatible with the CongaEmailTemplateId parameter. The DS7Preview parameter is compatible with Word and PDF templates. This parameter does not work with PowerPoint or Excel templates.

Value	Definition
0	Background Mode Preview is not enabled. (Default)
1	Background Mode Preview is enabled and the merged document can be previewed before sending through the background action.
2	Background Mode Preview is enabled and the document download tray is removed from the background preview UI.

**Example:**

&DS7=1

&DS7Preview=1

## DS8

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS8

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Local Template Tab. When enabled, makes the Local Template tab visible, regardless of the corresponding setting on the Setup page.

Value	Definition
0	The Local Template tab behaves according to the setting on the Setup page. (default)
1	The Local Template tab is displayed.

**Example:**

```
&DS8=1
```

## DS9

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DS9

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disable Template Builder link. Enables or disables the Template Builder in Conga Composer.

Value	Definition
0	The Template Builder is enabled. (default)
1	The Template Builder is disabled.

**Example:**

```
&DS9=1
```

## DS10

**Category:** Conga Appearance Parameter

**Compatible with:** Composer 8

**Parameter Name:** DS10

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Disable the Attach File button in the Conga Email Editor. You can find other, more advanced parameters for Conga Window Restrictions in Appearance Parameters.

Value	Definition
0	The Attach File button is enabled. (default)
1	The Attach File button is disabled.

**Example:**

&DS10=1

## DV0

**Category:** Data Gathering Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** DV0

**Parameter Value:** [0 or 999]

**Parameter Requirements:** None

**Parameter Description:** The Data Value parameters insert a value into the Master data set without requiring a custom field in Salesforce.

Also available: DV1, DV2

When used, a column for DV0 appears on the Master dataset in the Template Builder window. This value could be displayed in a watermark or page header on a Word template.

You can also use this parameter to look up a publicly accessible image and use that image in your output document. This will take some testing but is provided for your own exploration.

**Example:**

&DV0=Draft+Document  
&DV0=http://imageurl/

**Merge field in the template:** {{Image:DV0}}

## EC

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EC

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Enable Conga Execution. This parameter is usually set as part of a Salesforce formula to evaluate the contents of a field and, in turn, enable the execution of Conga if the field meets some criteria.

**Example:**

```
&EC={!IF( ISPICKVAL (Opportunity.StageName , "Closed/won"), "1" , "0")}
```

## ECL

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ECL

**Parameter Value:** [Text]

**Parameter Requirements:** &EC=0

**Parameter Description:** Enable Conga Execution Label. Displays a message on the Conga Composer dialog box when Enable Conga Execution is disabled. Displays up to 50 characters.

**Example:**

```
&EC=0
```

```
&ECL=Required+fields+have+not+been+set
```

## EmailAdditionalTo

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailAdditionalTo

**Parameter Value:** [email address]

**Parameter Requirements:** None

**Parameter Description:** EmailAdditionalTo sets the value of the Additional To field.

You can specify multiple addresses by separating each with a comma: `&EmailAdditionalTo=support@getconga.com,billing@getconga.com`. The email address can also be an email address from a Lead, User, or Contact in Salesforce.

You can also use dynamic field content for the parameter value: `&EmailAdditionalTo={!Contact.OtherEmailAddress}`

**Example:**

```
&EmailAdditionalTo=support@getconga.com
```

## EmailBCC

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** EmailBCC

**Parameter Value:** [email address]

**Parameter Requirements:** None

**Parameter Description:** EmailBCC (Email Blind Carbon Copy) sets the value of the BCC field.

You can also use dynamic field content for the parameter value: `&EmailBCC={!Contact.OtherEmailAddress}`. This parameter does not work if Compliance BCC Email is enabled in the Salesforce Setup.

 You can add up to 25 recipients for the **Email BCC** field in each SingleEmailMessage using Salesforce email.

**Example:**

```
&EmailBCC=support@congame.com
```

## EmailCC

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** EmailCC

**Parameter Value:** [email address]

**Parameter Requirements:** None

**Parameter Description:** EmailCC (Email CC Field) sets the value of the CC field.

You can also use dynamic field content for the parameter value. Example: `&EmailCC={!Contact.OtherEmailAddress}`. You can also include more than one email address, such as: `&EmailCC=support@congame.com,{!Contact.OtherEmailAddress}`.

 You can add up to 25 recipients for the **Email CC** field in each SingleEmailMessage using Salesforce email.

**Example:**

```
&EmailCC=support@congame.com
```

## EmailFromId

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7


**Parameter Name:** EmailFromID

**Parameter Value:** [15-character Salesforce ID of an Organization-Wide Address]

**Parameter Requirements:** None

**Parameter Description:** EmailFromID specifies an Organization-Wide Address as the From identity of the email.

The Organization-Wide Address is defined in **Setup** → **Email Administration** → **Organization-Wide Addresses**. After setting up the Org-Wide Address, edit the record to locate the Salesforce Id. It will begin with 0D2. Each Org-Wide Address might be restricted by Profile.

 When using the EmailFromId you must enable the View Setup and Configuration permission settings under the user's System Permissions in Salesforce that gives the end-user visibility, but not the ability to modify the Setup Menu.

This parameter is not for use with EmailReplyToId. They are mutually exclusive.

**Example:**

```
&EmailFromID=0D26100000003H1
```

## EmailLetterheadId

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailLetterheadId

**Parameter Value:** [Salesforce Email Letterhead ID]

**Parameter Requirements:** &CETID or &CongaEmailTemplateId, which specify the Conga Email Template used in the merge process.

**Parameter Description:** EmailLetterheadID specifies which Salesforce email letterhead is used in conjunction with a Conga Email Template.

In Conga Composer Release 7, Conga Email Templates used the Conga Email Staging object which only allowed the use of one Salesforce email letterhead. With Release 8, the Conga Email Staging object has been deprecated and email letterheads are no longer required; however, if you wish to use a Salesforce email letterhead, you can specify it using this parameter.

**Parameter Example:**

```
&CETID=a0WG0000006PNrS
```

```
&EmailLetterheadId=016i000000006FF
```

## Email Parameters

The standard Conga Composer dialog includes the option to generate the desired merge output and attach it directly to an outbound Salesforce email message or Conga Email Template. This feature can greatly reduce the time required to otherwise generate a merge file, save it to disk, launch an email message and attach the file.

 The Email Parameters work with Conga's integration with SendGrid.

Here's an example that would be applicable to a Conga Composer button contained on a Case. It specifies who to address the Email to, establishes the Related object, specifies the Additional To and CC fields, and then specifies an Email Template.

```
https://composer.congamerge.com
```

```
?SessionID={!API.Session_ID}
```

```
&ServerURL={!API.Partner_Server_URL_290}
```



&Id={!Contact.Id}&TemplateId=01H500000005t

&EmailTold={!Contact.Id}

&EmailRelatedTold={!Case.Id}

&EmailAdditionalTo={!Case.OwnerEmail}

&EmailCC=support@mycompany.com

&EmailTemplateId=00X60000000vrZ4

## Parameters

- [CETFilter](#) specifies the where clause to filter the list of email templates that display in the Conga Composer interface.
- [CongaEmailTemplateGroup](#) specifies the group of email templates to display in the Conga Composer interface.
- [CongaEmailTemplateId](#) specifies a Conga Email template.
- [EmailTold](#) specifies the email recipient's ID.
- [EmailTemplateId](#) specifies the template ID for the email.
- [EmailCC](#) specifies the value of the Carbon Copy (CC) field.
- [EmailAdditionalTo](#) specifies an Additional To field value.
- [EmailBCC](#) specifies an Email Blind Carbon Copy (BCC) value.
- [EmailFromId](#) specifies an organization-wide address as the sender of the email.
- [EmailLetterheadId](#) specifies the Salesforce email letterhead.
- [EmailRelatedTold](#) specifies the value of the Related To field.
- [EmailReplyTold](#) specifies the ID of the reply-to address.
- [EmailSubject](#) specifies the value of the Subject field.
- [EmailTemplateAttachments](#) saves attachments stored with the email template.
- [EmailUseSignature](#) includes the sender's user signature in the email.
- [SETI](#) relates email activity history (including enhanced email) to the Contact while using a custom email address if a solution does not specify an EmailTold.

## EmailRelatedTold

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailRelatedTold

**Parameter Value:** [ID of an Account, Opportunity or other Salesforce object]

**Parameter Requirements:** None

**Parameter Description:** EmailRelatedTo sets the value of the Related To field.

The Related To field causes the sent email to appear in the Activity History related list of the object identified. EmailRelatedToId cannot be related back to a who ID (Lead or Contact ID)

**Example:**

```
&EmailRelatedToId={!SalesforceObject.Id}
```

## EmailReplyToId

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailReplyToId

**Parameter Value:** [Salesforce User ID]

**Parameter Requirements:** None

**Parameter Description:** EmailReplyToId changes the apparent sender of an email by setting the message ReplyTo and DisplayName properties to those of the specified user.

This parameter is not for use with EmailFromID. The DisplayName property is not editable when using SendGrid. While the DisplayName property is changed, the email address is the email address of the person that launches the solution.

**Example:**

```
&EmailReplyToId=00536000002MDP3
```

or

```
&EmailReplyToID={!Owner.ID}
```

## EmailSubject

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailSubject

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** EmailSubject sets the value of the Subject field.

This is a Composer button parameter and not to be confused with the Subject line in the actual Composer Email Template. If you set this parameter and use EmailTemplateId, the value you enter here replaces the subject of the email template. Use + to indicate spaces. You can also use dynamic field content for the parameter value:

```
&EmailSubject=Account+Brief+for+{!Account.Name}
```

**Example:**

```
&EmailSubject=Thank+You
```

## EmailTemplateAttachments

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailTemplateAttachments

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &EmailTemplateId

**Parameter Description:** EmailTemplateAttachments indicates to preserve attachments or Salesforce files stored with the Salesforce email template.

Value	Definition
0	Attachments or Salesforce files stored on the Email Template are ignored.
1	Attachments or Salesforce files stored on the Email Template are preserved.

**Example:**

```
&EmailTemplateId=00X43532453dg7A
```

```
&EmailTemplateAttachments=1
```

## EmailTemplateId

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailTemplateId

**Parameter Value:** [Salesforce or Conga email template ID]

**Parameter Requirements:** None

**Parameter Description:** EmailTemplateId loads the subject and body fields from a Salesforce email template.

If the &EmailRelatedToId or &EmailToId parameter is not set, the email is sent to the primary contact. Attachments stored directly in the email template are not applied unless the EmailTemplateAttachments parameter is set to 1.

**Example:**

```
&EmailTemplateId=00X43532453dg7A
```

## EmailToId

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailToId

**Parameter Value:** [Contact or Lead ID]

**Parameter Requirements:** None


**Parameter Description:** EmailToId (Email To Field) sets the value of the To field.

You can also use dynamic field content for the parameter value: &EmailToId={!Contact.Id}

If the Conga Composer Master Object is an Opportunity, the To field is automatically set to the Primary Contact, without the use of this parameter. To override this behavior from an Opportunity, set the value of the parameter to + (&EmailToId=+). Then use the [EmailAdditionalTo](#) parameter to hard code the email you wish to use (for example, &EmailAdditionalTo=support@getconga.com). EmailToId=+ only works when manually updating a button and will not work in the Conga Solutions interface.

If the Conga Composer Master Object is a Quote, the To field is automatically set to the Contact on the Quote, without the use of this parameter.

This parameter works with Leads when using standard Salesforce email templates. This parameter will not work when the master object is a Lead and you are using Conga Email Templates; there is a Salesforce limitation concerning Leads. To use Leads with Conga Email Templates you must use the SendGrid integration. The Lead ID should be the User ID, and the Lead email should be included on the [EmailAdditionalTo](#) parameter.

 You can add up to 100 recipients for the **Email To** field in each SingleEmailMessage using Salesforce email.

**Example:**

```
&EmailToId=003i0000000b4oJi
```

## EmailUseSignature

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** EmailUseSignature

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** EmailUseSignature indicates whether to include or exclude the signature of the sending user from the email body.

This parameter affects only email that is delivered in an unattended manner, such as Background Mode (&DS7=12), Conga Batch, and Conga Trigger.

Value	Definition
0	The signature block is excluded.
1	The signature block is included. (default)

**Example:**

```
&EmailUseSignature=0
```

## ESAccountId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESAccountId

**Parameter Value:** [Account Id]

**Parameter Requirements:** &ESVisible

**Parameter Description:** ESAccountId (Adobe eSign Account ID) specifies the Account to which an Adobe eSign Agreement should be related.

**Example:**

```
&ESVisible=1
```

```
&ESAccountId={!Account.Id}
```

## ESAdditionalTo

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESAdditionalTo

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** ESAdditionalTo (Adobe eSign Additional To) sets the value of the Additional Recipients field.

 This parameter is deprecated and no longer supported. Use ESCopyTo instead.

**Example:**

```
&ESAdditionalTo=John@email.com
```

## ESAgreementName

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESAgreementName

**Parameter Value:** [text]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESAgreementName sets the Agreement Name. If not set, the agreement name is set to the output file name.

**Example:**

```
&ESVisible=1
```

```
&ESAgreementName=Contract+for+{!Account.Name}
```

You can also use dynamic field content for the parameter value:

```
&ESAgreementName={!Opportunity.AgreementName__c}
```

## ESAgreementTemplateId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESAgreementTemplateId


**Parameter Value:** [Salesforce ID of an Adobe eSign Agreement Template]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESAgreementTemplateId (Adobe eSign Agreement Template Id) specifies the Adobe eSign Agreement Template.

Conga Composer applies all of the settings from the specified Adobe eSign Agreement Template to the Adobe eSign Agreement record with the merged output file attached. The Adobe eSign Agreement Template values supersede other Adobe eSign parameters.

An Adobe eSign Agreement Template allows you to specify multiple Adobe eSign Agreement fields (signature type, subject, language, and so on).

 The Auto Send checkbox on the Agreement Template page layout is not compatible with Conga Composer release 8 because the process that auto-sends executes before Composer has a chance to attach the output file from the merge operation. Instead, use `&DS7=14` with `&ESSVBA=1` (the Composer Background Mode that automatically sends the Agreement after the output file has been attached).

Adobe eSign Agreement templates are not compatible with Composer in Salesforce Professional Edition. Composer requires access to the Salesforce API to use Agreement templates; Salesforce Professional Edition orgs do not have access to the Salesforce API.

**Example:**

```
&ESVisible=1
```

```
&ESAgreementTemplateId=a01549h6959dV5R
```

## ESContactId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESContactId


**Parameter Value:** [Salesforce ID of a Contact]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESContactId (Adobe eSign Contact ID) sets the value of the Contact field.

If the Conga Composer Master Object is an Opportunity, the Contact field is automatically set to the Primary Contact, without the use of this parameter.

If the Conga Composer Master Object is a Quote, the Contact field is automatically set to the Quote Contact, without the use of this parameter.

 This parameter may be required depending on the page layout of your Adobe eSign Agreement Object.

**Example:**

```
&ESVisible=1  
&ESContactId=0034100000Am5O6
```

You can also use dynamic field content for the parameter value:

```
&ESContactId={!Contact.Id}.
```

## ESContactId1

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESContactId1

**Parameter Value:** [Salesforce ID of a Contact]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESContactId1 (Adobe eSign Contact ID 1) sets an Additional Contact field.

 Also available: ESContactId2 and ESContactId3.

**Example:**

```
&ESVisible=1  
&ESContactId1=0034100000Am5O6
```

You can also use dynamic field content for the parameter value:

```
&ESContactId1={!Contact.Id}
```



## ESContactId2

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESContactId2

**Parameter Value:** [Salesforce ID of a Contact]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESContactId2 (Adobe eSign Contact ID 2) sets an Additional Contact field.

**Example:**

```
&ESVisible=1
```

```
&ESContactId2=0034100000Am506
```

You can also use dynamic field content for the parameter value:

```
&ESContactId2={!Contact.Id}
```

## ESContactId3

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESContactId3

**Parameter Value:** [Salesforce ID of a Contact]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESContactId3 (Adobe eSign Contact ID 3) sets an Additional Contact field.

**Example:**

```
&ESVisible=1
```

```
&ESContactId3=00143532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESContactId3={!Contact.Id}
```

## ESContractId

**Category:** Adobe eSign Integration Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESContractId

**Parameter Value:** [Salesforce ID of a Contract]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESContractId (Adobe eSign Contract ID) sets the value of the Contract field.

 This parameter may be required depending on the page layout of your Adobe eSign Agreement Object.

**Example:**

```
&ESVisible=1
```

```
&ESContractId=00143532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESContractId={!Contract.Id}
```

## ESCopyTo

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESCopyTo

**Parameter Value:** [email address or a field that contains one or more email addresses]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESCopyTo (Adobe eSign Copy To) sets the value of the Email a Copy To field.

**Example:**

```
&ESVisible=1
```

```
&ESCopyTo=support@appextremes.com
```

You can include multiple email addresses, each separated by a comma:

```
&ESCopyTo=support@appextremes.com, support@xyz.com
```

You can also use dynamic field content for the parameter value:

```
&ESCopyTo={!Contact.OtherEmailAddress}.
```

## ESCUSTOMFIELD

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESCUSTOMFIELD

**Parameter Value:** [Salesforce API name of the field on the Adobe eSign Agreement Object]

**Parameter Requirements:** ESVisible and ESCUSTOMFIELDVALUE

**Parameter Description:** ESCUSTOMFIELD sets the name of a custom field to set on the Adobe eSign Agreement Record created through the merge process.

Use the Salesforce API name of the field on the Adobe eSign Agreement Object.

This parameter is designed to be used for those Adobe eSign Agreement fields that do not have their own corresponding Conga Composer parameter.

You can use a total of 12 Adobe eSign custom fields and values with the format ESCUSTOMFIELD, ESCUSTOMFIELD1, ESCUSTOMFIELD2, ... ESCUSTOMFIELD10, ESCUSTOMFIELD11.

**Example:**

```
&ESVisible=1
```

```
&ESCUSTOMFIELD=Account__c
```

```
&ESCUSTOMFIELDVALUE=001ABCEDFGHIJL
```

## ESCUSTOMFIELDVALUE

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESCUSTOMFIELDVALUE

**Parameter Value:** [value]

**Parameter Requirements:** ESVisible and ESCUSTOMFIELD

**Parameter Description:** ESCustomFieldValue (Adobe eSign Custom Field Value) sets the value of the corresponding custom field.

You can use a total of 12 Adobe eSign custom fields and values with the format ESCustomFieldValue, ESCustomFieldValue1, ESCustomFieldValue2, ..., ESCustomFieldValue10, ESCustomFieldValue11.

**Example:**

```
&ESVisible=1
&ESCustomField=Account__c
&ESCustomFieldValue=001ABCEDFGHIJL
```

## ESDataMappingId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESDataMappingId

**Parameter Value:** [Id]

**Parameter Requirements:** ESVisible and Adobe eSign Data Mapping configured

**Parameter Description:** ESDataMappingId (Adobe eSign Data Mapping Id) specifies the Adobe eSign Data Mapping record.

Adobe eSign form field data mapping lets you define how Adobe eSign will update Salesforce.com objects with data it captures from signers.

**Example:**

```
&ESVisible=1
&ESDataMappingId=a0Ri0000005Jmvd
```

## ESExpireInXDays

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESExpireInXDays

**Parameter Value:** [value]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher.

**Parameter Description:** ESExpireInXDays (Adobe eSign Expire In X Days) sets the number of days before the email expires.

**Example:**

```
&ESVisible=1
```

```
&ESExpireInXDays=14
```

## eSignLive Integration Parameters

### Description

The following parameter enables Conga Composer to interact with eSignLive.

### Parameters

- [DS7](#) background mode is enabled and the merged document is delivered through eSignLive.
- [eslvisible](#) specifies whether eSignLive integrations are visible to the end-user.
- [eSignLiveRIID](#) sets the recipient of the document to be delivered.
- [eSignLiveRIEmail](#) enables the email functionality with Conga.
- [eslSignerIname](#) enables the Quick Add Recipient functionality with Conga.
- [eslPackageName](#) is the minimum required data to create a package in eSignLive.

## eSignLivePackageName

**Category:** eSignLive Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** eslPackageName

**Parameter Value:** [PackageName]

**Parameter Requirements:** None

**Parameter Description:** eSignLivePackageName is the minimum required data to create a package in eSignLive. The visibility of a package is set to `{{ACCOUNT}}` by default. Conga has created this parameter so that the package can be changed to `{{SENDER}}`.

**Example:**

```
&eslPackageName=eSignLive+Package+for+{!Opportunity.Name}+--+{!Today}
```

## eSignLiveR1Email

**Category:** eSignLive Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** eslSigner1email

**Parameter Value:** [Email Address]

**Parameter Requirements:** None

**Parameter Description:** eSignLiveR1Email (eSignLive Quick Add Recipient) enables the Quick Add Recipient functionality with Conga. This allows users to add an email address (and a name) of a recipient who is not a User, Contact, or Lead in their Salesforce org.

Conga supports up to 6 signers. This parameter can be used 6 times by changing the number after R. eSignLiveR2, eSignLiveR3, eSignLiveR4, ... eSignLiveR6.

**Example:**

```
&eslSigner1email=support@conga.com
```

## eSignLiveR1Name

**Category:** eSignLive Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** eslSigner1name

**Parameter Value:** [Firstname+Lastname]

**Parameter Description:** Enables the Quick Add Recipient functionality with Conga. This allows users to add a name (and email address) of a recipient who is not a User, Contact, or Lead in their Salesforce org.

This parameter can be used 6 times by changing the number after R. eSignLiveR2, eSignLiveR3, eSignLiveR4, ... eSignLiveR6.

**Example:**

```
&eslSigner1name=Joe+Smith
```

## eSignLiveVisible

**Category:** eSignLive Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** eslvisible

**Parameter Value:** [Firstname+Lastname]

**Parameter Description:** This parameter determines if eSignLive integration is visible or not. The merge button becomes Merge & eSignLive when &eslvisible=1

You must drag and drop the new button to the appropriate record.

This parameter requires a user to authenticate through eSignLive.

Value	Definition
0	eSignLive visibility is disabled.
1	eSignLive visibility is enabled.

**Example:**

&eslvisible =1

## ESLanguage

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESLanguage

**Parameter Value:** [text]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher

**Parameter Description:** ESLanguage sets the Language field.

See Supported Culture Codes for Adobe Sign for available language values.

**Example:**

&ESLanguage=br-FR

## ESLeadId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESLeadId

**Parameter Value:** [the Salesforce ID of the lead]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher

**Parameter Description** ESLeadId sets the value of the Lead field.

**Example:**

```
&ESVisible=1  
&ESLeadId=00Q43532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESLeadId={!Lead.Id}
```

## ESLeadId1

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESLeadId1

**Parameter Value:** [the Salesforce ID of the lead]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher

**Parameter Description:** ESLeadId1 sets an Additional Lead field.

**Example:**

```
&ESVisible=1  
&ESLeadId1=00Q43532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESLeadId1={!Lead.Id}
```

## ESLeadId2

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESLeadId2

**Parameter Value:** [the Salesforce ID of the lead]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher

**Parameter Description:** ESLeadId2 sets an Additional Lead field.



**Example:**

```
&ESVisible=1
```

```
&ESLeadId=00Q43532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESLeadId2={!Lead.Id}
```

## ESLeadId3

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESLeadId3

**Parameter Value:** [the Salesforce ID of the lead]

**Parameter Requirements:** ESVisible and Adobe eSign version 9.5 or higher

**Parameter Description:** ESLeadId3 sets an Additional Lead field.

**Example:**

```
&ESVisible=1
```

```
&ESLeadId3=00Q43532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESLeadId3={!Lead.Id}
```

## eslSignerIID

**Category:** eSignLive Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** eslSignerIID

**Parameter Value:** [Salesforce Contact, Lead, or User ID]

**Parameter Requirements:** None

**Parameter Description:** eslSignerIID (eSignLive Recipient 1 Id) sets the recipient of the document to be delivered.

This parameter can be used 6 times by changing the number after R. eSignLiveR2, eSignLiveR3, eSignLiveR4, ... eSignLiveR6. Conga supports 1 to 6 signers. For example: `&eslSigner6Id=`

**Example:**

```
&eslSigner1Id={!Contact.Id}
```

## ESMessage

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESMessage

**Parameter Value:** [text]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESMessage sets the value of the Message field.

 Can be populated via a QVar query.

### Example:

```
&ESVisible=1  
&ESMessage=Please+sign+the+attached+document
```

You can also use dynamic field content for the parameter value:

```
&ESMessage={!Opportunity.AgreementMessage__c}
```

## ESOpportunityId

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESOpportunityId

**Parameter Value:** [Salesforce ID of an opportunity]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESOpportunityId sets the value of the Opportunity field.

### Example:

```
&ESVisible=1  
&ESOpportunityId =00643532453dg3C
```

You can also use dynamic field content for the parameter value:

```
&ESOpportunityId = {!Opportunity.Id}
```

## ESPreview

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESPreview

**Parameter Value:** [0 or 1]

**Parameter Requirements:** ESVisible and Adobe eSign 9.5+

**Parameter Description:** ESPreview sets the Preview Field.

**Example:**

```
&ESVisible=1  
&ESPreview=1
```

## ESRecipient1

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient1

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient1 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient1=00Q43532453dg3C
```

## ESRecipient1Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient1Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient1

**Parameter Description:** ESRecipient1Roles specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient1=0031500001bxwff  
&ESRecipient1Role=Approver
```

## ESRecipient2

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient2

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient2 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient2=00Q43532453dg3C
```

## ESRecipient2Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient2Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient2

**Parameter Description:** ESRecipient2Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient2=00Q43532453dg3C  
&ESRecipient2Role=Signer
```

## ESRecipient3

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient3

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient3 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient3=00Q43532453dg3C
```

## ESRecipient3Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient3Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient3

**Parameter Description:** ESRecipient3Roles specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient3=00Q43532453dg3C  
&ESRecipient3Role=Signer
```

## ESRecipient4

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient4

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient4 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient4=00Q43532453dg3C
```

## ESRecipient4Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient4Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient4

**Parameter Description:** ESRecipient4Roles specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient4=00Q43532453dg3C  
&ESRecipient4Role=Signer
```

## ESRecipient5

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient5

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient5 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient5=00Q43532453dg3C
```

## ESRecipient5Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient5Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient5

**Parameter Description:** ESRecipient5Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient5=00Q43532453dg3C  
&ESRecipient5Role=Signer
```

## ESRecipient6

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient6

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient6 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient6=00Q43532453dg3C
```

## ESRecipient6Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient6Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient6

**Parameter Description:** ESRecipient6Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**



```
&ESVisible=1  
&ESRecipient6=00Q43532453dg3C  
&ESRecipient6Role=Signer
```

## ESRecipient7

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient7

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient7 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient7=00Q43532453dg3C
```

## ESRecipient7Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient7Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient7

**Parameter Description:** ESRecipient7Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient7=00Q43532453dg3C  
&ESRecipient7Role=Approver
```

## ESRecipient8

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient8

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient8 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient8=00Q43532453dg3C
```

## ESRecipient8Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient8Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient8

**Parameter Description:** ESRecipient8Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient8=00Q43532453dg3C  
&ESRecipient8Role=Approver
```

## ESRecipient9

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient9

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient9 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient9=00Q43532453dg3C
```

## ESRecipient9Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient9Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient9

**Parameter Description:** ESRecipient9Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient9=00Q43532453dg3C  
&ESRecipient9Role=Signer
```

## ESRecipient10

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient10

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient10 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient10=00Q43532453dg3C
```

## ESRecipient10Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient10Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient10

**Parameter Description:** ESRecipient10Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient10=00Q43532453dg3C  
&ESRecipient10Role=Approver
```

## ESRecipient11

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRecipient11

**Parameter Value:** [ID or email address]

**Parameter Requirements:** Adobe eSign version 11.0+

**Parameter Description:** ESRecipient11 allows you to specify the recipients of an Adobe eSign Agreement using the Adobe eSign Recipients Related List object.

The benefit of using this object rather than the discrete Lookup fields is that you may specify the ID of a Contact, User, Lead, Person Account or simply an email address as a Recipient.

You can specify up to 12 Adobe eSign recipients by incrementing the number in this parameter.

**Example:**

```
&ESRecipient11=bob@bob.com
```

## ESRecipient11Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient11Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient11

**Parameter Description:** ESRecipient11Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient11=bob@bob.com  
&ESRecipient11Role=Signer
```

## ESRecipient12Role

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESRecipient12Role

**Parameter Value:** [Signer, Approver, etc.]

**Parameter Requirements:** ESVisible and ESRecipient12

**Parameter Description:** ESRecipient12Role specifies the role of the Adobe eSign recipient. You can specify up to 12 roles by incrementing the number in this parameter.

**Example:**

```
&ESVisible=1  
&ESRecipient2=00Q43532453dg3C  
&ESRecipient2Role=Approver
```

## ESRemindRecipient

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRemindRecipient

**Parameter Value:** [0, 1, 2 or 3]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESRemindRecipient sets the Remind Recipient to Sign field.

Value	Definition
0	Remind Recipient is not set.
1	Remind Recipient is set to Never.
2	Remind Recipient is set to Every Day, Until Signed.
3	Remind Recipient is set to Every Week, Until Signed.

**Example:**

&ESVisible=1  
&ESRemindRecipient=2

## ESRequirePWSign

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRequirePWSign

**Parameter Value:** [0 or 1]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESRequirePWSign sets the Password Required to Sign field.

Value	Definition
0	Password Required to Sign is disabled. (default)
1	Password Required to Sign is enabled.

**Example:**

&ESVisible=1  
&ESRequirePWSign=1

## ESRequirePWView

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRequirePWView

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** ESRequirePWView sets the Password Required to View field.

Value	Definition
0	Password Required to View is disabled. (default)
1	Password Required to View is enabled.

**Example:**

&ESRequirePWView=1

## ESRequireSelfSign

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESRequireSelfSign

**Parameter Value:** [0 or 1]

**Parameter Requirements:** ESVisible and ESSignatureOrder

**Parameter Description:** ESRequireSelfSign sets the I Also Need To Sign field.

Value	Definition
0	I Also Need To Sign is disabled. (default)
1	I Also Need To Sign is enabled.

**Example:**

&ESVisible=1

&ESSignatureOrder=1

&ESRequireSelfSign=1

## ESSignatureGroup1

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** ESSignatureGroup1

**Parameter Value:** [Signer Group ID]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESSignatureGroup1 specifies the signature group in Adobe eSign.

You can specify up to 12 roles by incrementing the number in this parameter.

**Note:** Requires Adobe Sign v18.1 or greater

**Example:**



&ESVisible=1  
 &ESSignatureGroup1=00G000000pOU8yT

## ESSignatureOrder

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESSignatureOrder

**Parameter Value:** [1 or 2]

**Parameter Requirements:** ESVisible and ESRequireSelfSign

**Parameter Description:** ESSignatureOrder sets the Signature Order field.

Value	Definition
1	Signature Order is set to Recipient signs, then I sign.
2	Signature Order is set to I sign, then the Recipient signs.

**Example:**

&ESVisible=1  
 &ESSignatureOrder=1  
 &ESRequireSelfSign=1

## ESSignatureType

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESSignatureType

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** ESVisible

**Parameter Description:** ESSignatureTypes sets the Signature Type.

Value	Definition
0	Signature Type is not set.
1	Signature Type is set to Written Signature.

Value	Definition
2	Signature Type is set to e-Signature.

**Example:**

```
&ESVisible=1
&ESSignatureType=2
```

## ESSVBA

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESSVBA

**Parameter Value:** [0 or 1]

**Parameter Requirements:** ESVisible. In addition, API access must be enabled for Adobe eSign and the Adobe eSign API Key entered in the Adobe eSign configuration settings.

**Parameter Description:** ESSVBA sends an Adobe eSign Agreement via background action (enhance automation delivery). When automatically sending an Adobe eSign Agreement through the API, you may elect to trigger the built-in delivery mechanism of Adobe eSign.

**Note:** This parameter works with the &DS7=14 and &DS7Preview=1 parameters but not with the &DS7=24 parameter.

Value	Definition
0	Adobe eSignBackground Action is disabled. (default)
1	Sends an Adobe eSign agreement through the Adobe eSignAPI.

**Example:**

```
&ESVisible=1
&ESSVBA=1
```

## ESVisible

**Category:** Adobe eSign Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ESVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** ESVisible changes the Action picklist in the Conga Composer dialog box to Adobe eSign and updates the Merge button to Merge and Adobe eSign.

Adobe eSign is available separately on the AppExchange. You must subscribe to that service to use this integration parameter.

**Important:** This parameter is required for all other Adobe eSign parameters.

Value	Definition
0	Adobe eSign is not the designated Action. (default)
1	Action is set to Adobe eSign.

**Example:**

&ESVisible=1

## ExcelAutoFitColumns

**Category:** Excel Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ExcelAutoFitColumns

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Automatically sizes the width of all columns in an Excel output document.

Value	Definition
0	The width of the columns is not adjusted. (default)
1	The width of the columns is automatically adjusted.

**Example:**

&ExcelAutoFitColumns=1

# ExcelAutoFitRows

**Category:** Excel Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ExcelAutoFitRows

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Automatically sizes the height of all rows in an Excel output document.

Unfortunately, in Conga Composer Release 8, this parameter does not work well on rows in which the cells have been merged. We recommend not merging cells in your Excel template if you plan to use this parameter.

Value	Definition
0	The height of the rows is not adjusted. (default)
1	The height of the rows is automatically adjusted.

**Example:**

&ExcelAutoFitRows=1

## Excel Parameters

### Description

The following parameters affect the behavior of Excel templates.

### Parameters

- [ExcelAutoFitColumns](#) sets the Excel columns to auto-fit to content.
- [ExcelAutoFitRows](#) automatically sets the height of all Excel rows.
- [ExcelStripFormulas](#) strips formulas from the Excel output file.
- [ExcelPW](#) specifies a password value needed to open the merged Excel document.
- [SCHP](#) strips punctuation from Merge Field names.

## ExcelPW

**Category:** Excel Parameter

**Compatible with:** Composer 8


**Parameter Name:** ExcelPW

**Parameter Value:** [password]

**Parameter Requirements:** None

**Parameter Description:** With this parameter enabled each Excel output document will require the specified password value in order to be opened.

This parameter only works for Excel to Excel merges.

 Do not password protect the actual template.

**Example:**

&ExcelPW=hugeC@nary16

## ExcelStripFormulas

**Category:** Excel Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ExcelStripFormulas

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Strip formulas from Excel output file. When enabled, the output Excel workbook contains only the results of the formulas in the workbook, not the formulas themselves.

Value	Definition
0	Formulas remain in the Excel output file. (default)
1	Formulas are removed from the Excel output file.

**Example:**

&ExcelStripFormulas=1

## FlattenPDFForms

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** FlattenPDFForms

**Parameter Value:** [0 or 1]

**Parameter Requirements:** A PDF Form Template. (Not for use with Word, Excel or PowerPoint templates.)

**Parameter Description:** Flatten Form fields on the output PDF. This parameter addresses a limitation in the iPhone/iPad PDF Reader which does not display values in PDF form fields. By flattening, the values of the fields are blended into the PDF background and are thus visible on iPads.

Because this parameter flattens field contents, it is not compatible with form fields that contain code such as javascript.

Value	Definition
0	PDF form fields are not flattened. (default)
1	PDF form fields are flattened.

**Example:**

```
&FlattenPDFForms=1
```

## FPO

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** FPO

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** None

**Parameter Description:** FPO locks the output file format.

On computers and tablets using Apple's operating system, PDF files may appear empty when opened using Preview. To avoid this, open PDF files on Apple computers using Adobe Reader.

Value	Definition
0	Output Format is Same as template and the output format radio buttons are not locked. (default)
1	Output Format is Adobe Acrobat and the output format radio buttons are locked.
2	Output Format is Same as template and the output format radio buttons are locked.

**Example:**

&FP0=1

## Google Drive Parameters

### Description

Conga Composer offers integration with Google Drive, allowing you to generate and save files to Google Drive as Microsoft Office or Google documents.

### Parameters

- [DS7](#) enables merge operations to occur in the background.
- [GoogleVisible](#) enables Google as an available output option.
- [OFP](#) specifies the file path within Google Drive where files are saved.
- [QMode](#) for use with Conga Batch or Conga Trigger solutions.
- [SCI](#) defines where a copy of the merged output file is stored.

## GoogleVisible

**Category:** Google Drive Parameter

**Compatible with:** Composer 8

**Parameter Name:** GoogleVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Google Visible enables Google as an output option on the Conga Composer user interface.

Value	Definition
0	Hides Google as an available output option.
1	Enables Google as an output option by creating a Google Document.

**Example:**

```
&GoogleVisible=0
```

## HideContent

**Category:** Template Selection Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** HideContent

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &SelectTemplates=1

**Parameter Description:** HideContent suppresses Salesforce Content and Documents from being displayed in the Select Templates view of the Composer dialog box.

Value	Definition
0	Include Salesforce Content & Documents checkbox is visible. (default)
1	Include Salesforce Content & Documents checkbox is not displayed.

**Parameter Example:**

```
&SelectTemplates=1
```

```
&HideContent=1
```

## Id

**Category:** Data Gathering Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** Id

**Parameter Value:** [Salesforce ID]



**Parameter Requirements:** None

**Parameter Description:** Id (Master Object Id) sets the master object. All fields are retrieved from this object.

You can also use dynamic field content for the parameter value: `&Id={!Account.Id}`

This parameter is required in order for Conga Composer to function. A Conga Composer URL must contain one (and only one) instance of this parameter.

**Example:**

`&Id={!Account.Id}`

## LG0

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LG0

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** LG0 (Save Text) determines whether to store the text of the document (post-merge) to the body of the Task.

This feature works only with Microsoft Word templates.

Value	Definition
0	Does not save text. (Default)
1	Saves text.

**Parameter Example:** `&LG0=1`

## LG1

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LG1

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** LG1 (Activity Subject) sets the Subject to the text specified. Also used as in the subject of a Follow-Up Task, if one is created.

If not specified and the Activity Subject is blank, the output filename is used for the Activity Subject. Do not use the less than (<) or greater than (>) symbol with this parameter.

**Parameter Example:** `&LG1=Quote+Generated`

## LG2

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LG2

**Parameter Value:** [integer]

**Parameter Requirements:** None

**Parameter Description:** Sets the follow-up date by using the integer to add days to today's date.

**Example:**

`&LG2=14`

## LG3

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LG3

**Parameter Value:** [0, 1, or 2]

**Parameter Requirements:** None

**Parameter Description:** Sets the file attachment options.

Value	Definition
0	Attach File: is -None-. (default)

Value	Definition
1	Attach File: is Attach to[record] where [record] would be the type of master object record. For example, Opportunity.
2	Attach File: is Attach only, No Task Record.

**Example:**

&LG3=2

## LG4

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LG4

**Parameter Value:** [0,1, or 2]

**Parameter Requirements:** None

**Parameter Description:** Specifies whether to automatically log an activity after merging.

Value	Definition
0	Automatic Logging is disabled. (default) Disabling Automatic Logging also disables field updates ( <a href="#">MFTS</a> ).
1	Automatic Logging is enabled.
2	Master Field to Set fields are updated but no task record is created.

Activity Logging for Mail Merge only works for Master Object IDs. You cannot update Child records in a Mail Merge solution.

**Example:**

&LG4=1

## LG5

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** LG5

**Parameter Value:** [ID]

**Parameter Requirements:** None

**Parameter Description:** A Salesforce Id of a User or a field that contains this kind of Id. In the absence of this parameter, the Assigned To field is set to the current user.

**Example:**

`&LG5={!Opportunity.Assistant__c}`

## LG6

Sets the Assigned To field in the Follow-Up Activity

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** LG6

**Parameter Value:** [Salesforce ID]

**Parameter Requirements:**

**Parameter Description:** Sets the follow-up date by using the integer to add days to today's date.

You can also use dynamic field content for the parameter value: `&LG6={!Opportunity.Assistant__c}`

In the absence of this parameter, the Assigned To field of a follow-up activity is set to the current user.

**Example:**

`&LG6=00553532453dg3D`

## LG7

Sets the Activity Subject in the Follow-Up Activity

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** LG7

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** Sets the Subject to the text specified. If not specified the Follow-Up Activity Subject is set to Follow-Up: and the Activity Subject value.

You can also use dynamic field content for the parameter value: *&LG7={!Opportunity.EmailSubject}*

Do not use the less than (<) or greater than (>) symbol with this parameter.

**Example:**

&LG7=Please+follow-up

## LGAttachOption

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7

**Parameter Name:** LGAttachOption

**Parameter Value:** [0, 1, 2, 3, or 4]

**Parameter Requirements:** &LG2=14

Designates the location of the output file

**Parameter Description:** Sets the Salesforce object to which attachments generated during Activity Logging are stored.

Works with &AttachmentParentId.

Value	Definition
0	Attachments are saved to Notes & Attachments. (Default)
1	Attachments are saved to Salesforce Content.
2	Attachments are saved to Salesforce Chatter.
3	Attachments are saved to SpringCM.
4	Attachments are saved to Google Drive.

## LGParentID

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** LGParentId

**Parameter Value:** [QVar0 or QVar1 or QVar2]

### Important

This parameter must be populated via a QVAR query. You must create a query to locate the Parent ID. For more information on QVars (Querystring Variable), refer to the [Using Qvars in a Conga Composer Solution](#) section. Refer to the following example for your reference.

**Example:**

**SOQL Select Statement (Conga Query):** *SELECT Opportunity.Id FROM Quote WHERE Id="{pv0}"*  
*&QVar0ID=a03f4000000vnCL* (15 digit Salesforce ID of a query record created above.)

**Parameter Requirements:** &Id={!Opportunity.Id}

**Parameter Description:** Override the default record against which a completed task is created. In the absence of this parameter, the task is created against the Master Object.

You should use [&LG4=1](#) or [&AC0=1](#) in conjunction with *LGParentID* for maximum benefits.

**Example:**

*&QVar0ID=a03f4000000vnCL*

*&LGParentId={QVar0}*

## LGWhoID

**Category:** Activity Logging Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** LGWhoId

**Parameter Value:** [Salesforce ID of an object that allows Activity Tracking]

**Parameter Requirements:** None

**Parameter Description:** Specifies the WhoID (the related Contact/Lead) when creating a Task record.

**Example:**

&LGWhoId=0033000000GvHjL

## LinkService

**Category:** Link Service

**Compatible with:** Composer 8

**Parameter Name:** LinkService

**Parameter Value:** [0 or 1]

**Parameter Requirements:** The org must have the Conga File Storage feature enabled in Conga Composer Setup to utilize the Composer Link Service.

**Parameter Description:** LinkService enables Composer Link Service and allows Composer to generate a document link.

### Important

To generate the document link in an email, add the {Conga\_Doc\_Link} merge tag in the body of a Conga Email Template (defined by the [CongaEmailTemplateId](#) parameter).

Use the [NETId](#) parameter to send email notifications with the Composer Link Service when recipients interact with the online document.

Value	Definition
0	Disables Composer Link Service
1	Enables Composer Link Service

**Example:**

&LinkService=1

## LiveEditEnable

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** LiveEditEnable

**Parameter Value:** [0 or 1]

**Parameter Requirements:** The [&LiveEditVisible=1](#) parameter

**Parameter Description:** The LiveEditEnable parameter checks the Live Edit checkbox in the Composer user interface by default.

Value	Definition
0	{{LiveEdit}} check box not selected.
1	{{LiveEdit}} check box is selected.

**Example:**

&LiveEditVisible=1

## LiveEditVisible

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** LiveEditVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** The LiveEditVisible parameter makes the Live Edit check box available in the Conga Composer user interface.

Value	Definition
0	Live Edit check box is not available.
1	Live Edit check box is available.

**Example:**

&LiveEditVisible=1



## Locale and Culture Parameters

Conga Composer pulls the user's Locale field from the User object. This value then governs the user's interaction with Composer, which affects Numeric and Date Time field formatting, as well as picklist values for master object field values. For Conga Queries, the person writing the SOQL is responsible for adding the ToLabel() function around picklist field names.

You can override the user's culture with `&Culture` (for numbers AND dates) or `&CurrencyCulture` (for currency numbers only).

Primarily used for international scenarios, these parameters affect the appearance of number and date values in Microsoft Word templates.

These options override default data formatting options and are generally not necessary for most users.

### Parameters

- `Culture` specifies the cultural setting for numbers and dates.
- `CurrencyCulture` specifies the cultural settings for numbers.
- `TPV` specifies that picklist values are translated.

## LRTF

**Category:** Word Parameter

**Compatible with:** Composer 8

**Parameter Name:** LRTF

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** The LRTF (Lightning Rich Text Field) parameter allows rich text fields to be merged in the Lightning Experience without adding additional line breaks in resulting lists.

Value	Definition
0	Lightning Rich Text Field is disabled. (default)
1	Lightning Rich Text Field is enabled.

**Example:**

&LRTF=1

## MFTS0

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTS0

**Parameter Value:** [APIFieldName]

**Parameter Requirements:** MFTSValue0 and UF0

**Parameter Description:** MFTS0 (Master Field to Set) determines the field that is updated when the Composer solution runs.

The UF0 parameter is the pre-requisite that turns on the Composer Update Fields feature. The MFTS0 parameter identifies the field to update. This should be the API Name of the field. The MFTSValue0 parameter defines the value you want to update the field with.

You can increment the MFTS and MFTSValue parameters to update additional fields. See [MFTS1](#) and [MFTSValue](#). If you have a CHECKBOX on your object that you want to be enabled when you merge, set your MFTSValue to TRUE.

 Master Field to Set does not work with Global Merge Solutions.

In the example below, the Member\_Status\_\_c field is updated with a value of Patron every time the Composer solution runs.

**Example:**

&UF0=1 or &MFTS0=Member\_Status\_\_c or &MFTSValue0=Patron

## MFTS1

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTS1

**Parameter Value:** [ApiFieldName]

**Parameter Requirements:** UF0 and MFTSValue1

**Parameter Description:** MFTS1 (Master Field to Set) determines the field that is updated when the Composer solution runs.

The UF0 parameter is the pre-requisite that turns on the Composer Update Fields feature. The MFTS1 parameter identifies the field to update. This should be the API Name of the field. The MFTSValue1 parameter defines the value you want to update the field with.

You can increment the MFTS and MFTSValue parameters to update additional fields. See MFTS2 and MFTSValue2.

In the example below, the OrderNumber\_\_c field will be updated with a value of 12 every time this Composer solution runs.

**Example:**

```
&UF0=1 or &MFTS1=OrderNumber__c or &MFTSValue1=12
```

## MFTS2

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTS2

**Parameter Value:** [ApiFieldName]

**Parameter Requirements:** MFTSValue2 and UF0

**Parameter Description:** MFTS2 (Master Field to Set) determines the field that is updated when the Composer solution runs.

The UF0 parameter is the pre-requisite that turns on the Composer Update Fields feature. The MFTS2 parameter identifies the field to update. This should be the API Name of the field. The MFTSValue2 parameter defines the value you want to update the field with.

You can increment the MFTS and MFTSValue parameters to update additional fields. See [MFTS0](#) and [MFTSValue0](#).

In the example below, the TrackingNumber\_\_c field is updated with a value of XYZ200ABCD every time this Composer solution runs.

**Example:**

```
&UF0=1 or &MFTS2=TrackingNumber__c or &MFTSValue2=XYZ200ABCD
```

## MFTSId0

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSId0

**Parameter Value:** [text]

**Parameter Requirements:** UF0, MFTS0 and MFTSValue0

**Parameter Description:** MFTSId (Master Field to Set ID) sets a field on a non-Master record (a record other than the Master Object.)

 This parameter may be populated via a QVar query.

**Example:**

```
&UF0=1
&MFTS0=Opportunity_Stage__c
&MFTSValue0=New
&MFTSId0={!Opportunity.Id}
```

## MFTSId1

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSId1

**Parameter Value:** [text]

**Parameter Requirements:** UF0, MFTS1 and MFTSValue1

**Parameter Description:** MFTSId1 (Master Field to Set) sets a field on a non-Master record (a record other than the Master Object.)

 This parameter may be populated via a QVar query.

**Example:**

```
&UF0=1
&MFTS1=Opportunity_Stage__c
&MFTSValue1=New
&MFTSId1={!Opportunity.Id}
```

## MFTSId2

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSid2

**Parameter Value:** [text]

**Parameter Requirements:** UF0, MFTS2 and MFTSValue2

**Parameter Description:** MFTSid2 (Master Field to Set Id 2) sets a field on a non-Master record (a record other than the Master Object).

 This parameter may be populated via a QVar query.

**Example:**

```
&UF0=1
&MFTS2=Thank_You_Letter
&MFTSValue2=Sent
&MFTSid2={!Contact.Id}
```

## MFTSValue0

**Category:** Activity Logging Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSValue0

**Parameter Value:** [text]

**Parameter Requirements:** UF0 and MFTS0

**Parameter Description:** MFTSValue0 (Master Field to Set Value) determines the value of the master field to be set.

 Do not use with &, %, < or > or other special characters.


Use TODAY to set the Master Field to Set Value to today's date.

Examples: `&MFTSValue0={!TEXT(TODAY())}` or `&MFTSValue0=TODAY` or `&MFTSValue0={!Today}`

You can add days using this syntax: `&MFTSValue0={!TEXT(TODAY()+7)}`

If run in a formula (using Conga Batch or Conga Trigger), TODAY needs to be treated as Static text: `"&MFTSValue0="+ {!TEXT( TODAY() )} +`

If using a date/time field, use this syntax: `&MFTSValue0=Now`

 Do not use parentheses around Master Field values. For example, do not use TODAY() or Now().

When referencing a standard checkbox field with the MFTS parameter, use True as the MFTSValue: &MFTS0=HasProducts\_\_c &MFTSValue0=True

**Example:**

&UF0=1 or &MFTS0=Member\_Status\_\_c or &MFTSValue0=Patron

## MFTSValue1

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSValue1

**Parameter Value:** [text]

**Parameter Requirements:** UF0 and MFTS1

**Parameter Description:** MFTSValue1 (Master Field to Set Value 1) determines the value of the master field to be set.

Note: Do not use with &, %, < or > or other special characters.

If you are using a date/time field you should use this syntax: &MFTSValue0=Now

You can use TODAY to set the Master Field to Set Value to today's date:

Examples: &MFTSValue0={!  
TEXT(TODAY())} or &MFTSValue0=TODAY or &MFTSValue0=TODAY or &MFTSValue0={!Today}

If run in a formula (using Conga Batch or Conga Trigger), TODAY needs to be treated as Static text.

**Example:**

&UF0=1 or &MFTS1=Opportunity\_Stage\_\_c or &MFTSValue1=New

## MFTSValue2

**Category:** Activity Logging Parameter


**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** MFTSValue2

**Parameter Value:** [text]

**Parameter Requirements:** UF0 and MFTS2

**Parameter Description:** MFTSValue2 (Master Field to Set Value 2) determines the value of the master field to be set.

 Do not use with &, %, < or > or other special characters.

**Example:**

```
&UF0=1
&MFTS2=TrackingNumber__c
&MFTSValue2=Sent
```

## MTOFN

**Category:** Output Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** MTOFN

**Parameter Value:** [text]

**Parameter Requirements:** None

**Parameter Description:** Multi-template Output File Name sets the output file name when using multiple templates or running batches, affecting the file name of .zip files or consolidated PDFs.

**Example:**

```
&MTOFN={!Account.Name}+--+Invoice
```

## NETId

**Category:** Link Service

**Compatible with:** Composer 8

**Parameter Name:** NETId

**Parameter Value:** [ID]

**Parameter Requirements:** &LinkService=1

 **Important**

The org must have Conga File Storage and Enable Email Notifications features enabled in Conga Composer Setup to utilize the Composer Link Service.

**Parameter Description:** The NETId (Notification Email Template Id) pre-selects a Conga Email Template to send as the Notification Email when a document is generated and sent with the Composer Link Service.

The parameter value is a Conga Email Template Id. Specify only one Conga Email Template Id as the NETId parameter value. The Conga Email Template body must have content or the Notification Email will not send.

⚠ Exclude this parameter to bypass sending an email notification for document interactions with Composer Link Service.

**Example:**

```
&LinkService=1  
&NETId=a03f400000oHbPE
```

## OCNR

**Category:** Advanced Control Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** OCNR

**Parameter Value:** [0,1, or 2]


**Parameter Requirements:**

**Parameter Description:** Disables the default behavior when the Close button is clicked. By default, the Salesforce record is refreshed when Close is clicked.

If your Composer Solution involves sending an email, the OCNR parameter can cause the solution to send the following error: "There was an issue sending the Composer request." To avoid this error, ensure background mode (the DS7=12 parameter) is enabled. If you are using &OCNR=0, close Composer by clicking Return to record.

Value	Definition
0	Salesforce record is refreshed. (default)
1	Salesforce record is not refreshed.



Value	Definition
2	Prevents Conga Composer from reloading the window when a merge completes.  <div style="border: 1px solid red; padding: 5px; margin-top: 10px;">  This parameter value is only for Salesforce environments with Lightning Console enabled and is only applicable when using the DS7 background mode parameter with all values except 3. </div>

**Example:**

&OCNR=1

## OFN

**Category:** Output Parameter


**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** OFN

**Parameter Value:** [text and/or merge field]

**Parameter Requirements:** None

**Parameter Description:** OFN (Output File Name) sets the Output File Name, overriding the default name of the file.

 You may use static text or merge field names for the values to place in the filename. Use a plus sign (+) to represent literal spaces.

 Do not include symbols unsupported in filenames, ;, #, ", \, /, >, <, etc. Character length is from 2 to 80 characters.

The following special fields can also be used as the value of the OFN parameter:

- {Template.Label}


This special field refers to the template label or template name (as defined in the Conga Template record or the Salesforce template repository). If using a local template, the filename appears in place of the label.


- {Template.Id}

This special field refers to the 15 character Salesforce ID of a Conga Template or Salesforce template repository record. It does not apply to local templates.

**Multiple Templates** - When merging multiple templates, you may optionally designate individual output file names for each template by including the output file name in brackets before the template Id specified:

```
&TemplateId=[Invoice+with+Discount]00143532453dg3B,
[!Account.Name]+Terms+and+Conditions]001435121234499
```

 You may not use {Template.Label} nor {Template.Id} within the brackets before the template Id is specified. An error will occur if attempted: "The following template Id/s do not exist or are not accessible and will be omitted:"

 Internet Explorer 7 will replace spaces in a file name with the web--safe characters %20 if the output file is opened before saving to disk. For example, the file in the usage example above would appear as ABC%20Corp.%20-%20Quote.doc if the file is opened before saving to disk.

**Lightning Component:** The OFN parameter needs to leverage a formula field within the OFN parameter for the Lightning Component. Refer to the following steps as an example:

1. Create a formula field on the Salesforce object. (For example, OFN Label)
2. Insert desired Output File Name (text and/or merge field) into the formula field.
3. Set the OFN parameter to OFN={!APXTConga4\_ *formula field name*} (For example, OFN={!APXTConga4\_OFN\_Lable\_\_c})

**Example:**

```
&OFN={!Account.Name}+-+Quote
```

## OFP

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** OFP

**Parameter Value:** [path]

**Parameter Requirements:** &GoogleVisible=1 or &SC1=Google or &BoxVisible=1 or &SC1=Box or &BigTinCanVisible=1 or &SC1=BigTinCan or &OneDriveVisible=1 or &eslvisible=1

**Parameter Description:** OFP (Output File Path) specifies the file path within the file storage system where the merged output files will be saved. Each folder name should be separated by a backslash or forward slash. We recommend using forward slashes because

backslashes can cause issues in Salesforce Lightning and Salesforce formulas (see tip below).

- Box Integration
  - For the Box integration, the file path must be `&OFP=RootFolder/Folder+Name/File+Name`. Also, avoid using spaces in the Salesforce Opportunity name.
  - The Box OFP parameter is case-sensitive; the value must exactly match the folder definition in Box.
  - Single backslashes work as expected in Conga Composer Button URLs. When used in formulas, single backslashes cause the formula to fail. When converting button URLs to formulas, use a double backslash (`\\`). In formulas, the backslash is an escape character that specifies that the character that follows should be taken literally.
- BigTinCan Integration
  - For the Bigtincan integrations, the file path must be `&OFP=Channel/Story+Name`.
  - There is no need to reference the tab.
  - The OFP parameter must be two segments only. The two segments must represent the channel/story. The channel must be created through the bigtincan hub. The story can be dynamically created.

#### Parameter Example:

```
&GoogleVisible=1
```

```
&OFP=Corporate+Docs/{!Opportunity.Name}
```

## OneDrive Parameters

### Description

Conga Composer offers integration with OneDrive, so that you are no longer limited to storing your files within Salesforce.com.

### Parameters

- [DS7](#) enables merge operations to occur in the background.
- [OneDriveVisible](#) enables OneDrive as an available output option.
- [OFP](#) specifies the file path within OneDrive where files are saved.
- [QMode](#) for use with Conga Batch or Conga Trigger solutions.
- [SCI](#) defines where a copy of the merged output file will be stored.

## OneDriveVisible

**Category:** Integration Parameters

**Compatible with:** Composer 8

**Parameter Name:** OneDriveVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** This parameter determines if OneDrive integration is visible or not. This parameter will require a user to authenticate through OneDrive.

Value	Definition
0	Hides OneDrive as an available output option.
1	Enables OneDrive as an output option.

**Example:**

&OneDriveVisible=1

## Output Parameters

### Description

The Output File Parameters Group lists the parameters that primarily affect the behavior of the merge templates used in a merge operation.

### Parameters

- [AmazonS3Visible](#) enables Amazon S3 as an output option on the Conga Composer user interface,
- [APDF](#) specifies whether multiple PDF documents are consolidated into one.
- [APF](#) specifies whether PowerPoint output files are assembled.
- [AttachmentID](#) specifies that the attachments associated with the Salesforce IDs will be automatically selected.
- [AttachmentParentId](#) specifies the Salesforce record where the output file is attached.
- [AttachmentSourceID](#) specifies the Salesforce ID of an object that allows attachments.
- [AWD](#) specifies whether Word documents are assembled (and if so, how).
- [DefaultPDF](#) sets the default output file type.
- [DisableRowNums](#) disables row numbering for detail datasets.
- [FPO](#) locks the output file format.
- [LiveEditEnable](#) checks the {{LiveEdit}} checkbox by default.

- [LiveEditVisible](#) makes the {{LiveEdit}} check box available in the Conga Composer user interface.
- [MTOFN](#) specifies a file name when using multiple templates or batches.
- [OFN](#) specifies the output file name.
- [PDFA](#) converts the output file to PDF/A1a or PDF/A1b file types
- [PDFVersion](#) specifies the PDF version in which the generated document should be saved.
- [Select Attachments](#) specifies that the Salesforce ID of up to 20 attachments of the master object will be included in an Attachments tab for an end-user to select during the merge process.
- [ZipFiles](#) specifies whether output files are grouped.
- [ZipPDFs](#) specifies that PDF documents generated in the merge process should be zipped (compressed).

## PDFA

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** PDFA

**Parameter Value:** [1a or 1b]

**Parameter Requirements:** [TemplateID](#) parameter specifying a Word Docx template and either the [DefaultPDF](#) parameter or [APDF](#) parameter.

**Parameter Description:** The PDFA parameter converts Microsoft Word Docx templates into PDF/1a or PDF/1b files. You must use either the DefaultPDF or APDF parameters in the solution along with the PDFA parameter.

 This parameter only supports Microsoft Word Docx templates. PDF templates are not compatible with the PDFA parameter

Parameter Value	Definition
1a	Converts the output file to a PDF/1a file type
1b	Converts the output file to a PDF/1b file type

**Example:**

```
&TemplateId=a01268Hc74Vb9Mh &DefaultPDF=1 &PDFA=1a
```

## PDFEF

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 8

**Parameter Name:** PDFEF

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** PDFEF embeds full font sets in the PDF output file.

Value	Definition
0	Composer does not attempt to embed full fonts. (default)
1	Composer attempts to embed full fonts in the PDF output file.

**Example:**

&PDFEF=1

## PDFIQ

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** PDFIQ

**Parameter Value:** [5 to 100]

**Parameter Requirements:** None

**Parameter Description:** PDFIQ (PDF Image Quality) adjusts the quality of images in output PDF files.

The quality scale ranges from 5 (low quality, small file size) to 100 (high quality, large file size). (Default value is 100.)

**Example:**

&PDFIQ=50

## PDFQR

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** PDFQR

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** PDFQR enables Quick-Render mode on PDF output. It provides high-quality, low-speed PDF rendering or low-quality, high-speed rendering.

Value	Definition
0	Enables high-quality, low-speed PDF rendering. (default)
1	Enables low-quality, high-speed PDF rendering.

**Example:**

&PDFQR=1

## PDFVersion

Specify the PDF version in which the generated document should be saved.

**Category:** Output Parameter

**Compatible with:** Composer 8

**Parameter Name:** PDFVersion

**Parameter Value:** [1.2, 1.3, 1.4, 1.5, 1.6, or 1.7]

**Parameter Requirements:** None

**Parameter Description:** This parameter allows the administrator to specify the PDF version in which the generated document should be saved. As a template input, this parameter supports Microsoft Word, PowerPoint, Excel, and PDF.

### Limitation

This parameter does not work with ZipPDF, ZipFiles, APIMode parameters, or Conga Batch consolidated outputs.

**Example:**

- If the input template is already in PDF format:

```
&PDFVersion=1.5
```

- If the input template is Word, PowerPoint, or Excel format and the administrator wants to convert it to PDF format:

```
&DefaultPDF=1
```

```
&PDFVersion=1.5
```

## PowerPoint Parameters

### Description

The following parameters affect the behavior of PowerPoint templates.

### Parameters

- [APF](#) specifies whether PowerPoint output files are assembled.
- [PPTPreserveRows](#) preserves the table size (rows) in PowerPoint output files.
- [PPTXDUC](#) specifies that category values update in PowerPoint charts.

## PPTPreserveRows

**Category:** PowerPoint Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** PPTPreserveRows

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** PowerPoint Preserve Rows preserves the table size (rows) when merge fields are placed within a table.

Value	Definition
0	Empty rows are removed. (default)
1	The table size is preserved whereby empty rows may appear.

**Example:**



&PPTPreserveRows=1

## PPTXDUC

PowerPoint Do Not Update Categories (PPTXDUC)

**Category:** PowerPoint Parameter

**Compatible with:** Composer 8

**Parameter Name:** PPTXDUC

**Parameter Value:** [0 or 1]

**Parameter Requirements:**

**Parameter Description:** Do not Update Categories in a PowerPoint (.pptx) Chart.

Value	Definition
0	Category values in .pptx charts are updated. (default)
1	Category values in .pptx charts are not updated.

**Example:**

&PPTXDUC=1

## PS0

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** PS0

**Parameter Value:** [0 or 1]]

**Parameter Requirements:** Output to a PDF Template and Enable PDF Encryption (&PS0=1) enabled.

**Parameter Description:** PS0 enables PDF Encryption. It encrypts the output PDF and disables print, copy, and form-fill.

These settings apply to any PDF output, whether from a PDF form or after converting a Word or Excel template.

Value	Definition
0	PDF is not encrypted. (default)
1	PDF is encrypted.

**Example:**

&PS0=1

## PS1

**Category:** Adobe PDF Template Parameter

**Compatible with:** [Composer 7](#) and [Composer 8](#)

**Parameter Name:** PS1

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Output to a PDF Template and Enable PDF Encryption (&PS0=1) enabled.

**Parameter Description:** PS1 allows PDF printing for a merged document.

Value	Definition
0	PDF printing is not permitted.
1	PDF printing is permitted.

**Example:**

&PS1=1

## PS2

**Category:** Adobe PDF Template Parameter

**Compatible with:** [Composer 7](#) and [Composer 8](#)

**Parameter Name:** PS2

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Output to a PDF Template and Enable PDF Encryption (&PS0=1) enabled.

**Parameter Description:** PS2 allows PDF copying. It enables copying text to the clipboard from the output PDF.

Value	Definition
0	PDF copying is not permitted. (default)
1	PDF copying is permitted.

**Example:**

&PS2=1

## PS3

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** PS3

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Output to a PDF Template and Enable PDF Encryption (&PS0=1) enabled.

**Parameter Description:** PS3 allows PDF form fill-in. It enables form fields to be editable in the output PDF (for those fields not specified as read-only).

Value	Definition
0	PDF form fill-in is not permitted. (default)
1	PDF form fill-in is permitted.

**Example:**

&PS0=1

&PS3=1

## PS4

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 8

**Parameter Name:** PS4

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Output to a PDF Template and Enable PDF Encryption (&PS0=1) enabled.

**Parameter Description:** PS4 allows screen readers to have access to PDF content. You can use this parameter to disable screen access.

Value	Definition
0	Screen Readers do not have access to PDF content. (Default)
1	Screen Readers have access to PDF content.

**Example:**

```
&PS0=1
```

```
&PS4=1
```

## PSPW

**Category:** Adobe PDF Template Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** PSPW

**Parameter Value:** [text]

**Parameter Requirements:** &PS0=1, which enables PDF security options.

**Parameter Description:** PSPW sets the password to open the output PDF.

**Example:**

```
&PS0=1
```

```
&PSPW=OpenSesame
```

## QMode

**Category:** Automation-Background Mode Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QMode

**Parameter Value:** [value]

**Parameter Requirements:** None

**Parameter Description:** Parameters for Conga Batch and Conga Trigger. The QMode parameter is required when working with Conga Batch or Conga Trigger solutions.

If you use the QMode parameter to distribute documents (such as email or eSignature) and you also want to save copies of the document, ensure the SC1 parameter is used in conjunction with QMode.

**Example:**

```
&QMode=Attachments
```

### QMode Parameter Values for Release 8

QMode Value	Description
&QMode=Attachments	Attaches the merged output files to the Master Object records
&QMode=Download	<p>Downloads the merged output files to the internet browser.</p> <p>This parameter value is available only to Conga Batch. There is no download option available in Conga Trigger.</p> <p><b>Recommended:</b> &amp;ContentWorkspaceId=[Workspace ID] to specify the Content Workspace (Library) where Conga Batch will store the merged output file if the Conga Batch solution is configured to run on a schedule. Without this parameter, the merged output file from a scheduled Conga Batch solution configured to download will be stored in the running user's personal Salesforce Content Workspace (Library).</p> <p>If either the &amp;DefaultPDF=1 or &amp;FP0=1 parameter is not included in the Conga Batch formula field, the merged output files will be consolidated into a .zip file. However, if one of these PDF output parameters is included in the formula field, the merged output file will be a consolidated PDF.</p>

QMode Value	Description
&QMode=SendEmail	<p>Emails the output file.</p> <p>Requires:</p> <p>&amp;TemplateID unless using only a Conga Email Template (&amp;CongaEmailTemplateID) and no document template.</p> <p>&amp;EmailTemplateId=[Salesforce Email Template ID] or &amp;CongaEmailTemplateID=[Conga Email Template ID]</p> <p>&amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</p> <p>If you cannot use the &amp;EmailToId or &amp;ContactId parameters to specify an email recipient by their Contact Id, you can instead specify &amp;EmailAdditionalTo=[email address] and &amp;EmailSubject=[subject text].</p>
&QMode=Chatter	<p>Posts the merged output file in the Salesforce Chatter feed.</p> <p>By default, Conga Trigger posts files to the Master Object record feed. To override this behavior and post a file to another record's feed, a Chatter Group or a specific User's feed, use &amp;AttachmentParentID.</p>
&QMode=Content	<p>Attaches the merged output files to the running user's personal Workspace (Library) in Salesforce Content.</p> <p>Recommended: &amp;ContentWorkspaceId=[Salesforce Content Workspace ID] to specify an alternate Content Workspace (Library).</p> <p>Does not permit saving additional attachments outside of the initial document created in your Content Workspace (for example, cannot save a copy to Attachments or Chatter in addition to your Content Workspace).</p> <p>Use this method if you are scheduling a content deliver.</p>

QMode Value	Description
&QMode=Documents	<p>Attaches the merged output files to a folder on the Documents tab.</p> <p>Requires: &amp;DocFolderId=[Salesforce Documents Folder ID]</p> <p>Does not permit saving additional attachments outside of the initial document created in your Document folder (i.e. cannot save a copy to Attachments or Chatter in addition to your Document folder).</p>
&QMode=DocuSign	<p>Sends the merged output files via DocuSign (one DocuSign Envelope per Master Object record). Requires:</p> <ul style="list-style-type: none"> <li>• &amp;DocuSignEndPoint= [See <a href="#">DocuSignEndpoint</a> for relevant values]</li> <li>• &amp;DocuSignR1ID=[Salesforce Contact, Lead, or User ID]</li> <li>• &amp;DocuSignR1Role= [See <a href="#">DocuSignR1Role</a> for relevant values]</li> <li>• &amp;DocuSignR1Type= [See <a href="#">DocuSignR1Type</a> for relevant values]</li> <li>• DocuSign Credentials on the USER record</li> <li>• The user's email address on their User record in Salesforce must match the user's email address in DocuSign.</li> </ul> <p>There are many combinations of possible DocuSign parameters. Please see <a href="#">DocuSign Integration Parameters</a> for more information.</p>
&QMode=EchoSign	<p>Sends the merged output files via Adobe eSign (one Adobe eSign Agreement per Master Object record). For more information, see <a href="#">Adobe eSign Integration Parameters</a>.</p>
&QMode=EchoSignCreate	<p>Merges the output files and attaches each to an Adobe eSign Agreement but does not automatically send those Agreements.</p>

QMode Value	Description
&QMode=GoogleDrive	<p>Saves the merged output files in Google Drive.</p> <p>Requires the Google Refresh Token on Conga Batch Setup tab.</p> <p>Use &amp;OFP to specify in which folder the merged output files are to be saved.</p> <p>Remove &amp;SC0=1 and &amp;SC1=Google from the Conga Batch formula field to prevent the storage of duplicate copies of the merged output file in Google Drive.</p>
&QMode=GoogleDocs	<p>Converts the merged output files to Google Documents, Spreadsheets, or Slides, which are saved in Google Drive.</p> <p>Requires Google Refresh Token on Conga Trigger Setup tab.</p> <p>Recommended: &amp;OFP to specify in which folder the merged output files are to be saved.</p>
&QMode=SalesforceFile	<p>Sends the merged output files to Salesforce Files for the Master Object.</p> <p>The &amp;SC0=1 and &amp;SC1=SalesforceFile parameters are required in the URL for this QMode.</p>
&QMode=Sertifi	<p>Sends the merged output files via Sertifi. The browser page refreshes and lands on the Sertifi EContract. There are many combinations of possible Sertifi parameters. For more information, see Sertifi Integration Parameters.</p>
&QMode=SpringCM	<p>Saves the merged output files in a SpringCM folder, as defined by the SpringCM configuration rules.</p>
&QMode=Box	<p>Sends the output documents to Box.</p>
&QMode=OneDrive	<p>Sends the output documents to the user's OneDrive repository.</p>
&QMode=CongaSign	<p>Sends the output documents to Conga Sign.</p>
&QMode=Contracts	<p>Sends the output documents for negotiation using Conga Contracts for Salesforce's Send for Negotiation email service.</p>



QMode Value	Description
&QMode=AmazonS3	Sends the output documents to AmazonS3
&QMode=LinkService	<p>Sends the output documents with Composer Link Service.</p> <p><b>Requires:</b></p> <ul style="list-style-type: none"> <li>• Composer File Storage enabled in Composer Setup.</li> <li>• &amp;CongaEmailTemplateID to designate a specific Conga Email Template</li> <li>• The {Conga_Doc_Link} merge tag to successfully generate the document link in the email.</li> <li>• &amp;TemplateID unless using only a Conga Email Template and no document template.</li> <li>• &amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</li> </ul> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>✔ Use the NETId parameter to send Notification Emails with Composer Link Service.</p> </div>
&QMode=EmailLink	Allows you to send the composer-generated link via a unique URL in a Conga Email Template with Trigger or Batch.
&QMode=GenerateLink	Allows you to generate a unique URL for the Composer generated document and adds that link to the document history record.

### QMode parameter values for Release 7

Value	Definition
1	<p>Download the output file to your computer.</p> <p>(One of the following parameters should be used with QMode=1: LG3=1 OR LG4=1.) (default)</p>
2	<p>Attaches the merged output to the master object.</p> <p>LG4=1 is the only Activity Logging parameter needed when using QMode=2.</p> <p>Analogous to Qmode=Attachments</p>

Value	Definition
3	<p>Emails the output file.</p> <p>Requires &amp;Templateld</p> <p>For an Opportunity with a Primary Contact (Contact Roles), there is no need to set the &amp;EmailTold. Also, if you use &amp;EmailTemplateld, you must specify the &amp;EmailTold.</p>
4	Delivers the output via Adobe eSign. Requires Adobe eSign parameters.
5	<p>Attaches the output file to a folder on the Documents tab.</p> <p>QMode=5 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
6	<p>Attaches the output file to a workspace in Salesforce Content.</p> <p>QMode=6 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
7	Delivers the output via DocuSign. Requires DocuSign parameters.
94	Creates an Adobe eSign Agreement but does not automatically deliver it. Requires Adobe eSign parameters.

## QueryID

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QueryID

**Parameter Value:** [Salesforce ID of a Query record]

**Parameter Requirements:** None

**Parameter Description:** QueryID retrieves the SOQL query stored in the specified Salesforce record, executes the SOQL query and makes the SOQL query results available through a dataset as seen on the Template Builder window.

It may optionally include filter criteria.

You may include up to 50 QueryID values in a Conga Composer URL, thus making 50 queries accessible in a single merge operation. Conga Composer will retrieve a query that contains a maximum of 2,500 rows.

Example of two Queries: `&QueryId=[contacts]a00143532453dg32,[cases]a00143532453dg32`

Note on Query Aliases: An optional query alias may precede the query ID: Enclose the alias in square brackets Minimum 2 characters, maximum 20 Alphanumeric characters only

Reserved sheet names may not be used (Master, Org, etc.) The alias must be unique within the button URL Query With Alias: `&QueryId=[Contacts]a00143532453dg3` Queries with

Example of two Aliases: `&QueryId=[Accounts]a00143532453dg32,[Cases]a0043532453dgF2`

#### Example:

`&QueryId=[queryAlias]a0341000000ABcd`

## QueryStringField

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QueryStringField


**Parameter Value:** [APIFieldName]

**Parameter Description:** The QueryStringField parameter gives you the ability to add Conga Composer parameters into a separate Salesforce formula field, and then reference that custom field in the merge process.

### To reference the custom field in the merge process:

1. Create a custom formula field on the object on which Composer is built
2. Add Composer parameters to the custom field as a formula, similar to what you do when creating a formula field for Conga Batch
3. Reference the API Name of the newly created formula field as the value of the QueryStringField parameter in the Composer button URL, as shown above

See [Use Case for Query String Fields](#) for more information.

 You can include up to three QueryStringField parameters: QueryStringField, QueryStringField1, and QueryStringField2.

#### Example:

&QueryStringField=CongaSign\_recipients\_\_c

## QueryStringField1

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QueryStringField1


**Parameter Value:** [APIFieldName]

**Parameter Description:** The QueryStringField1 parameter gives you the ability to add Conga Composer parameters into a separate Salesforce formula field, and then reference that custom field in the merge process.

To reference that custom field in the merge process:

1. Creating a custom formula field on the object on which Composer is built.
2. Adding Composer parameters to the custom field as a formula, similar to what you do when creating a formula field for Conga Batch.
3. Referencing the API Name of the newly created formula field as the value of the QueryStringField parameter in the Composer button URL, as shown above.

See [Use Case for Query String Fields](#) for more information.

 You can include up to three QueryStringField parameters: QueryStringField, QueryStringField1, and QueryStringField2.

**Example:**

&QueryStringField=CongaSign\_recipients\_\_c

&QueryStringField1=Custom\_field\_\_c

## QueryStringField2

**Category:** Advanced Control Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QueryStringField2

**Parameter Value:** [APIFieldName]


**Parameter Description:** The QueryStringField2 parameter gives you the ability to add Conga Composer parameters into a separate Salesforce formula field, and then reference

that custom field in the merge process.

To reference that custom field in the merge process:

1. Creating a custom formula field on the object on which Composer is built.
2. Adding Composer parameters to the custom field as a formula, similar to what you do when creating a formula field for Conga Batch.
3. Referencing the API Name of the newly created formula field as the value of the QueryStringField parameter in the Composer button URL, as shown above.

See [Use Case for Query String Fields](#) for more information.

 You can include up to three QueryStringField parameters: QueryStringField, QueryStringField1, and QueryStringField2.

#### Example:

```
&QueryStringField=CongaSign_recipients__c
&QueryStringField1=Custom_field__c
&QueryStringField2=Custom_field_two__c
```

## QVar0Format

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QVar0Format

**Parameter Value:** [code]

**Parameter Requirements:** None

**Parameter Description:** Querystring Variable Format

Formats the result of a QVar query. The value consists of a five-digit "code" where the first digit is the value for A, the second B, the third C, and so on. Use the values below to define the desired formatting results.

A = Delimiter between data elements [0=none, 1=comma, 2=pipe, 3=semi-colon, 4=colon, 5=hyphen] (2-Pipe is useful for delimiting values that are passed to a Report/Query)

B = Symbol to enclose each data element [0=none, 1=apostrophes, 2=quotes]

C = Symbol after delimiter [0=none, 1=space]

D = Truncate each data element to 15-characters [0=off, 1=on] (Useful when passing values to a Salesforce Report/Query)

E = Reserved for future use

Also available: QVar1Format, QVar2Format

**Example:**

&QVar0Format=20010

## QVar0Id

**Category:** Data Gathering Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QVar0Id

**Parameter Value:** [ID or {QVar0}]

**Parameter Requirements:** The 15-character Salesforce ID of a query record. May optionally include filter criteria.

**Parameter Description:** QVar, which stands for Querystring Variable, lets you execute a Conga Query and use the result of the query as a value for another Conga Composer parameter. Three QVar ID parameters are available: QVar0Id, QVar1Id, and QVar2Id.

Usage Guidelines:

- Set &QVar0Id to the Salesforce ID of a Conga Query record. Optionally, you may pass filter criteria to the Query.
- The Conga Query must return only one column of data and must not retrieve more than **50** rows of data.
- Multiple rows of data are returned as a comma-separated list.
- Use {QVar0} to insert the QVar0 result into another Conga Composer parameter.
- The resulting values for all QVar variables appear on the Template Builder window.
- The resulting value may be inserted into most Conga Composer parameters, including &ReportId, &QueryId, &TemplateId, &TemplateGroup and the Email parameters.
- Duplicate values are automatically stripped from the resulting data.
- QVars are not recommended for retrieving date values.
- Use QVars to retrieve Conga Template IDs to pass to &TemplateID when a solution uses &SelectTemplates.
- May be used with &MFTSValue parameter.

**Example:**

&QVar0ID=a03f4000000vnCL

## QVar1Format

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QVar1Format

**Parameter Value:** [code]

**Parameter Requirements:** &LG2=14

**Parameter Description:** Querystring Variable Format

Formats the result of a QVar query.

The [code] value follows the convention below.

Code: [A][B][C][D][E]

A = Delimiter between data elements [0=none, 1=comma, 2=pipe, 3=semi-colon, 4=colon, 5=hyphen] (2-Pipe is useful for delimiting values that are passed to a Report)

B = Symbol to enclose each data element [0=none, 1=apostrophes, 2=quotes]

C = Symbol after delimiter [0=none, 1=space]

D = Truncate each data element to 15-characters [0=off, 1=on] (Useful when passing values to a Salesforce Report)

E = Reserved for future use

Also available: QVar1Format, QVar2Format

## QVar1Id

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QVar1Id

**Parameter Value:** [ID or {QVar1}]

**Parameter Requirements:** The 15-character Salesforce ID of a query record. May optionally include filter criteria.

**Parameter Description:** QVar, which stands for Querystring Variable, lets you execute a Conga Query and use the result of the query as a value for another Conga Composer parameter. Three QVar ID parameters are available: QVar0Id, QVar1Id, and QVar2Id.

## Usage Guidelines:

- Set &QVar0Id to the Salesforce ID of a Conga Query record. Optionally, you may pass filter criteria to the Query.
- The Conga Query must return only one column of data and must not retrieve more than 50 rows of data.
- Multiple rows of data are returned as a comma-separated list.
- Use {QVar0} to insert the QVar0 result into another Conga Composer parameter.
- The resulting values for all QVar variables appear on the Template Builder window.
- The resulting value may be inserted into most Conga Composer parameters, including &ReportId, &QueryId, &TemplateId, &TemplateGroup and the Email parameters.
- Duplicate values are automatically stripped from the resulting data.
- QVars are not recommended for retrieving date values.
- Use QVars to retrieve Conga Template IDs to pass to &TemplateID when a solution uses &SelectTemplates.
- Can be used with &MFTSValue parameter.

**Example:**

&QVar0ID=a03f4000000vnCL

## QVar2Format

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QVar2Format

**Parameter Value:** [code]

**Parameter Description:** Querystring Variable Format

Formats the result of a QVar query.

The [code] value follows the convention below.

Code: [A][B][C][D][E]

A = Delimiter between data elements [0=none, 1=comma, 2=pipe, 3=semi-colon, 4=colon, 5=hyphen] (2-Pipe is useful for delimiting values that are passed to a Report)

B = Symbol to enclose each data element [0=none, 1=apostrophes, 2=quotes]

C = Symbol after delimiter [0=none, 1=space]

D = Truncate each data element to 15-characters [0=off, 1=on] (Useful when passing values to a Salesforce Report)



E = Reserved for future use

Also available: QVar1Format, QVar2Format

## QVar2Id

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** QVar2Id

**Parameter Value:** [ID or {QVar2}]

**Parameter Requirements:** The 15-character Salesforce ID of a query record. May optionally include filter criteria.

**Parameter Description:** QVar, which stands for Querystring Variable, lets you execute a Conga Query and use the result of the query as a value for another Conga Composer parameter. Three QVar ID parameters are available: QVar0Id, QVar1Id, and QVar2Id.

Usage Guidelines:

- Set &QVar0Id to the Salesforce ID of a Conga Query record. Optionally, you may pass filter criteria to the Query.
- The Conga Query must return only one column of data and must not retrieve more than 50 rows of data.
- Multiple rows of data are returned as a comma-separated list.
- Use {QVar0} to insert the QVar0 result into another Conga Composer parameter.
- The resulting values for all QVar variables appear on the Template Builder window.
- The resulting value may be inserted into most Conga Composer parameters, including &ReportId, &QueryId, &TemplateId, &TemplateGroup and the Email parameters.
- Duplicate values are automatically stripped from the resulting data.
- QVars are not recommended for retrieving date values.
- Use QVars to retrieve Conga Template IDs to pass to &TemplateID when a solution uses &SelectTemplates.
- Can be used with &MFTSValue parameter.

**Example:**

&QVar2ID=a03f4000000vnCL

## ReportId

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** ReportId

**Parameter Value:** [ID]

**Parameter Requirements:** The 15-character Salesforce ID of a Report. May optionally include filter criteria.

**Parameter Description:** Retrieves the contents of the specified report (identified by its 15-character Salesforce Id) and makes the report contents available through a dataset as seen on the Template Builder window.

You may include up to 50 Report ID values in a Conga Composer URL, thus making 50 reports accessible in a single merge operation. Conga Composer will retrieve a report that contains a maximum of 2,500 rows.

 **Important**

While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

An optional report alias may precede the report ID.

- Enclose the alias in square brackets
- Minimum 2 characters, maximum 20
- Alphanumeric characters only
- Reserved sheet names may not be used (Master, Org, etc.)
- The alias must be unique within the button URL

**Example:**

&ReportId=00143532453dg3A

(Two Reports)

&ReportId=00O43532453dg3A,00O43532453dgF2

(Report with Alias)

&ReportId=[MyContacts]00143532453dg3A

(Report with Two Aliases)

&ReportId=[MyAccounts]00O43532453dg3A,[MyCases]00O43532453dgF2

# ReportVar

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** ReportVar

**Parameter Value:** [integer]

**Parameter Requirements:** At least two ReportID values.

**Parameter Description:** The ReportVar parameter (Report Variable) is a placeholder to represent a value passed to multiple reports, thereby resulting in a shorter overall URL length.

In the example below, the value of ReportVar (California|New+York|New+Jersey) would be passed to the selection criteria for each report. Note the placement of {ReportVar} to indicate where to insert the value.

**Example:**

```
&ReportVar=California|New+York|New+Jersey
```

```
&ReportId=00O987654321?pv0={ReportVar},
```

```
00O876543210?pv0={ReportVar},
```

```
00O654321033?pv0={ReportVar}
```

## Important

While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.

# ReturnPath

**Category:** Advanced Control Parameters

**Compatible with:** Composer 8

**Parameter Name:** ReturnPath

**Parameter Value:** [Record Id or redirect URL]

**Parameter Requirements:** None

**Parameter Description:** Return Path specifies where to refresh after merging.

**Example:**

```
&ReturnPath={!Opportunity.ID}
```

-or-

If you want to set Account's recent page (*<http://mydomain.salesforce.com/lightning/o/Account/list?filterName=Recent>*) as a ReturnPath, the parameter value should be:

```
&ReturnPath=lightning/o/Account/list?filterName=Recent
```

## RM

**Category:** Conga Contracts

**Compatible with:** Composer 8

**Parameter Name:** RM

**Parameter Value:** 0, 1

**Parameter Requirements:** Composer Enterprise Edition

**Parameter Description:** RM (Recursive Merge) is meant specifically for Composer solutions used with Conga Contracts and allows for dynamic content with embedded merge fields in background mode.

Recursive merge is a Composer Enterprise Edition feature that allows customers to store merge text inside fields in Salesforce. Composer queries the clause or text fields to create the subject document. A second merge is performed to identify and populate any merge fields that might be contained in the initial merge results.

 **Important**

Recursive Merge is NOT supported with Excel and PPT. Recursive Merge currently only supports fields from the Master Dataset.

Value	Definition
0	Recursive Merge disabled.
1	Recursive Merge enabled.

**Example:**

```
&RM=1
```

## RTLFont

**Category:** Word Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** RTLFont

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &LG2=14

**Parameter Description:** Enables right-to-left merging in Word templates for languages (such as Arabic and Hebrew) that are read right-to-left.

Identifies the entire template as right-to-left text; if instead only certain fields should be identified as right-to-left, please use the Word field prefix RTL: (see the Conga Composer Release 8 Word Templates Guide for further information).

Word templates that include right-to-left languages are better supported by the Conga Composer PDF rendering engine. We recommend rendering your Word template as a PDF for optimum results.

Value	Definition
0	Right-to-left text is disabled.
1	Right-to-left text is enabled.

**Example:**

&RTLFont=1

## Salesforce1 Parameters

### Description

Conga Composer solutions work with Salesforce1 to provide mobile solutions. For more information, see [Composer for Salesforce1](#).

### Parameter

- [SF1](#) enables the use of Conga Composer for Salesforce1.

# Salesforce Chatter Parameters

## Description

Conga Composer can store output documents in Salesforce Chatter, rather than as attachments under the Notes & Attachments related list.

Conga Composer saves files to Chatter as part of the Activity Logging functions. We recommend the following settings for integrating with Chatter:

[&SC0=1](#) (Enables Save a Copy)

[&SC1=Chatter](#) (Enables File Attachments to log to Chatter)

If you wish to override the default behavior where files are saved on the Chatter Feed for the master object, you may use the AttachmentParentId parameter to identify the record on which the file should be posted. You may set AttachmentParentId to a User Id, Group ID or record Id.

## Parameters

- [AssembleChatter](#) allows multiple files to be attached to a single Chatter post.
- [ChatterCommunityId](#) specifies the Chatter Community receiving an output file attachment.

# Salesforce Content Parameters

## Description

Conga Composer can store output documents in Salesforce Content, rather than as attachments under the Notes & Attachments related list. Composer can also retrieve items from Salesforce Content or Salesforce Documents to be included in the merge process.

Set up Salesforce content integration before using custom field settings. For more information, see [Set Up the Salesforce Content Integration](#).

Salesforce Documents that reference a URL rather than a static file are not supported for security reasons.

## Parameters

- [ContentCustomFieldName](#) sets the name of a custom field to set on a Salesforce content record.
- [ContentCustomFieldValue](#) sets the value of the corresponding custom field.
- [ContentVisible](#) enables Salesforce content as an output option.

- [ContentWorkspaceId](#) sets the Salesforce ID of the content workspace under which attachments are stored.
- [ContentDocumentId](#) specifies the Salesforce ID of the Salesforce File record ID where you want to use Salesforce Files versioning features.
- [contentShareType](#) determines the permission type granted to the user for a shared file.
- [contentVisibility](#) determines whether a file is available to all users, internal users, or shared users.

## SC0

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** SC0

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None.

**Parameter Description:** SC0 (Save a Copy) saves a copy of the merged output file.

Value	Definition
0	Disables Save a Copy. (Default)
1	Enables Save a Copy.

**Example:**

```
&SC0=1
```

## SC1

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** SC1

**Parameter Value:** [Attachments, SalesforceFile, Chatter, Content, Documents, Google, SpringCM]

**Parameter Requirements:** SC0

**Parameter Description:** SC1 (Save a Copy Location) defines where a copy of the merged output file is stored.

Value	Definition
Attachments	<p>Saves a copy of the merged file in Notes &amp; Attachments on the master object.</p> <div style="border: 1px solid #c8e6c9; padding: 10px; margin-top: 10px;"> <p>✔ You can also save the merged file to Notes &amp; Attachments of a parent object through a look-up using <a href="#">AttachmentParentId</a>.</p> </div>
Chatter	<p>Saves a copy of the merged file in the Chatter feed for the master object.</p> <p>Requires Edit My Own Posts permission for user profile. Also update System Settings to check Moderate Chatter.</p> <p>You can also save the merged file to the Chatter feed of a parent object via a look up using <a href="#">AttachmentParentId</a>.</p>
Content	<p>Saves a copy of the merged file in Salesforce Content.</p> <p>The merged file is saved to the Private Library by default. Use the <a href="#">ContentWorkspaceId</a> parameter to save to a different content library.</p>
Documents	<p>Saves a copy of the merged files in Salesforce Documents.</p> <p>The merged file is saved to My Personal Documents by default. Use the DocFolderID parameter to save to a different document folder.</p> <p>Spring '16 and newer Salesforce orgs, by default, will upload files to the Attachments related list on records as Salesforce Files. In order for Conga Templates to work properly and to save Conga output documents as attachments, you must disable this setting. View New Customers: Disable Salesforce Files Settings for additional information.</p>
Google	Saves a copy of the merged file in the Google Drive.



Value	Definition
SalesforceFile	Saves a copy of the merged file in Salesforce Files for the Master Object.
SpringCM	Saves a copy of the merged file in SpringCM.
Box	Saves a copy of the merged file in Box.
OneDrive	Saves a copy of the merged file in the user's OneDrive repository.
BigTinCan	Saves a copy of the merged file in BigTinCan.
AmazonS3	Saves a copy of the merged file in Amazon S3.

**Example:**`&SC0=1``&SC1=Attachments`

## SCHP

**Category:** Data Gathering Parameter**Compatible with:** Composer 7 and Composer 8**Parameter Name:** SCHP**Parameter Value:** [0, 1 or 2]**Parameter Requirements:** None**Parameter Description:** Strip Punctuation from Merge Field Names (Strip: Comma, Hyphen, Parenthesis)

This setting removes punctuation from merge field names, making it possible to refer to those fields in Excel templates and also helps Word templates when parenthesis or other characters cause merge problems.


Value	Definition
0	Allows merge fields to contain parentheses.
1	Removes parentheses from merge field names.

Value	Definition
2	Removes all punctuation (including colons) from the merge field name.

**Example:**

```
&SCHP=2
```

## SCMEndpoint

 The SpringCM integration has been deprecated. Features will not be disabled; however, the Conga Support team cannot provide assistance with the integration.

**Category:** SpringCM Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** SCMEndpoint

**Parameter Value:** [Prod, UAT]


**Parameter Requirements:** None

**Parameter Description:** Configures the Conga Composer button to connect to a specific SpringCM endpoint.

**Example:**

```
&SCMEndpoint=Prod
```

## SCMPreview

 SpringCM integration has been deprecated. Features will not be disabled; however, the Conga Support team cannot provide assistance with the integration.

**Category:** SpringCM Parameter

**Compatible with:** Composer 8

**Parameter Name:** SCMPreview

**Parameter Value:** [0, 1]

**Parameter Requirements:** DS7=8

**Parameter Description:** Displays the SpringCM document preview window after the merge process completes.

The solution must run in background mode.

Value	Definition
0	The SpringCM Preview Document window does not appear after the solution completes.
1	The SpringCM Preview Document window appears after the solution completes.

**Example:**

```
&DS7=8
```

```
&SCMPreview=1
```

## SCMVisible

- ❗ SpringCM integration has been deprecated. Features will not be disabled; however, the Conga Support team cannot provide assistance with the integration.

**Category:** SpringCM Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** SCMVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Makes the SpringCM button visible on the Conga Composer dialog box.

Value	Definition
0	The SpringCM button is not visible. (default)
1	The SpringCM button is visible.

**Example:**

```
&SCMVisible=1
```

## SelectAttachments

**Category:** Output Parameter


**Compatible with:** Composer 8

**Parameter Name:** SelectAttachments

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** Displays the 20 most recent attachments or Salesforce files from the master object. Results display in descending order by created date.

 The Allow use of Local Templates setting must be checked in Composer Setup for the Attachments tab to appear in the Composer user interface.

Value	Definition
0	Select attachments or Salesforce files is disabled.
1	Select attachments or Salesforce files is enabled.

**Example:**

```
&SelectAttachments=1
```

## Select Templates

**Category:** Template Selection Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** SelectTemplates

**Parameter Value:** [0, 1 or 2]

**Parameter Requirements:** None

**Parameter Description:** The SelectTemplates parameter displays the Multiple Template Selection dialog box which allows users to select and arrange up to 20 templates.

You can select templates from any of the following sources:

- Conga Template Manager
- Salesforce Content
- Salesforce Documents Tab

The Salesforce Content Title and Description fields can be located by a search using the dialog box. At this time, the attachments are not indexed.

Salesforce Content may be suppressed by including the &HideContent=1 parameter. Salesforce Content can be set as the default option by including the &DefaultPane=1 parameter.

Notes:

- This parameter works especially well with the TemplateGroup parameter, which can be used to ensure that each template has appropriate merge fields.
- This parameter also works especially well as an inline window, rather than as a traditional pop-up window.
- If Background Mode (&DS7) is also included, a template must be selected.
- This parameter works with a QVar returning a list of template IDs only when using an OAuth button. It does not work for the Lightning Component.

Value	Definition
0	Select Templates is disabled. (default)
1	Select Templates is enabled.
2	Select Templates is enabled but the Multiple Template Selection dialog box is skipped, allowing for the selection of multiple templates on an ad-hoc basis from the regular Conga Composer user interface.

**Example:**

```
&SelectTemplates=1
```

## SertifiCCSignerEmail

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiCCSignerEmail

**Parameter Value:** [Email Address]

**Parameter Requirements:** None

**Parameter Description:** Specify the CC Signer email address.

**Example:**

```
&SertifiCCSignerEmail=mergey.composer@congamerge.com
```

## SertifiCustomSettings

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiCustomSettings

**Parameter Value:** [Text]

**Parameter Requirements:** Sertifi for Salesforce version 5.87 or later.

**Parameter Description:** Sertifi Custom Settings: Since there are multiple Sertifi Application Settings available for end-users to create and use, SertifiCustomSettings specifies the Sertifi Custom Setting to use during the merge process. Salesforce Setup | Develop | Custom Settings | Sertifi Application Settings | Manage. Without this parameter, the Sertifi Custom Setting that will be used is "Settings" -- their default Custom Setting.

**Example:**

`&SertifiCustomSettings=Quote`

## SertifiEmailTo

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiEmailTo

**Parameter Value:** [Email Address]

**Parameter Requirements:** None

**Parameter Description:** Specifies the recipient's email address.

**Example:**

`&SertifiEmailTo=mergey@congamerge.com`

## SertifiExpirationDate

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiExpirationDate

**Parameter Value:** [Expiration Date in YYYY-MM-DD format]

**Parameter Requirements:** `&SertifiVisible=1`

**Parameter Description:** Sets the expiration date of the eContract.

**Example:**

`&SertifiVisible=1`

`&SertifiExpirationDate=2015-10-31`

## SertifiFileName

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiFileName

**Parameter Value:** [text]

**Parameter Requirements:** Sertifi for Salesforce version 5.87 or later.

**Parameter Description:** Defines the Folder Name on the Sertifi EContract. The Folder Name applies to the collection of files to be delivered.

You can also use dynamic field content for the parameter value: `&SertifiFileName={!Opportunity.Name}+--+{!Today}`

**Example:**

`&SertifiFileName=Conga+Sertifi+Contract`

## Sertifi Integration Parameters

### Description

Sertifi delivers Conga Composer documents via email, much like Salesforce itself, but adds the ability to acquire, track and file signatures. You can learn more about Sertifi for Salesforce here: <http://www.sertifi.com/>

Sertifi is available separately on the AppExchange. You must subscribe to that service to use this integration parameter.

### Parameters

- [SertifiCCSignerEmail](#) specifies the email address for a CC signer.
- [SertifiCustomSettings](#) creates a Sertifi operation from custom objects or quotes.
- [SertifiEmailTo](#) specifies a recipient's email address.
- [SertifiFileName](#) specifies the folder name on the Sertifi EContract.
- [SertifiPaymentAmount](#) specifies the payment amount on the Sertifi EContract.
- [SertifiPaymentName](#) specifies the Payment Name on the Sertifi EContract.
- [SertifiPaymentType](#) specifies the payment type on a Sertifi EContract.
- [SertifiSigner1Email](#) specifies the email address of the first recipient/signer of a Sertifi email.
- [SertifiSigner1ID](#) specifies the Contact ID of the first recipient/signer of a Sertifi email.
- [SertifiVisible](#) changes the Composer Action picklist to Sertifi.

## Sertifi Pass Through Method

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** Sertifi Pass Through Method

**Parameter Value:** [Sertifi Parameter=Sertifi Value]

**Parameter Requirements:** Sertifi E-Sign for Salesforce.com version 216.316 or newer.

**Parameter Description:** Sertifi Pass-Through Method. This parameter allows the pass-through of Sertifi parameters and values in the `&Sertifi[SertifiParameter]=[SertifiValue]` format. These Sertifi parameters and values should be obtained directly from Sertifi or through the Sertifi documentation.

**Example:**

```
&SertifiApplySignature=1
```

## SertifiPaymentAmount

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiPaymentAmount

**Parameter Value:** [number]

**Parameter Requirements:** Sertifi for Salesforce version 5.87 or later.

**Parameter Description:** Defines the payment amount in U.S. dollars on the Sertifi EContract when requesting payment.

**Example:**

```
&SertifiPaymentAmount=100
```

## SertifiPaymentName

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiPaymentName



**Parameter Value:** [text]

**Parameter Requirements:** Sertifi for Salesforce version 5.87 or later

**Parameter Description:** Defines the Payment Name on the Sertifi EContract when requesting payment.

**Example:**

&SertifiPaymentName={!Account.Name}+Legal+Fees

## SertifiPaymentType

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiPaymentType

**Parameter Value:** [OneTime or Recurring]

**Parameter Requirements:** Sertifi for Salesforce version 5.87 or later.

**Parameter Description:** Defines the payment type on the Sertifi EContract when requesting payment.

Value	Definition
OneTime	Specifies the Payment Type as a one-time payment.
Recurring	Specifies the Payment Type as a monthly payment.

**Example:**

&SertifiPaymentType=OneTime

## SertifiSigner1Email

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner1Email

**Parameter Value:** [Email address]

**Parameter Requirements:** None

**Parameter Description:** Specifies the recipient of the merged document to be delivered via Sertifi. `&SertifiSigner1Email` specifies the Signer1 email address, `&SertifiSigner2Email` specifies the Signer2 email address.

Also available: `SertifiSigner2Email`. The email address must be associated with only one Salesforce Contact record. Email addresses that are associated with more than one Contact record will cause errors.

**Example:**

```
&SertifiSigner1Email=ccomposer@congamerge.com
```

OR

```
&SertifiSigner1Email={!Contact.Email}
```

## SertifiSigner1Id

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner1Id

**Parameter Value:** [Contact ID]

**Parameter Requirements:**

- A Salesforce Contact ID or a field containing a Contact ID.
- Recipients 3, 4, and 5 require automatic Sertifi delivery via the API using either `&DS7=301`, `&DS7=302`, or `&QMode=Sertifi`.
- Sertifi for Salesforce version 5.87 or later.

**Parameter Description:** Specifies additional recipients of the merged document to be delivered via Sertifi by their Contact ID. These additional recipients may only be specified when running the Composer solution over the API.

There are five signer parameters available: `SertifiSigner1Id`, `SertifiSigner2Id`, `SertifiSigner3Id`, `SertifiSigner4Id`, and `SertifiSigner5Id`.

You can also use dynamic field content for the parameter value: `&SertifiSigner1Id={!Contact.Id}`

**Example:**

```
&SertifiSigner1Id=003r570ygN42fGe
```

## SertifiSigner2Email

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner2Email

**Parameter Value:** [Email address]

**Parameter Description:** Specifies the recipient of the merged document to be delivered via Sertifi. `&SertifiSigner1Email` specifies the Signer1 email address, `&SertifiSigner2Email` specifies the Signer2 email address.

The email address must be associated with only one Salesforce Contact record. Email addresses that are associated with more than one Contact record will cause errors.

## SertifiSigner2Id

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner2Id

**Parameter Value:** [Contact ID]

**Parameter Description:** Specifies additional recipients of the merged document to be delivered via Sertifi by their Contact ID. These additional recipients may only be specified when running the Composer solution over the API.

There are five signer parameters available: `SertifiSigner1Id`, `SertifiSigner2Id`, `SertifiSigner3Id`, `SertifiSigner4Id`, and `SertifiSigner5Id`.

You can also use dynamic field content for the parameter value: `&SertifiSigner1Id={!Contact.Id}`

## SertifiSigner3Email

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner3Email

**Parameter Value:** [Email address]

**Parameter Requirements:**

- Automatic Sertifi delivery via the API using either `&DS7=301`, `&DS7=302`, or `&QMode=Sertifi`
- Sertifi for Salesforce version 5.87 or later

**Parameter Description:** Specifies additional recipients of the merged document to be delivered via Sertifi.

There are five SertifiSigner Email parameters: SertifiSigner1Email, SertifiSigner2Email, SertifiSigner3Email\*, SertifiSigner4Email\*, and SertifiSigner5Email\*.

\* These recipients may only be specified when running the Composer solution over the API.

You can also use dynamic field content for the parameter value: `&SertifiSigner3Email={!Contact.Email}`

**Example:**

`&SertifiSigner3Email=Carlos.Composer@congamerge.com`

## SertifiSigner4Email

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner4Email

**Parameter Value:** [Email address]

**Parameter Requirements:** None

**Parameter Description:** Specifies additional recipients of the merged document to be delivered via Sertifi.

There are five SertifiSigner Email parameters: SertifiSigner1Email, SertifiSigner2Email, SertifiSigner3Email\*, SertifiSigner4Email\*, and SertifiSigner5Email\*.

\* These recipients may only be specified when running the Composer solution over the API.

You can also use dynamic field content for the parameter value: `&SertifiSigner3Email={!Contact.Email}`

## SertifiSigner5Email

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiSigner5Email

**Parameter Value:** [Email address]

**Parameter Requirements:** None

**Parameter Description:** Specifies additional recipients of the merged document to be delivered via Sertifi.

There are five SertifiSigner Email parameters: SertifiSigner1Email, SertifiSigner2Email, SertifiSigner3Email\*, SertifiSigner4Email\*, and SertifiSigner5Email\*.

\* These recipients may only be specified when running the Composer solution over the API.

You can also use dynamic field content for the parameter value: `&SertifiSigner3Email={!Contact.Email}`

## SertifiVisible

**Category:** Sertifi Integration Parameter

**Compatible with:** Composer 8

**Parameter Name:** SertifiVisible

**Parameter Value:** [0 or 1]

**Parameter Requirements:** &LG2=14

**Parameter Description:** Changes the Action picklist in the Composer dialog box to Sertifi and updates the Merge button to Merge and Sertifi. Sertifi is available separately on the AppExchange. You must subscribe to that service to use this integration parameter.

This parameter is required for all other Sertifi parameters.

Value	Definition
0	Sertifi is not the designated Action. (default)
1	The Action is set to Sertifi.

**Example:**

&SertifiVisible=1

## SETI

**Category:** Email Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** SETI

**Parameter Value:** [Contact Id]

**Parameter Requirements:** &EmailAdditionalTo

**Parameter Description:** SETI stands for Skip to Email Id. If a solution does not specify an EmailTold, but would need to relate email activity history (including enhanced email) to the Contact while using a custom email address specified by EmailAdditionalTo, then the SETI parameter should be used as indicated in the example.

**Parameter Example:**

```
&EmailAdditionalTo=firstname.lastname@company.com  
&SETI={!Case.ContactId}
```

## SF1

**Category:** Salesforce1 Integration Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** SF1

**Parameter Value:** [0 or 1]

**Parameter Requirements:** DS7

**Parameter Description:** The SF1 parameter enables the use of Conga Composer for Salesforce1. This parameter is required for a button to appear on a mobile device. For more information, see [Composer for Salesforce1](#).

Due to Salesforce.com limitations, Conga Composer for Salesforce1 works with all custom objects and only these standard objects:

- Account
- Campaign
- Case

- Contact
- Custom objects
- Group
- Lead
- Opportunity
- Quote
- Order

Parameter Value	Definition
0	Conga Composer for Salesforce1 is not enabled. (default)
1	Conga Composer for Salesforce1 is enabled.

**Example:**

```
&DS7=11
&SF1=1
```

## SMD

**Category:** Data Gathering Parameter

**Compatible with:** Composer 7 & Composer 8

**Parameter Name:** SMD

**Parameter Value:** [0 or 1]

**Parameter Description:** Substitutes the master object data with another data set. the report or query must be aliased with [Master]. The Query must be limited to only return a single row of data.

The benefit of using this parameter is that in a Word Template, you do not need to use detailed region tags (e.g. {{TableStart:ReportData}}), simply place the Report Data fields wherever you like on the template, as though they were Master fields.

Applicable only when using exactly one Report as the only data source.

Value	Definition
0	The master data is not substituted.
1	Substitutes the master object data with another data set.

## SpringCM Integration Parameters

⚠ SpringCM integration has been deprecated. Features will not be disabled; however, the Conga Support team cannot provide assistance with the integration.

### Description

SpringCM provides secure file sharing, document management, and collaboration services. You can easily add, find, share, and manage content directly from standard or custom objects in [salesforce.com](https://salesforce.com). SpringCM is available separately from the Salesforce AppExchange. You must subscribe to that service to use this integration. Learn more about SpringCM on their [website](#).

### Parameters

- [SCMEndpoint](#) connects to a specific SpringCM endpoint.
- [SCMPreview](#) displays the SpringCM document preview window after the merge process completes.
- [SCMVisible](#) enables the SpringCM button on the Conga Composer dialog box.

## Template Fields Used in Parameters

The following are special fields provided by Conga Composer to be used as values for parameters in the Conga Composer button or link URL. They can be used for any parameter where appropriate.

{Template.Label}

Retrieves the label of the template (from the Conga Template or Salesforce Template repository record) being used in the merge process. If a local template is being used, the file name is retrieved.

&OFN={Template.Label}+for+{!Opportunity.Name}

{Template.Id}

Retrieves the Salesforce ID for the template being used in the merge process. Does not apply to local templates.

&TemplatelD={Template.Id}



{EmailTemplate.Label}

Retrieves the label of the email template (either Conga Email Template or Salesforce email template) being used in the merge process.

&EmailSubject={!Contact.FirstName}-{EmailTemplate.Label}

{EmailTemplate.Id}

Retrieves the ID for the email template (either Conga Email Template or Salesforce email template) being used in the merge process.

&OFN={EmailTemplate.Id}

## Template Group

**Parameter Category:** Template Selection Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** TemplateGroup

**Parameter Value:** [value]

**Parameter Requirements:** Templates must be stored in Conga Template Manager.

**Parameter Description:** The TemplateGroup parameter filters the list of file templates that appear on the Conga Composer dialog box to those that match this criteria. May include one or more values, separated by commas.

You can also specify multiple Template Groups by separating each with a comma.

Example: *&TemplateGroup=Invoices,Opportunities*

This example would display all templates with a Template Group name that includes the words Invoices and/or Opportunities. If you also had a Template Group called New Customer Invoices, templates in that group would appear as well since it includes the word Invoices.

Template Group value must be set on each Conga Template record.

**Example:**

&TemplateGroup=Invoices

# Templateld

**Category:** Template Selection Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** Templateld

**Parameter Value:** [Template record ID]

**Parameter Requirements:** None

**Parameter Description:** The Templateld parameter pre-selects the template specified in the Template List box.

## Important

You can include up to 20 Template Ids in Composer Business Edition and up to 40 Template Ids in Composer Enterprise Edition. Additional [product data limits](#) apply.

With multiple templates specified and the output format set to:

- Adobe PDF, the system will convert each output file to PDF, join the collection and return a single PDF.
- Same as Template, the system will return a zipped file containing each output file. To return a zipped collection of PDFs, include the parameter &ZipPDFs=1.

### Example:

```
&Templateld=a035000002abcde,a035000002ABCdE,a035000002aBc12
```

# Template Selection Parameters

## Description

You can use template selection parameters to include up to 10 merge templates to use in your Conga Composer solution.

## Parameters

- [DefaultLocal](#) specifies the default template source.
- [DefaultPane](#) sets Salesforce Content as the default option in the Select Templates view.
- [DS4](#) locks the list of template choices so that a user cannot select a different template.
- [DS8](#) displays the Local Template tab.

- [HideContent](#) hides Salesforce Content and Documents from the Select Templates view.
- [SelectTemplates](#) selects and arranges up to 10 templates.
- [TemplateID](#) preselects a template in a solution.
- [TemplateGroup](#) filters a list of templates that appear in the Conga Composer dialog box.
- [TNC](#) specifies criteria to filter the template names that appear on the Conga Composer page.

## TNC

TNC - Template Name Contains


**Category:** Template Selection Parameter

**Compatible with:** [Composer 7](#) and [Composer 8](#)

**Parameter Name:** TNC

**Parameter Value:** [value]

**Parameter Description:** Filters the list of templates that appear on the Conga Composer page to those that match these criteria.

 This parameter works with templates stored either in the Conga Template Manager or Communication Templates. This parameter works on its own or can be used in addition to *&TemplateGroup*.

## TPV

**Category:** Advanced Data Formatting Parameter


**Compatible with:** [Composer 7](#) and [Composer 8](#)

**Parameter Name:** TPV

**Parameter Value:** [0 or 1]

**Parameter Requirements:** None

**Parameter Description:** TPV (Translate Picklist Values) retrieves translated picklist field values. For organizations where picklist values have been translated via the [Translation Workbench](#), this setting enables Conga Composer to retrieve the translated values.

 This parameter does not work with SOQL queries; use `toLabel()` on the picklist field instead.

Example of SOQL query:

```
SELECT Payment_Frequency__c, Quantity, UnitPrice
FROM OpportunityLineItem
WHERE OpportunityId = '{pv0}' ORDER BY UnitPrice DESC
```

Example of the query using `toLabel()`:

```
SELECT toLabel(Payment_Frequency__c), Quantity, UnitPrice
FROM OpportunityLineItem
WHERE OpportunityId = '{pv0}' ORDER BY UnitPrice DESC
```

Value	Definition
0	Picklist values are not translated. (default)
1	Picklist values are translated to the language specified in the Personal Information settings of the user.

**Example:**

&TPV=1

## UFO

**Category:** Activity Logging Parameter

**Compatible with:** Composer 8

**Parameter Name:** UFO

**Parameter Value:** [0 or 1]

**Parameter Description:** Update Fields

Updates any Master Field to Set operations defined (either in the Conga Composer URL or the Conga Template record).

Value	Definition
0	Does not update fields. (Default)
1	Updates fields.

## WordCheckboxSize

**Category:** Word Parameter

**Compatible with:** Composer 8

**Parameter Name:** WordCheckboxSize

**Parameter Value:** [0 to 10]

**Parameter Description:** Adjusts the size of the checkboxes used in CHECKBOX: fields in Word templates. 10 is the default value so choosing another value would make the size of the checkboxes smaller.

**Example:**

```
&WordCheckboxSize=7
```

## Word Parameters

### Description

You can use the following parameters to expand the behavior of Word templates.

### Parameters

- [AWD](#) specifies whether Word documents are assembled (and if so, how).
- [RTLFont](#) specifies whether to enable Right-to-Left language formatting.
- [WordCheckboxSize](#) specifies the font size of checkboxes in Word templates.
- [WordRNP](#) removes number padding in Word output documents.
- [WordStripIfs](#) specifies whether IF statements are stripped out of fields in a Word template.
- [WPD](#) locks Word documents to prevent editing.
- [WPDPW](#) specifies the password for locked documents.
- [LRTF](#) allows rich text fields to be merged in the Lightning Experience without adding additional line breaks in resulting lists.

# WordRNP

**Category:** Word Parameter


**Compatible with:** Composer 8

**Parameter Name:** WordRNP

**Parameter Value:** [0 or 1]

**Parameter Description:** Remove Number Padding in Word

Removes the number padding added by Microsoft Word when a numeric picture switch is used in a Conga Composer Release 8 solution.

 Without this parameter, a numeric picture switch in a Conga Composer Release 8 solution using a Word template formats a small number with leading spaces. For example, a value of 30 with a picture switch of \$#,#### appears on the template as \$ 30. Using the WordRNP parameter, the same value appears as \$30. This parameter is needed only in Composer 8 solutions.

Value	Definition
0	Number padding is left as the default method in Word.
1	Number padding is removed from the output of numeric formatting.

# WordStripIfs

**Category:** Word Parameter


**Compatible with:** Composer 8

**Parameter Name:** WordStripIfs

**Parameter Value:** [0 or 1]

**Parameter Description:** Word Strip IF Statements

When enabled, the output Word document contains only the results of IF fields, not the IF fields themselves.

 This is particularly useful to hide the alternatives that might appear within an IF field, or when sending output to Google Docs.

This parameter does not support images.

Value	Definition
0	IF statements in the Word template are left as-is. (default)
1	Only the results of the IF statements in the Word template are shown.

**Example:**

&WordStriplfs=1

## WPD

**Category:** Word Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** WPD

**Parameter Value:** [0 or 1]

**Parameter Description:** Enable Word Protect Documents

The Word document output is locked to prevent editing, except for merge fields that contain the TEXTBOX: prefix. See the Microsoft Word Templates Guide for further details.

Value	Definition
0	The Word document output is editable (default).
1	The Word document output is locked.

**Example:**

&WPD=1

## WPDPW

Do not enable password protection in the Word Template while also using the WPDPW parameter. In that scenario, Conga Composer will honor the WPDPW parameter's value and override the existing password protection in the template document.

**Category:** Word Parameter

**Compatible with:** Composer 8

**Parameter Name:** WPDPW

**Parameter Value:** [text]

**Parameter Requirements:** &WPD=1

**Parameter Description:** Word Protected Document Password

Specifies the password with which a Word document is protected and locks the Word document to prevent editing.

Users must enter the specified password to enable editing in the Word document.

To enable editing in Word:

1. Click the File tab in the Word document.
2. Click the Protect Document dropdown.
3. Select Restrict Editing.
4. Click the Stop Protection button on the right-hand sidebar.
5. Enter the specified password and click OK.

For more information, see [Add or remove protection in your document, workbook, or presentation](#).

**Example:**

&WPDPW=OpenSesame

## xid

**Category:** External Data Parameters

**Compatible with:** Composer 8

**Parameter Name:** xid

**Parameter Value:** [QueryAlias:ExternalConnection:QueryId]

**Parameter Requirements:** None

**Parameter Description:** The xid parameter indicates an external data source.

- QueryAlias – Choose an alias for your query and enter it here. Required. Cannot contain spaces or special characters, and must be between 3 and 20 characters long. This QueryAlias becomes the dataset name in Template Builder and in your template.



- ExternalConnection - ExternalConnection is the alias of the data connection you set up in Step 1) Set up and Configure external data sources.
- QueryID - Salesforce Id of the Conga Query you set up in Step 2) Build your Conga Query.

 The xid parameter supports both [Conga Keys](#) and standard record IDs.

Use [pv functionality](#) to dynamically filter the results of your query. When using pv functionality with external data, you must define pv in both the Conga Query and in the xid parameter. For example, the following query syntax has pv0 defined, in addition to the related second xid parameter value in the example below: Products?

```
$format=json&$filter=UnitsInStock gt {pv0}
```

**Example:**

```
&xid=MyOracleDataAlias:MyOracleExternalConnection:a045000001DjTva,  
ExternalProduct:CongaData:a045000001DjTva?pv0={!Account.AverageStock}
```

## XMD

**Category:** Data Gathering Parameter


**Compatible with:** Composer 7 and Composer 8


**Parameter Name:** XMD

**Parameter Value:** [0 or 1]

**Parameter Description:** Exclude Master Object Data

Excludes the master object data from being included in the merge process.

 The benefit of using this parameter is that in a Word Template, you do not need to use detail region tags (e.g. {{TableStart:ReportData}}), simply place the Report Data fields wherever you like on the template, as though they were Master fields.

 Applicable only when using exactly one Report or Query as the only data source. Naming of the Report or Query in the button is not allowed and causes the data to disappear.

Value	Definition
0	The master data is included in the merge process. (default)

Value	Definition
1	The master data is excluded from the merge process.

## ZipFiles

**Category:** Output Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** ZipFiles

**Parameter Value:** 0 or 1

**Parameter Requirements:** Multiple Template IDs specified (e.g. &TemplateId=a01268Hc74Vb9Mh,a0149mJh85ht78Cxj).

**Parameter Description:** ZipFiles determines if output files are not be grouped or if output files are created as individual files during multi-template merges (not grouped into a .zip file).

 Composer's default behavior is to merge and download multiple templates into a Zip file.

Value	Definition
0	Output files will be created as individual files during multi-template merges, not grouped into a .zip file.
1	Output files will be zipped (default behavior).

**Example:**

```
&TemplateId=a01268Hc74Vb9Mh,a0149mJh85ht78Cxj
&ZipFiles=0
```

## ZipPDFs

**Category:** Output Parameter

**Compatible with:** Composer 7 and Composer 8

**Parameter Name:** ZipPDFs

**Parameter Value:** [0 or 1]

**Parameter Requirements:** Multiple Template IDs specified (e.g. &TemplateId=a01268Hc74Vb9Mh,a0149mJh85ht78Cxj) Output format set to Adobe PDF (&DefaultPDF=1 or &FP0=1).

**Parameter Description:** In a multiple-template merge, indicates that output PDF files should be zipped rather than assembled into a single (large) PDF.

Value	Definition
0	In a multiple template merge, assembles PDFs into one large PDF. (default)
1	In a multiple template merge, indicates that output PDF files should be zipped rather than assembled into a single PDF.

## Advanced Composer Solutions

Tips and Tricks, Global Reporting, and Integrations for Composer.

Select one of the following topics for more information:

- [Composer Tips and Tricks](#)
- [Composer Integrations](#)
- [Global Reporting with Conga Composer](#)
- [Conga Composer and DocuSign for Salesforce Integration](#)
- [Transition from SpringCM to the Conga Product Suite](#)

## Composer Tips and Tricks

Advanced tips and tricks for Conga Composer.

Select one of the following topics for more information:


- [Adding a File Name to a Conga Sign Transaction Related List](#)
- [Configuring the Composer Lightning Component for a Salesforce Community](#)
- [Dynamic Template Selection Using Salesforce Formula Field](#)
- [Formatting Currency Values on a Dynamic, Record-by-Record Basis](#)
- [Formula-Driven Parameters](#)
- [How Do I Take an Existing Button or Link and Convert it to a Solution Manager Solution?](#)
- [How to Call Conga Composer from Apex](#)

- [Illegal Characters for a Composer Button URL](#)
- [Create Auto-Incrementing Version Numbers](#)
- [How to Create a Conga Composer Related List View button on a Page Layout](#)
- [How to Create a Date Picker Pre-Selection Page in Composer](#)
- [How to create a Visualforce button to call Conga Composer](#)
- [How to Integrate Conga Composer in Salesforce Communities](#)
- [Live Edit Issue Workarounds](#)
- [Queries to parse data in a Conga Batch or Workflow formula](#)
- [Dynamically show content with TableStart/TableEnd and Rowburner](#)
- [Requirements for Sending Automatic Emails with Conga](#)
- [Streamlining Template Selection in Conga Composer](#)
- [Use Case for QueryStringFields](#)
- [Use Salesforce Functions \(IF, OR, ISPICKVAL\) in a Conga Composer Button URL](#)
- [Use a Salesforce Attachment or Salesforce Files as a Conga Template](#)
- [Using Composer to Update Fields in Salesforce](#)
- [Using Live Edit to a Solution](#)
- [Using Qvars in a Conga Composer Solution](#)
- [Transitioning from Courier to a Composer and Batch Solution](#)

## Adding a File Name to a Conga Sign Transaction Related List

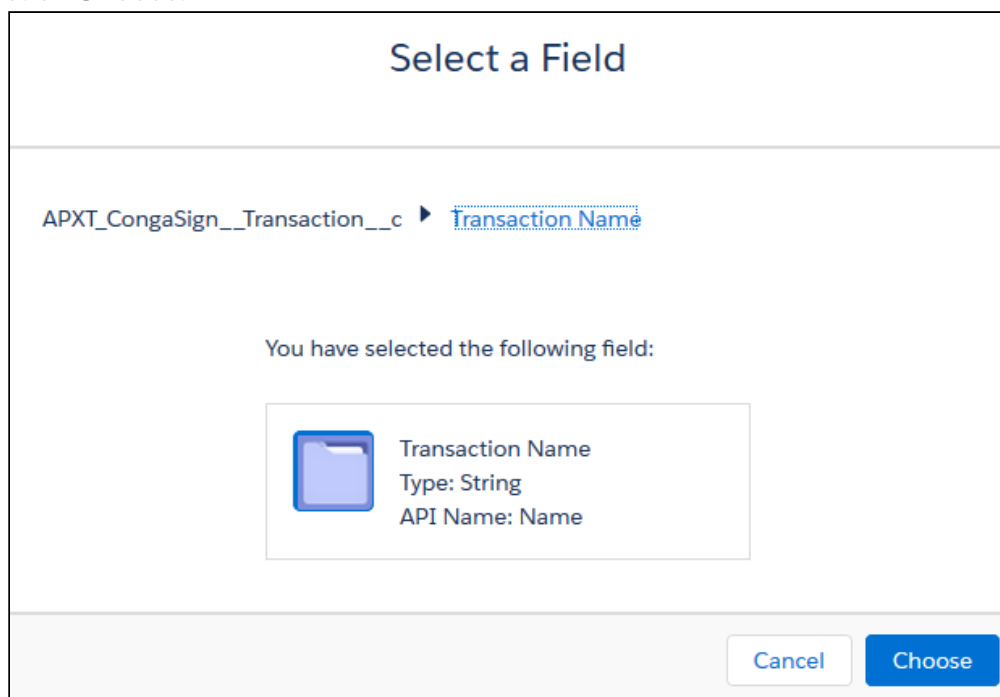
Adding the File Name to the Conga Sign Transaction Related List

You can add the file name to the Conga Sign Transaction Related List. To add the File Name to a Transaction Related List on the Opportunity object:

 **The instructions below use the Process Builder in the example. Conga does not support Process Builder.**

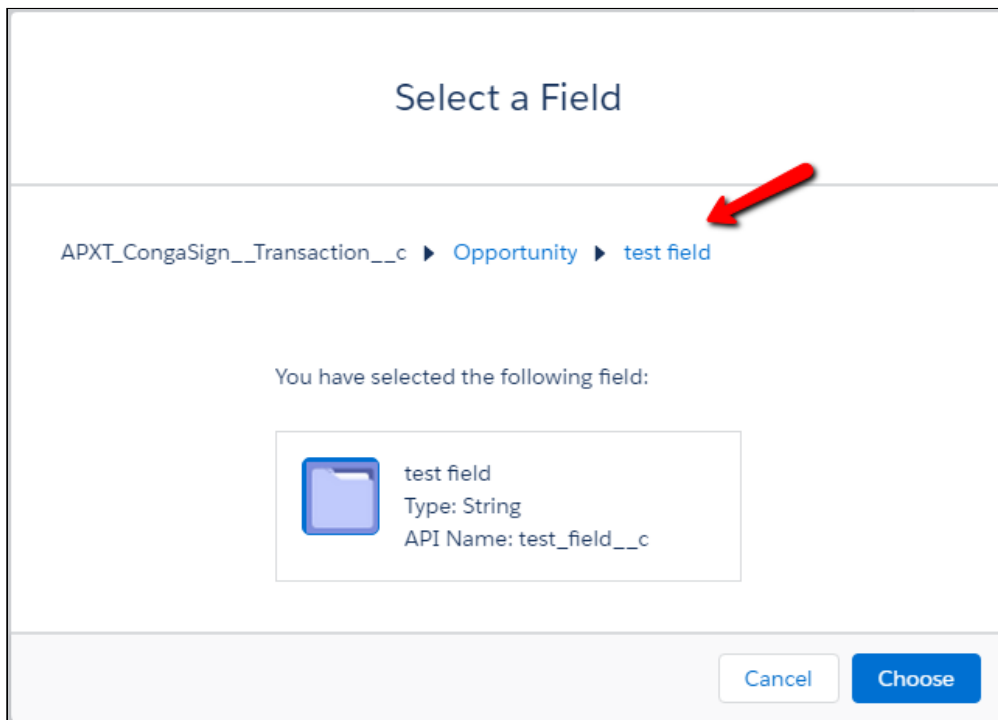
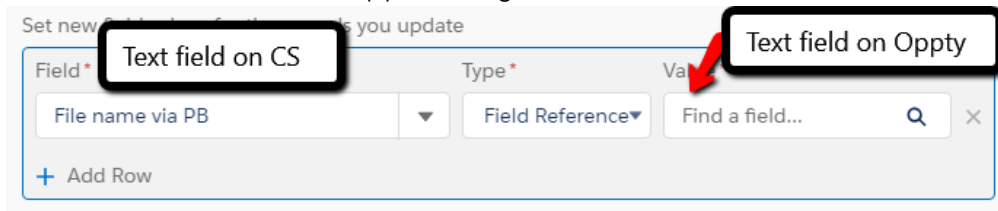
1. Create a text field on the Opportunity object, for example: `test_field__c`. (The Conga Composer MFTS parameter writes the OFN parameter to the text field when merged.)
2. Create a text field on the Conga Sign Transaction object, for example: `File_name_via_PB__c`. (The field is populated by the Process Builder.)
3. Add the `File_name_via_PB__c` field to the Conga Sign Transaction Related List on the Opportunity page layout.
4. Setup a Composer and Conga Sign button. Use the MFTS0 parameter to update the `test_field__c` in the button URL. You must use the same value in the OFN parameter and the MFTSValue0 parameter.

5. Setup a Process Builder.
  - a. Locate the Process Builder from the Quick Find search in Salesforce Setup.
  - b. Click **New**.
  - c. Add a Process Name and choose **A record changes** as the field value for the **The process starts when** field.
  - d. Click **Save**.
  - e. Click **+ Add Object** and select **Conga Sign Transaction** as the field value for the Object field.
  - f. Select **Only when a record is created** as the field value for the **Start the process field** and then click **Save**.
  - g. Click **+ Add Criteria** and fill out the Criteria Name field.
  - h. Choose **Conditions are met as the field value** for the **Criteria for Executing Actions** field.
  - i. In the Set Conditions section, click the Lookup icon to select a Field's lookup icon to select a field. Click the dropdown arrow, select **Transaction Name**, and then click **Choose**.



- j. Change the Operator field value to **Is Null** and change the Value field to **False**. Keep the Type as Boolean.
- k. Set the Conditions field to **All of the conditions are met (AND)** and then click **Save**.
- l. Click **+ Add Criteria** under Immediate Actions and add a Criteria Name.
- m. Click **+ Add Action** and choose **Update Records** as the Set Action Type to field value.

- n. Provide an Action Name and choose **Select the APXT\_CongaSign\_\_Transaction\_\_c record that started your process** as the Record Type field value. Click **Choose** and then click **Save**.
- o. Select the Criteria for Updating Records or select **No criteria—just update the records!** if no further criteria are needed.
- p. Set the values for the correct field to update. Choose the new custom text field on the Conga Sign Transaction object (in our example this is the File name via PB field). Change the Type field's value to **Field Reference**. Select the new custom text field on the related object using the object lookups. (in our example, this is the test\_field\_\_c on the Opportunity).



- q. Click **Save**.
  - r. Click **Activate**.
  - s. Click **Confirm**.
6. Launch the Composer button and go through the steps to create a Conga Sign Transaction.

# Configuring the Composer Lightning Component for a Salesforce Community

Use the Composer Lightning Component in Salesforce Communities to allow Community users to easily merge documents. Follow the steps below to configure the Composer Lightning Component for use in a Salesforce Community.


- ⚠ • The Composer Lightning Component does not integrate with Community home pages. The Composer Lightning Component is intended for use on Record Detail pages in Communities. Additionally, the Composer Lightning Component is not intended for High-Volume Community Users, as opposed to standard Community users.
- Composer does not support proxy URLs in Salesforce Communities/Experience Cloud.
- Users cannot successfully launch the Composer Lightning Component in Communities when accessing it from Salesforce Mobile.

## Set Up the Composer Lightning Component for a Customer Community


### Prerequisites to Setup the Composer Lightning Component for a Customer Community

- Create and activate Remote Site Settings for the Customer Community.
  - a. Navigate to Salesforce Setup and use the Quick Find search box to access Remote Site Settings under the Security section.
  - b. Click **New Remote Site**.
  - c. Name the new Remote Site and paste in the Customer Community domain URL in the Remote Site URL field.
    - Example Remote Site URL for a Customer Community: *https://granitehorse-16c4551c140.force.com*
  - d. Click **Save**.
- Clone a standard Customer Community User profile, such as the Customer Community User Profile, to create a new a custom Community user profile.
  - a. In the Profile section of Salesforce Setup, click **Clone** next to the standard Customer Community User profile.
  - b. Name the name the new custom profile and click **Save**.
- Enable the API Enabled permission for the new custom Community profile.
  - Navigate to the new custom Community profile in Salesforce Setup.

- Click **System Permissions**.
- Click **Edit**.
- Check the Enabled checkbox field next to API Enabled and click **Save**.
- Assign the new custom Community profile to Community users that will use the Composer Lightning Component.

 To assign a custom or standard Community profile to a Community user, the Community user must have initially been created from a Contact record. For more information on creating a Community user, see [Create Communities Users](#).

- Assign Conga Composer licenses to Community users that will use the Composer Lightning Component.
- Assign the Composer Communities Permission Set to Community users that will use the Composer Lightning Component.
  - As an alternative to assigning the Composer Communities Permission Set, ensure that the custom Community user profile has [necessary profile permissions to use Conga Composer](#).

 Community Users do not need access to the Modify Metadata Through Metadata API Functions permission to use the Composer Lightning Component in Communities. In this use case, Composer saves the refresh token into the browser cookie instead of accessing custom metadata.

- (optional) Check the *Allow internal users to log in directly to the community* checkbox field in the Administration section of the Workspace of a Community. This allows internal users to log into a Community and use the Composer Lightning Component.
  - Navigate to Salesforce Setup and use the Quick Find search box to select **All Communities**.
  - Click **Administration**.
  - Click **Login & Registration**.
  - Check the *Allow internal users to log in directly to the community* checkbox and click **Save**.

### Set Up the Composer Lightning Component for a Customer Community

It is recommended to use the [background mode parameter \(DS7\)](#) with the Composer Lightning Component in a Community to provide the best user experience.

1. Navigate to Salesforce Setup and use the Quick Find search box to access **All Communities** under the Communities section.
2. Click **Builder** next to the Customer Community to access the Community Builder.




3. Use the dropdown menu in the top left corner to navigate to a Record Detail page.
4. Click the **Components** option on the left hand side.
5. Scroll down to the Custom Components section and drag the Conga Composer Lightning Component onto the Record Detail page.
6. Select a Solution Collection to use for the Composer Lightning Component.
7. Click **X** after selecting the Solution Collection.
8. Click **Publish**.

## Set Up the Composer Lightning Component for a Partner Community

### Prerequisites to Set Up the Composer Lightning Component for a Partner Community


- Create and activate Remote Site Settings for the Partner Community.
  - a. Navigate to Salesforce Setup and use the Quick Find search box to access **Remote Site Settings** under the Security section.
  - b. Click **New Remote Site**.
  - c. Name the new Remote Site and paste it in the Partner Community domain URL in the Remote Site URL field.
    - Example Remote Site URL for a Partner Community: *https://granitehorse-16c4551c140.force.com/*
  - d. Click **Save**.
- Clone a standard Partner User profile, such as the Partner User Profile, to create a new custom Partner User profile.
  - a. In the Profile section of Salesforce Setup, click **Clone** next to the standard Partner User profile.
  - b. Name the name custom profile and click **Save**.
- Enable the API Enabled permission for the new custom Partner User profile.
  - a. Navigate to the new custom Partner User profile in Salesforce Setup.
  - b. Click **System Permissions**.
  - c. Click **Edit**.
  - d. Check the Enabled checkbox field next to API Enabled and click **Save**.
- Assign the new custom Partner User profile to Partner Users that will use the Composer Lightning Component.

 To assign a custom or standard Partner Users profile to a Partner User, the Partner User must have initially been created from a Contact record. For more information on creating a Partner User, see [Create Partner Users](#).

- Assign Conga Composer licenses to Partner Users that will use the Composer Lightning Component.

- Assign the Composer Communities Permission Set to Partner Users that will use the Composer Lightning Component.
  - As an alternative to assigning the Composer Communities Permission Set, ensure that the custom Partner User profile has the [necessary profile permissions to use Conga Composer](#).
- Enable Public/Read Default External Access in Sharing Settings.
  - a. Navigate to Salesforce Setup and use the Quick Find search box to access Sharing Settings under the Security section.
  - b. Click **Edit**.
  - c. Change the Default External Access column's value to Public Read/Write for all Conga objects.
  - d. Click **Save**.
- (Optional) Check the *Allow internal users to log in directly to the community* checkbox field in the Administration section of the Workspace. This allows internal users to log into a Community and use the Composer Lightning Component.
  - a. Navigate to Salesforce Setup and use the Quick Find search box to select **All Communities**.
  - b. Click **Administration**.
  - c. Click **Login & Registration**.
  - d. Check the *Allow internal users to log in directly to the community* checkbox and click **Save**.

## Setup the Composer Lightning Component for a Partner Community

 It is recommended to use the [background mode parameter \(DS7\)](#) with the Composer Lightning Component in a Community to provide the best user experience.

1. Navigate to Salesforce Setup and use the Quick Find search box to access All Communities under the Communities section.
2. Click **Builder** next to the Partner Community to access the Community Builder.
3. Use the dropdown menu in the top left corner to navigate to a Record Detail page.
4. Click the **Components** option on the left hand side.
5. Scroll down to the Custom Components section and drag the Conga Composer Lightning Component onto the Record Detail page.
6. Select a Solution Collection to use for the Composer Lightning Component.
7. Click **X** after selecting the Solution Collection.
8. Click **Publish**.

## Frequently Asked Questions

### Does the Composer Lightning Component work in a Community when it is integrated with Conga Sign?

*Yes - Instances where the Composer Lightning Component leverages Conga Sign will run successfully in a Salesforce Community.*

### As an Admin, why can I see the Composer Lightning Component in a Community but a Community user cannot?

*It is likely that the Community users do not have the required Composer-related permissions to view the Composer Lightning Component. Ensure that the Community users have the necessary access to Conga object and Apex classes. Additionally, check to see that org-wide sharing settings are set up appropriately.*

### How does licensing work with Community users launching the Composer Lightning Component in a Community

*Each Community user must have an assigned Conga Composer license. For more information on assigning Conga Composer licenses, see [Assign User Licenses](#).*

### Can Community users launch the Composer Lightning Component in a Community while accessing it from Salesforce Mobile?

*No, the Composer Lightning Component is not compatible in Communities when attempting to launch it from Salesforce Mobile.*

## Dynamic Template Selection Using Salesforce Formula Field

You can use a Salesforce formula field (in this example a CASE function) to determine which template is used in the merge process.

The CASE function will evaluate some field - for example, Opportunity Stage - and based on the value of that field will return the appropriate template ID. This template ID is then passed to the `Templateld` parameter in the Conga Composer button/link URL to select the appropriate template. Start by building a formula field (text output) on your master object. Use the Advanced Formula tab to insert the CASE function. A Case statement is an extended If statement where the expression indicates what field you want to evaluate, the value is a potential value of that field, and the result is the result you want when that field equals the value.

For example: `CASE( expression, value1, result1, value2, result2, ...,else_ result)`

Modify your Case statement to evaluate the desired field and return the appropriate results. In the example below, we evaluate the Opportunity Stage field and return various template IDs accordingly.

After the last value and correlating result, we've entered " " as our else result which populates if none of the other conditions are true.

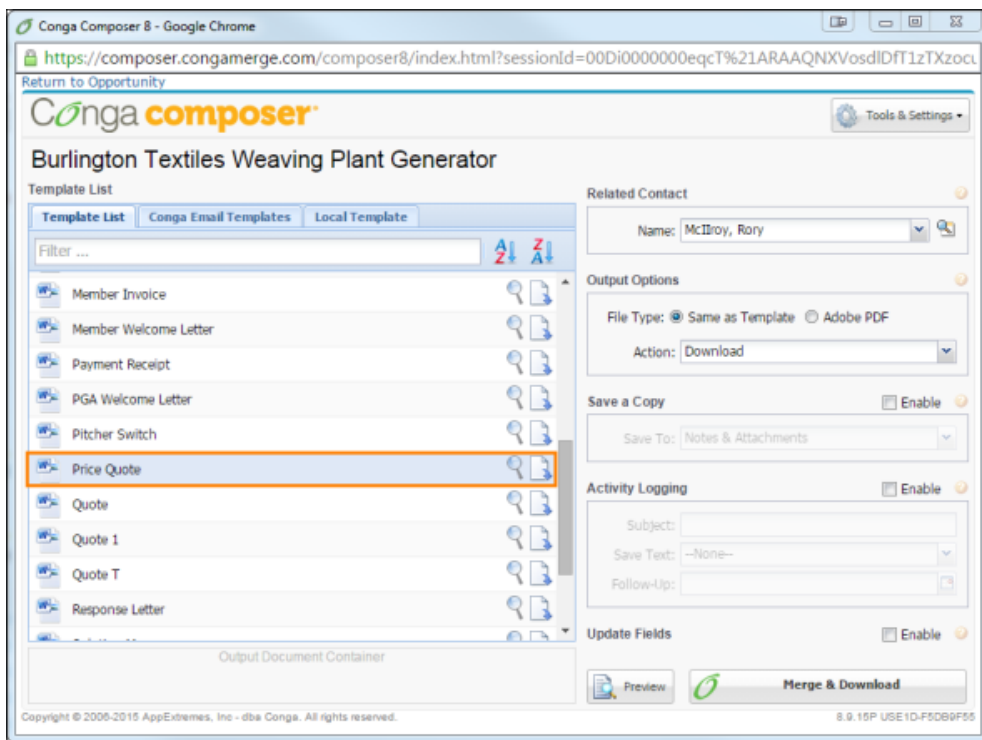
For example: `CASE(StageName, "Prospecting", "a00000000CiPuS", "Value Proposition", "a008000000CZNDh", "Proposal/Price Quote", "a008000000DiYKJ", "Negotiation/Review", "a08000000E1uXu", " ")`

This formula field can be added to the page layout or not; either way, it will always be evaluated and populated, as long as the field-level security is set to "visible" for any user profiles using Conga. Although for template development purposes it may be handy to have it on the page layout to begin with. Also, if you want users to be able to see the else result, you'll want the field added to the page layout. Finally, users on Professional Edition of Salesforce will have to add the field to the page layout, as field-level security is not an option.

Add this custom formula field into your Conga Composer button URL as part of the TemplateId parameter.



Conga Composer now looks to the value of your Case statement formula field to see which template ID is designated and then pre-selects that template in the Conga Composer dialog box.



## Formatting Currency Values on a Dynamic, Record-by-Record Basis

Format currency values based on a field on a record. Let's say we're creating a quote from an Opportunity and we'd like format currency values based on a field in the Opportunity. Here's how:

1. Create a picklist field on the Opportunity to contain the culture choices. Let's call the field Culture. In this example, let's use the following choices:

Picklist values	
Action	Values
<a href="#">Edit</a>   <a href="#">Del</a>	en-US
<a href="#">Edit</a>   <a href="#">Del</a>	en-GB
<a href="#">Edit</a>   <a href="#">Del</a>	de-DE

2. Insert the CurrencyCulture parameter into the Conga Composer URL:  
`&CurrencyCulture={!Opportunity.Culture}`
3. Format the currency fields in your Word template using the `\# currency` picture switch: `{ MERGEFIELD Opportunity_LineItem_SalesPrice \# Currency }`

These settings would result in output such as:

<b>Culture</b>   de-DE	<table border="1"> <thead> <tr> <th>Unit Price</th> <th>Total Price</th> </tr> </thead> <tbody> <tr> <td>360,00 €</td> <td>1.800,00 €</td> </tr> </tbody> </table>	Unit Price	Total Price	360,00 €	1.800,00 €
Unit Price	Total Price				
360,00 €	1.800,00 €				
<b>Culture</b>   en-GB	<table border="1"> <thead> <tr> <th>Unit Price</th> <th>Total Price</th> </tr> </thead> <tbody> <tr> <td>£360.00</td> <td>£1,800.00</td> </tr> </tbody> </table>	Unit Price	Total Price	£360.00	£1,800.00
Unit Price	Total Price				
£360.00	£1,800.00				

Disable Wrap Text in cells that use Currency formatting to keep the value and the symbol on the same line: **Table** → **Table Properties** → **Cell** → **Options** → **Uncheck Wrap Text**.

Also view: [Dynamic template selection using Salesforce formula field](#)

## Formula-Driven Parameters

Rather than defining a parameter with a particular value, you can use Salesforce formulas to provide a limited amount of dynamic flexibility to your Conga Composer URLs. Say you'd like users of a particular profile to always generate PDFs from Conga Composer, whereas users of other profiles may choose whether to use the native file format or PDFs. You could accomplish this by using a formula to control the behavior of the FP0 parameter.

In this example, after the base URL, we've enabled the Default PDF parameter and made the Force PDF parameter (fp0) dynamic, based on the user's Profile. For System Administrators, fp0 is disabled. For others, fp0 is enabled.

```
https://composer.congamerger.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Contact.Id}
&DefaultPDF=1
&FP0={!IF( $Profile.Name = "System Administrator", "0", "1")}
```

In practice, we would recommend using an ID, such as `$Profile.Id`, rather than a text string, such as the Profile Name. You can learn more about the Salesforce formula language by searching for Operators and Functions in the [Salesforce.com](#) online help.

# How Do I Take an Existing Button or Link and Convert it to a Solution Manager Solution?

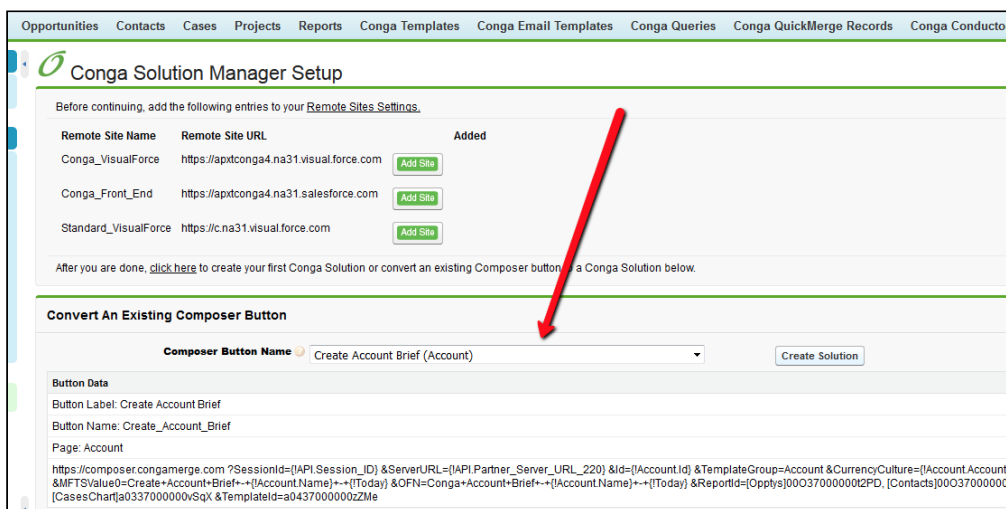
Convert an existing Composer button or link to a Conga Solution record to be managed with Solution Manager.

**⚠ This feature is not supported in Conga Composer version 8.10 and above. You must have Conga Solution Manager installed in your Salesforce Org. To get Conga Solution Manager, upgrade to the latest release of Conga Composer.**

You'll need to create a custom setting entry and enable it. To do so, follow the steps below:

1. Go to **Setup** and search for **Custom Settings** under Develop.
2. Click **Manage** next to Conga Solutions Settings and then click **New**.
3. Call it **PROD**, check the box and leave the endpoint URL alone.
4. Save that and navigate to the **Conga Solutions Setup** tab.


There will now be a new page below the Remote Sites that allows you to pick buttons and create solutions (see image below).



When you find the Conga buttons, just click Create Solution and Update the button. Now the button is part of a Conga Solutions record and can be managed with Solution Manager.


To work properly, the button you are converting must be a valid solution with good ids, etc. Also, it cannot be a Composer Release 7 button (i.e. one that points to appextremes in the URL). Lastly, any queries created in the Release 7 Query Manager must be copied to the newer Query Manager (the one that is part of the Composer Release 8 package).

## How to Call Conga Composer from Apex

 Due to the complicated nature of supporting programmers and programming languages, the integration of Conga with Visualforce, Apex, Javascript (or other programming languages), and custom code is not supported.

Calling Conga Composer from Apex:

1. Create a Remote Site setting record for "https://composer.congamerge.com"
2. In your code, make an HTTP GET call with a long timeout (~60 seconds).
3. Your code should call https://composer.congamerge.com/composer8/index.html
4. Include the typical querystring parameters in the request
5. Include an APIMode parameter in the Conga Composer endpoint URL:

 If present, remove the [DS7 \(background mode\) parameter](#) from the Composer Endpoint URL. DS7 may interfere with the APIMode parameter.

APIMode Value	Action in Conga Composer
1	Attaches the document to the master object
2	Reserved
3	Delivers the document via Conga Sign
4	Reserved
5	Attaches the document to the Documents Tab (requires &DocFolderId)
6	Attaches the document to Salesforce Content (requires &ContentWorkspaceld)
12	Emails the document according to the Email parameters
13	Delivers the document via Adobe eSign
17	DocuSign



When successful, the API Mode parameter results in a Salesforce Record ID or an error message.

**⚠** Calling Conga Composer in a programmatic loop, whether from Apex or any other programming language, constitutes a violation of the Master Services Agreement and is subject to immediate suspension of your account. This includes the use of script, code, program, or alike to mimic the functionality of Conga products or services.

## Illegal Characters for a Composer Button URL

The following characters cannot be used as part of the parameter values in a Composer button URL. Using any of these characters will cause your solution to fail.

Character Name	Character
Colon	:
Hash	#
Open Parenthesis	(
Close Parenthesis	)
Double Quotes	"
Percentage	%
Backslash	\

## URL Encoding Illegal Characters

The above illegal characters can be encoded using URL Encoding. This will allow you to use them in your parameter values in your Composer URL. Below are their encoded values:

Character Name	Character	Encoded Value
Colon	:	%3A
Hash	#	%23


Character Name	Character	Encoded Value
Open Parenthesis	(	%28
Close Parenthesis	)	%29
Double Quotes	"	%22
Percentage	%	%25
Backslash	\	%5C

## Using the SUBSTITUTE Function in Salesforce

If you are referencing a Salesforce field as a parameter value in your Composer URL, and this field may contain ampersands, you can create a Salesforce formula field and use the SUBSTITUTE Function to replace Ampersands within the field value. You would then use the formula field in the parameter instead. See this article for details: [Replace Ampersands in Reference Field Data](#)

## Create Auto-Incrementing Version Numbers

How to setup auto-incrementing version numbers for documents generated by Conga Composer.

 This is an advanced feature that is not supported by Conga Technical Customer Success.

You can setup auto-incrementing version numbers for documents generated by Conga Composer. This is especially useful for documents like quotes or contracts where it's common to create a few iterations before it's approved or finalized. This technique is record-specific in Salesforce, so if you're working from the same Opportunity record and generating multiple versions of the same quote over time, Conga Composer uses some simple formulas and field updates to keep track of each version.

This requires a good understanding of the Activity Logging Parameters, which allow you to update the value of a field upon merging.

To update the values of fields upon merging:

**Step 1: Create two new custom fields on your preferred object:**

- Field #1
  - Data Type: Number
  - Name: Version #
  - Length: 3 digits
  - Decimal Places: 0
  - Default Value: 1

The default value will apply to any new records, but this field will need to be assigned a value of 1 for any existing records.

Opportunity Custom Field			
<b>Version #</b>			
<a href="#">Back to Opportunity Fields</a>			
<a href="#">Validation Rules (0)</a>			
<b>Custom Field Definition Detail</b>			
<a href="#">Edit</a> <a href="#">Set Field-Level Security</a> <a href="#">View Field Accessibility</a>			
<b>Field Information</b>			
Field Label	Version #	Object Name	<a href="#">Opportunity</a>
Field Name	Version	Data Type	Number
API Name	Version__c		
Description			
Help Text			
Created By	<a href="#">Robert Boyd</a> , 10/9/2013 6:30 PM	Modified By	<a href="#">Robert Boyd</a> , 10/9/2013 6:30 PM
<b>General Options</b>			
Required	<input type="checkbox"/>		
Unique	<input type="checkbox"/>		
External ID	<input type="checkbox"/>		
Default Value	1		
<b>Number Options</b>			
Length	3		
Decimal Places	0		

- Field #2
  - Data Type: Formula - Number
  - Name: Next Version #
  - Decimal Places: 0
  - Formula Syntax: Version\_\_c + 1

This field does not need to be added to the page layout if you're using Enterprise Edition or above, but Professional Edition users will need to park it somewhere on the layout for the formula to evaluate.

Opportunity Custom Field			
<b>Next Version #</b>			
<a href="#">Back to Opportunity Fields</a>			
<b>Custom Field Definition Detail</b>			
<a href="#">Edit</a> <a href="#">Set Field-Level Security</a> <a href="#">View Field Accessibility</a>			
<b>Field Information</b>			
Field Label	Next Version #	Object Name	<a href="#">Opportunity</a>
Field Name	Next_Version		
API Name	Next_Version__c		
Description			
Help Text			
Created By	<a href="#">Robert Boyd</a> , 10/9/2013 6:33 PM	Modified By	<a href="#">Robert Boyd</a> , 10/9/2013 6:33 PM
<b>Formula Options</b>			
Data Type	Formula		
Decimal Places	0		
Version__c + 1			

## Step 2: Add MFTS parameters to your Composer button

This is where the magic happens. There are 4 required button parameters to make this work:

- &MFTS0 – References the API name of the field to update
- &MFTSValue0 – Specifies the value to be pushed into the field referenced by MFTS0
- &UF0 – Turns on the Composer Update Fields feature.

Here is an example of the button URL. This example uses an Opportunity record:

```
https://composer.congamerger.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&MFTS0=Version__c
&MFTSValue0={!Opportunity.Next_Version__c}
&UF0=1
```

The MFTS0 parameter's field value, `Version__c`, is the API field name. The MFTSValue0 parameter's field value, `{!Opportunity.Next_Version__c}`, is a merge field. The MFTS0 parameter specifies the API name of your version number that is auto-incremented when you merge and log activities. The value that is entered into that field is supplied by MFTSValue0, which is retrieving the merge field of the next version number.

### Step 3: Decide how to use it

Now that the mechanics are in place to increment version numbers as you merge documents, there are a few ways you can use this.

The OFN parameter allows you to specify a file name for your output, so what better place to put a version number? You can do this like so: `&OFN=Quote+Version+{!Opportunity.Version__c}`

If this were the second iteration of a quote in Word format, the resulting file name would be "Quote Version 2.docx"

Now that you've got a field that displays the current version number, you can add it to your template just like any other merge field. Stuff it in the footer of a contract, append the version number to the title of your proposal or add it to your quote as a watermark -tons of options here. If you are creating a completed task as part of your activity logging process, putting the version number in the subject is a great way to keep track of multiple documents over time. You can do this with the AC1 parameter like so: `&AC1=Quote+Version+{!Opportunity.Version__c}+created+on+{!TODAY() }`


If this were the second iteration of a quote, the completed activity would look like this:

Activity History					
Action	Subject	Name	Task	Due Date	Assigned To
<a href="#">Edit</a>   <a href="#">Del</a>	Quote Version 1 created on 10/10/2013	Kevin Guthrie	✓	10/10/2013	Robert Boyd
<a href="#">Edit</a>   <a href="#">Del</a>	Email: Your Invoice from Demo Studios		✓	9/14/2013	Robert Boyd
<a href="#">Edit</a>   <a href="#">Del</a>	Email: URGENT: Change Order - SP Salesworks		✓	9/13/2013	Robert Boyd

Auto-incrementing version numbers can be used in a variety of ways depending on your requirements. It also works great for Conga Batch and Conga Trigger for batch and event-triggered merges.

## How to Create a Conga Composer Related List View button on a Page Layout

A List View button that allows an end-user to dynamically select the child records they want to include in a Conga Composer operation using checkboxes.

 Javascript is not supported in Salesforce lightning. Using a Javascript button is not currently a feature available to use in the Lightning Experience. Conga cannot support customized code. Due to the complicated nature of supporting programmers and programming languages, the integration of Conga with Visualforce, Apex, and Javascript (or other programming languages) is not supported.

You can build a List View button that allows the end-user to dynamically select the child records they would like to include in a Conga Composer operation using checkboxes.

To create a list button on the child object:

1. Change your button settings to Display Type- List Button. Display Checkboxes (for Multi-Record Selection) should be checked.
  - Behavior - Execute JavaScript
  - Content Source - OnClick JavaScript
2. Copy the below text and paste it into your button. Be sure to update the Object Type in the first line to the Child Object that is the same Object as the Object your list button is on. Also, change the field in the seventh line to the Parent ID of the Object your button is on.


```
var ids= {!GETRECORDIDS( $ObjectType.OpportunityLineItem )};var
idsReport=ids.join("|");var
idsQuery=ids.join("'|");https://composer.congamerge.com" +"?sessionId={!
$Api.Session_ID}" +"&ServerUrl=
{!$Api.Partner_Server_URL_80}" +"&Id={!Opportunity.Id}" + "&ReportId=000E0000000P57r?
pv0=" + idsReport
+"&QueryId=a030000000JuspY?pv0='" + idsQuery + "''";window.open( CongaURL, "Conga",
```

```
"width=700,height=550,menubar=0" );
```

3. Create a query with all of the child fields you want to include. The WHERE clause should be the Child ID updated to `IN ({{pv0}})`
  - Example Query for Opportunity Product: `SELECT Description, ListPrice, Product_Family__c, Quantity, UnitPrice, TotalPriceFROM OpportunityLineItemWHERE Id IN ({{pv0}})`
4. Add button to layout.
5. Edit the Parent Object Page Layout. In edit mode go to your child related list and click Properties (the tool icon).
6. Open the button options at the bottom of the pop-up, add the button and save.

## How to Create a Date Picker Pre-Selection Page in Composer

This technique allows you to create a Visualforce page that prompts a user for 'Start Date' and 'End Date' values that can be passed to corresponding filters on a Salesforce Report or Conga Query.

 Due to their complicated nature, the integration of Conga with JavaScript, Visualforce, and Apex (or other programming languages) is not supported. This solution does not work in Salesforce Lightning due to the usage of Javascript.

Your Date field in Salesforce MUST be a Date Field, NOT a Date/Time Field.

1. Create a Visualforce Page called "DatePicker".
  - Navigate to **Setup** → **Build** → **Develop** → **Visualforce Pages**
  - Click **New**.
  - Name your page "DatePicker".
2. Paste the Visualforce code below into the page. The default start date is 1 week before today's date and the default end date is 1 week after today's date. These can be altered on line 24 (default start date) and line 32 (default end date), if necessary.
3. Create a standard Conga Composer button that you want to use with the Datepicker.
4. Replace the first line of the Conga Composer button URL with `/apex/Datepicker` .
  - Example Button Syntax on Account Object:
 

```
/apex/Datepicker
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_80}
&id={!Account.Id}
```

5. Create a Salesforce Report or Conga Query to filter on a dynamic date range.
  - Your report or query will need at least two filters to provide a date range. For a standard range, your filter operators will be greater or equal to your start date and less or equal to your end date. For example, a Salesforce Report with the start date as the 'pv1' filter and the end date as the 'pv2' filter:
6. Pass your start and end dates to your report or query.
  - For Salesforce Reports, you'll pass these variables:
    - Start Date: {StartDateR}
    - End Date: {EndDateR}
  - For Conga Queries, you'll pass these variables:
    - Start Date: {StartDateQ}
    - End Date: {EndDateQ}
  - Report-based example:
    - /apex/DatePicker  
 ?sessionId={!API.Session\_ID}  
 &serverUrl={!API.Partner\_Server\_URL\_80}  
 &id={!Account.Id}  
 &ReportId=[DynamicOpps]000E0000002T4H8?pv0={!Account.Id}~pv1={StartDateR}~pv2={EndDateR}
  - Query-based example:
    - /apex/DatePicker  
 ?sessionId={!API.Session\_ID}  
 &serverUrl={!API.Partner\_Server\_URL\_80}  
 &id={!Account.Id}  
 &QueryID=[DynamicOpps]a04E0000004SmqG?pv0={!Account.Id}~pv1={StartDateQ}~pv2={EndDateQ}

When a user clicks this button, the Visualforce page is shown. The user can then select the start and end dates before clicking "Launch". The date values are then substituted into the URL and the browser is re-directed to the Composer dialog.

This article provides instructions for Conga Composer Release 8. If you wish to run Composer Release 7, in the below Visualforce page, you'll need to change the fifth line from the bottom to:

```
window.parent.location.href = "https://www.appextremes.com/apps/Conga/PM.aspx?" + qs;
```

### Visualforce Page

```
<apex:page language="en" showHeader="false" sidebar="false" standardStylesheets="true"
><link type="text/css" href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.6/themes/cupertino/jquery-ui.css" rel="stylesheet" /><script type="text/javascript" src="https://
```

```

ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.js"></script><script type="text/
javascript" src="https://jquery-ui.googlecode.com/svn/trunk/ui/i18n/jquery.ui.datepicker-
en-GB.js"></script><script type="text/javascript" src="https://ajax.googleapis.com/ajax/
libs/jqueryui/1.8.16/jquery-ui.js"></script><apex:pageBlock title="Please select the start
and end dates:"><p>Start Date:<br /><input id="startDate" type="text"/></p><p>End
Date:<br /><input id="endDate" type="text"/></p><!--$(selector).datepicker({defaultDate:
new Date(2009, 1-1, 26)});$(selector).datepicker({defaultDate: '01/26/2009'});$
(selector).datepicker({defaultDate: +7});$(selector).datepicker({defaultDate: '+1m
-1d'});--><script type="text/javascript">$(function() {$("#startDate").datepicker(); // You
must leave this line$("#startDate").datepicker($.datepicker.regional['en-US']);$
("#startDate").datepicker('setDate', '-1w'); // This is optional});</script><script
type="text/javascript">$(function() {$("#endDate").datepicker(); // You must leave this
line$("#endDate").datepicker($.datepicker.regional['en-US']);$
("#endDate").datepicker('setDate', '+1w'); // This is optional});</script><br /><input
type="button" onclick="launchConga()" value="Launch" /><script>function launchConga(){var
startDate = document.getElementById("startDate");var endDate =
document.getElementById("endDate");// For Queries...var sDate = new
Date(startDate.value);var eDate = new Date(endDate.value);var startDateForQueries =
sDate.getFullYear() + "-" + ("0" + (sDate.getMonth() + 1)).slice(-2) + "-" + ("0" +
sDate.getDate()).slice(-2);var endDateForQueries = eDate.getFullYear() + "-" + ("0" +
(eDate.getMonth() + 1)).slice(-2) + "-" + ("0" + eDate.getDate()).slice(-2);var qsIndex =
window.parent.location.href.indexOf('?') + 1;var qs =
window.parent.location.href.substring(qsIndex, window.parent.location.href.length);qs =
unescape(qs);// Use these lines if you're passing to Reports:qs = qs.replace(/{\StartDateR}/
g , startDate.value);qs = qs.replace(/{\EndDateR}/g , endDate.value);// Use these lines if
you're passing to Queries:qs = qs.replace(/{\StartDateQ}/g, startDateForQueries );qs =
qs.replace(/{\EndDateQ}/g, endDateForQueries );window.parent.location.href = "https://
composer.congamerge.com?" + qs;}</script></apex:pageBlock></apex:page>

```

### Sample Button (Account)

```

/apex/DatePicker
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_80}
&id={!Account.Id}
&ReportId=[DynamicOpps]000A00000044TEO?pv0={StartDateR}~pv1={EndDateR}

```



# How to create a Visualforce button to call Conga Composer

To create a Visualforce command button to call Conga Composer, you'll need to access the current Session ID and Partner Server URL endpoint. We've seen at least two techniques to accomplish this.

⚠ Due to their complicated nature, the integration of Conga with JavaScript, Visualforce, and Apex (or other programming languages) is not supported. JavaScript buttons will not work in Salesforce Lightning. We recommend you choose a different implementation technique or engage [Conga Services](#) to build a custom solution.

To create a Visualforce command button to call Conga Composer, you'll need to access the current Session ID and Partner Server URL endpoint. We've seen at least two techniques to accomplish this.

## Technique #1

Alternatively, if you absolutely must access Salesforce Reports, you would need a SessionId and ServerUrl from a "front-end" server, and then pass those values into the Visualforce page. This is easy to do if you happen to be calling your Visualforce page from a button since you can pass the SessionId and ServerUrl as querystring parameters, like this:

```
/apex/MyPage
?SessionId={!$Api.Session_ID}
&ServerUrl={!$Api.Partner_Server_URL_130}
```

## Technique #2

Another technique to create a simple Conga Composer button on a Visualforce page would be as follows. We particularly like this technique because you can create a normal Conga Composer button on an object, and simply "point" to the standard button on the Visualforce page.

```
<apex:commandButton value="Create PDF Quote"
action="{!URLFOR($Action.Opportunity.View,Opportunity.Id)}"
onclick="openConga()"/>
<Script Language="JavaScript">
function openConga() { window.open('{!
URLFOR($Action.Opportunity.Do_The_Conga,Opportunity.Id)}',
```

```
'','scrollbars=yes,menubar=no,height=600,width=800,resizable=yes,toolbar=no,location=no,status=yes'); }  
</Script>
```

Do\_The\_Conga is referring to the API name of the standard button which you are trying to click via the Visualforce button.

## How to Integrate Conga Composer in Salesforce Communities

You can set up Conga Composer to integrate with Salesforce Communities.

**⚠ The following instructions are for your Salesforce System Administrator and are not part of the Conga support package.**

You can set up Conga Composer to integrate with Salesforce Communities. Follow the steps below to integrate Conga Composer with Salesforce Communities.

- ⚠**
- Conga Composer is not compatible with the Guest User profile in Salesforce Communities.
  - Composer does not support proxy URLs in Salesforce Communities/Experience Cloud.

Set up a [Salesforce Community](#) (basic setup: branding, color scheme, header, footer, email templates, chatter branding, login and logout URL, notification templates, self-registration settings, Lightning navigation customize tabs, etc).

1. Activate Share Group settings (optional).
2. Set up Partner Roles per Partner Account (Partner Portal only).
3. Create Sharing Sets.
4. Create Sharing Settings (High Volume Community and Portal Users only).
5. Clone a Profile (for either Customer or Partner Profile) to create a new custom profile and enable the following:
  - Connected App Access to Conga Composer
  - API Enabled permission in System Permissions
  - Custom Objects Permissions and Layouts
    - The new profile requires appropriate access and permissions to Conga objects.
  - A custom profile is required as standard Community and Partner profiles cannot leverage APIs.

6. Add the custom profile(s) to the Community Members selected profile list.
7. Enable Contact records as [External User Contacts](#).
8. For Communities accessed by external users: modify the initial Composer URL to reflect the Community Domain as shown in the example below:
  - From: /apex/APXTConga4\_\_Conga\_Composer
  - To: https://bluecubecommunity.force.com/...Conga\_Composer
9. For Communities that are accessed by internal users: use the non-Oauth Composer URL that includes a session ID.
  - `https://composer.congamerge.com`  
`?sessionid={!$Api.Session_ID}`  
`&serverUrl={!API.Partner_Server_URL_370}`
10. Add the ReturnPath parameter to your Composer button URL with the parameter value `{!$Site.Prefix}/{!Object.Id}`. This ensures that Composer successfully redirects back to the object record in the Community.
  - The object id referenced in the ReturnPath parameter value must be the same as the object ID in the Composer base button URL.
  - You must test a Composer button with the ReturnPath parameter described above as an external Community or Portal user.

#### Community accessed by external users:

```
https://bluecubecommunity.force.com/1tng/apex/APXTConga4__Conga_Composer
?serverUrl={!API.Partner_Server_URL_370}
&Id={!Account.Id} &templateid=a0ro0000002QTTF
&returnpath={!$Site.Prefix}/{!Account.Id}
&ds7=401
```

#### Community accessed by internal users:

```
https://composer.congamerge.com
?sessionid={!$Api.Session_ID}
&serverUrl={!API.Partner_Server_URL_370}
&id={!Account.Id}
&TemplateID=a0C0b00000Uvr6o
&ReturnPath={!$Site.Prefix}/{!Account.Id}
&ds7=401
```

To send an email using Composer LC in a Community environment, the following criteria must be met:

- The community user must have either a Customer community plus license or a Partner community license
- The following General User Permissions are needed for the community user's profile:
- Access Activities

- Allow sending of List Emails
- Edit Tasks
- Mass Email
- Send Email

General User Permissions	
Access Activities	<input checked="" type="checkbox"/>
Allow Access to Customized Actions	<input type="checkbox"/>
Allow sending of List Emails	<input checked="" type="checkbox"/>
Allow user to access privacy data	<input type="checkbox"/>
Assign Topics	<input type="checkbox"/>
Convert Leads	<input type="checkbox"/>
Create Libraries	<input type="checkbox"/>
Create Topics	<input type="checkbox"/>
Delete Topics	<input type="checkbox"/>
Edit Events	<input type="checkbox"/>
Edit Opportunity Product Sales Price	<input type="checkbox"/>
Edit Tasks	<input checked="" type="checkbox"/>
Edit Topics	<input type="checkbox"/>
Export Reports	<input type="checkbox"/>
Import Leads	<input type="checkbox"/>
Import Personal Contacts	<input type="checkbox"/>
Knowledge One	<input type="checkbox"/>
Manage Articles	<input type="checkbox"/>
Manage Leads	<input type="checkbox"/>
Mass Edits from Lists	<input type="checkbox"/>
Mass Email	<input checked="" type="checkbox"/>
Merge Topics	<input type="checkbox"/>
Multi-Factor Authentication for User Interface Logins	<input type="checkbox"/>
Remain in Salesforce Classic	<input type="checkbox"/>
Report Builder	<input type="checkbox"/>
Run Flows	<input type="checkbox"/>
Run Reports	<input type="checkbox"/>
Send Email	<input checked="" type="checkbox"/>
Send Non-Commercial Email	<input type="checkbox"/>
Skip Device Activation at Login	<input type="checkbox"/>
Transfer Cases	<input type="checkbox"/>
Transfer Leads	<input type="checkbox"/>
Update Consent Preferences Using REST API	<input type="checkbox"/>
View Content in Portals	<input type="checkbox"/>
View Encrypted Data	<input type="checkbox"/>

If the Composer solution generates and saves a Salesforce File to the object record, add the ContentNetworkId parameter to the Composer solution to specify the appropriate Network Id for the Community. If the Composer solution generates and saves a Salesforce File to the object record, add the contentShareType and contentVisibility parameters to determine file access and permission.

Queries must be used instead of Salesforce Reports for related lists.

It is recommended to use Conga Trigger in the Community. Trigger allows you to set a System Admin as the running user of a Salesforce Workflow Rule, so a Community user does not need any additional permissions. The user simply triggers the workflow and the permissions belonging to the running user of the workflow are then used to trigger Composer.

## Live Edit Issue Workarounds

LiveEdit allows you to preview and edit your document before it is delivered according to the specified method like email and download. To use Live Edit, see [Using Live Edit to a Solution](#).

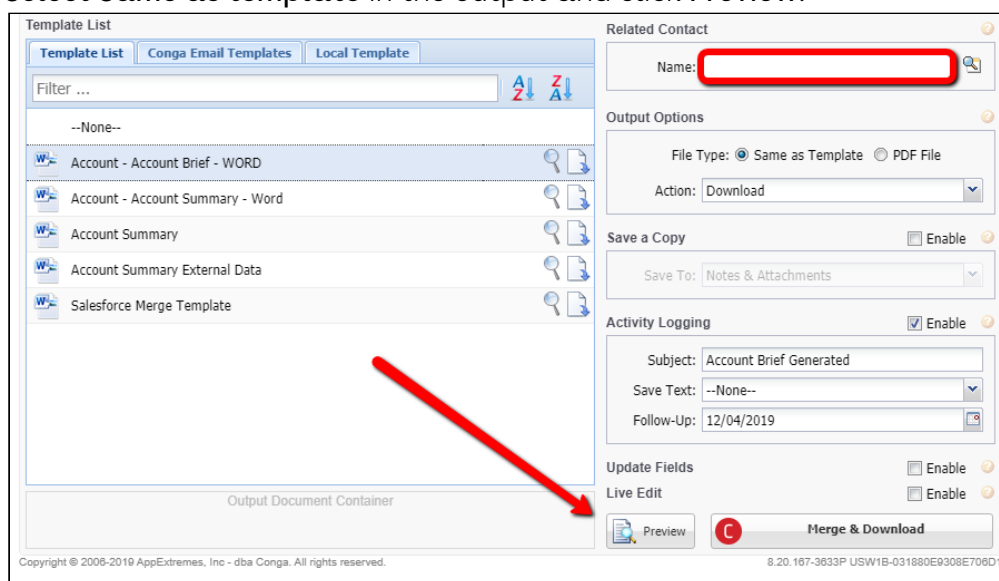
Live Edit is a simple copy/paste of the content out of the Word document, edited in Rich Text, and pasted back to the Word document.

- ❗ Live Edit does not support all Composer processes. For example, headers and footers are not supported and Live Edit does not handle watermarks or background colors.

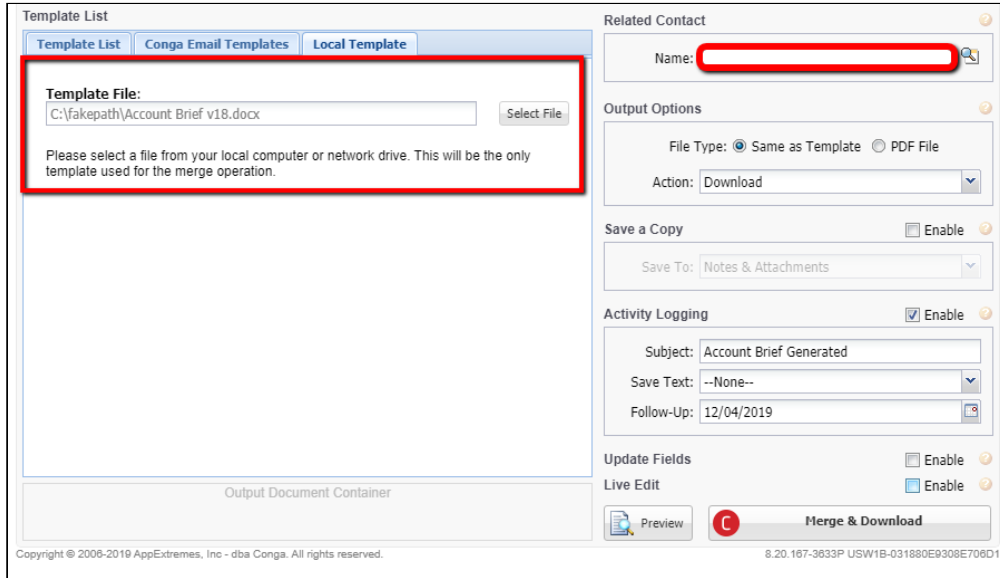
## Live Edit Workaround

To work around Live Edit limitations with Composer:

1. Ensure your Composer button is not in background mode.
2. Remove Live Edit parameters from your button.
3. Click your Composer button
4. Select **Same as template** in the output and click **Preview**.



5. A merged Word document is created. Edit and Save the document.
6. With the current dialog box open, use the Local Template tab and navigate to the Word document and finish the Merge process.



## Queries to parse data in a Conga Batch or Workflow formula

If you are passing a date range to your report, you must parse the data in a Conga Batch or Conga Trigger formula.

**⚠ This article is provided for your usage but does not include support from Conga.**

### Formula Field Query:

```
"&Queryid=[Milestones]a0336000003solv?pv0=" + Id + "~pv1=" + TEXT(Start_Date__c) + "~pv2=" + TEXT(Completion_Date__c) +
```

### Report Query:

```
"&ReportId=[Milestones]a0336000003solv?pv0=" + Id + "~pv1=" + TEXT(MONTH(Start_Date__c)) + "/" + TEXT(DAY(Start_Date__c)) + "/" + TEXT(YEAR(Start_Date__c))
```

## Dynamically show content with TableStart/TableEnd and Rowburner

**⚠ TableStart (TS)/TableEnd(TE) tags and Rowburner only work with Word Templates and do not work with Excel, PowerPoint, or PDF templates.**

## Overview

TS/TE tags and Rowburner are Composer features that remove the content in a document or a row in a table when no data is returned by a query.

Benefits of using these features include:

- Dynamically showing a table row based on conditional criteria
- Using a Word table with no borders to remove sections of content
- Removing sections of content based on business rules

To use Rowburner, create a Conga Query that returns no data when your row or content should be hidden.

## Opportunity Rowburner Example

As a Rowburner example, let us use an Opportunity with three Opportunity Line Items (OLIs). Each OLI is a product that belongs to a different product family: apps, services, and training.

Action	Product	Quantity	Date	Sales Price	Total Price	Product Family
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">CSRKing</a>	50.00	11/1/2013	\$120.00	\$6,000.00	Apps
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Installation: Portable</a>	1.00	11/4/2013	\$5,000.00	\$5,000.00	Services
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">On-Site App Training</a>	1.00	11/8/2013	\$2,000.00	\$2,000.00	Training

We want to display a row corresponding to each product on the Conga Template that we use with this Opportunity, but only if that product is present on the OLIs.

First, we write three Conga Queries, each with a filter for a different product family:

### Apps Product Family Query

```
SELECT Id, Name, Quantity, UnitPrice, TotalPrice
FROM OpportunityLineItem
WHERE Id = '{pv0}' AND Product_Family__c = 'Apps'
```

### Services Product Family Query

```
SELECT Id, Name, Quantity, UnitPrice, TotalPrice
FROM OpportunityLineItem
```

```
WHERE Id = '{pv0}' AND Product_Family__c = 'Services'
```

### Training Product Family Query

```
SELECT Id, Name, Quantity, UnitPrice, TotalPrice
FROM OpportunityLineItem
WHERE Id = '{pv0}' AND Product_Family__c = 'Training'
```

When Opportunity has products in all three families, all row(s) are displayed. Each query returns OLIs based on the product family.

### Example Template


<b>{{TableStart:Apps}}</b>	<b>{{Product_Name}}</b>	<b>{{Quantity}}</b>	<b>{{Sales_Price}}</b>	<b>{{Total_Price}}</b>	<b>{{TableEnd:Apps}}</b>
<b>{{TableStart:Services}}</b>	<b>{{Product_Name}}</b>	<b>{{Quantity}}</b>	<b>{{Sales_Price}}</b>	<b>{{Total_Price}}</b>	<b>{{TableEnd:Services}}</b>
<b>{{TableStart:Training}}</b>	<b>{{Product_Name}}</b>	<b>{{Quantity}}</b>	<b>{{Sales_Price}}</b>	<b>{{Total_Price}}</b>	<b>{{TableEnd:Training}}</b>
			Total:	<b>{{=SUM(Above) \# Currency}}</b>	

### Example Output

CSRKing	50	\$120.00	\$6,000.00
Installation: Portable	1	\$5,000.00	\$5,000.000
On-Site App Training	1	\$2,000.00	\$2,000.00
		<b>Total:</b>	<b>\$13,000.00</b>

However, if products are removed so that one of the queries returns no data (i.e., no products in a particular product family), rows are automatically removed:

Burned Row



CSRKing	50	\$120.00	\$6,000.00
On-Site App Training	1	\$2,000.00	\$2,000.00
		<b>Total:</b>	<b>\$8,000.00</b>

Let us now extend our example to include sections of content rather than just table rows. Placing any content (text, merge fields, tables, images, pages, etc.) between your TS/TE tags will either display or hide the content.

For example, let us revisit our earlier template, this time with more content, and then remove the Services product from our Opportunity.

### Example Template



Product Name	Quantity	Unit Price	Total Price
{{TableStart:Apps}}{{Name}}	{{Quantity}}	{{UnitPrice}}	{{TotalPrice}}{{TableEnd:Apps}}
{{TableStart:Services}}{{Name}}	{{Quantity}}	{{UnitPrice}}	{{TotalPrice}}{{TableEnd:Services}}
{{TableStart:Training}}{{Name}}	{{Quantity}}	{{UnitPrice}}	{{TotalPrice}}{{TableEnd:Training}}
<b>Total</b>			<b>{{SUM(ABOVE) \# Currency}}</b>


{{TableStart:Services}}

Congratulations on becoming part of the Conga Platform! Your order includes implementation services from Conga, and your new Engagement Manager will be in touch soon to kick off your project.

{{TableEnd:Services}}

## Example Output

Burned Row



Product Name	Quantity	Unit Price	Total Price
CSRKing	50	\$120.00	\$6,000.00
On-Site App Training	1	\$2,000.00	\$2,000.00
<b>Total</b>			<b>\$13,000.00</b>

Burned Content



 For Rowburner to work, a table row must have at least two columns.

## Use Cases

Using TS/TE tags and Rowburner, you can easily show/hide content dynamically. Consider the following use cases for meeting business requirements around dynamic content:

### Conditional Contractual Language

If you are not using the Conga Contracts for Salesforce Clause Library, Rowburner is an excellent feature for displaying or hiding terms and conditions, additional provisions specific to a product, or any kind of contractual language that may conditionally apply to a particular transaction. Queries can be used in multiple places in a template (like the OLI example above) to burn a table row and/or additional language according to a particular product.

Here is an example using a Rowburner in a table with borders removed:

	<p><b>1.3.</b> Supplier will document and Customer shall review and approve a set of business rules for the services under Supplier's responsibility. The mutually agreed business rules will be documented in the Runbook. Supplier will monitor BPO process compliance against the Runbook and will review the Runbook with Customer once a year to ensure content is up to date.</p>	
<p>{{TableStart:TMBP.Q}}</p>	<p><b>1.4.</b> Supplier will monitor Supplier-implemented system interfaces (as described in Schedule C Section 8) and raise a ticket with either the Software Help Desk or Customer IT Support when an error is detected, depending on source of the error. Whenever possible, Customer shall notify Supplier at least sixty (60) days in advance of any planned changes to the format of the automated data feeds so that Supplier can take appropriate actions to update the interface and avoid unnecessary error and process interruption.</p>	<p>{{TableEnd:TMBP.Q}}</p>

## Granular Control over Line-Item Placement

If you are merging in related lists of records and need specific control over the order of line items that cannot be set in Salesforce, you can use specific queries with Rowburner. This allows you to dictate which types of records are merged into which rows by using different filters on the associated SOQL queries, and Rowburner will remove any rows that do not apply to a particular merge.

In this example, we merge all Recurring products first, followed by all One-Time products:

Description	Item	Qty	Unit Price	Extended	Discount	Total
<p>{{TableStart:QLIRecurring}}{{SBQQ__QUOTELINE_QUOTE_PRINT_PRODUCT_NAME}} - {{SBQQ__QUOTELINE_SEGMENTLABEL}}</p>	<p>{{SBQQ__QUOTELINE_PRODUCTFAMILY}}</p>	<p>{{SBQQ__QUOTELINE_QUANTITY}}\# Number0}</p>	<p>{{SBQQ__QUOTELINE_REGULAR_PRICE}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_REGULAR}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_TOTALDISCOUNT}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_NETTOTAL}}\# Currency}} {{TableEnd:QLIRecurring}}</p>
<p>{{TableStart:QLIOneTime}}{{SBQQ__QUOTELINE_QUOTE_PRINT_PRODUCT_NAME}} - {{SBQQ__QUOTELINE_SEGMENTLABEL}}</p>	<p>{{SBQQ__QUOTELINE_PRODUCTFAMILY}}</p>	<p>{{SBQQ__QUOTELINE_QUANTITY}}\# Number0}</p>	<p>{{SBQQ__QUOTELINE_REGULAR_PRICE}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_REGULAR}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_TOTALDISCOUNT}}\# Currency}}</p>	<p>{{SBQQ__QUOTELINE_NETTOTAL}}\# Currency}} {{TableEnd:QLIOneTime}}</p>
				<p>{{AUTOSUM(ABOVE)}}\# Currency}}</p>	<p>{{AUTOSUM(ABOVE)}}\# Currency}}</p>	<p>{{AUTOSUM(ABOVE)}}\# Currency}}</p>

## Conditional Content

If you have any form of conditional content - excess usage charges, regional taxes, product-dependent clauses, number of signers that change based on deal size, etc., you can use TS/TE tags and/or Rowburner to show/hide any of that content.

For example, let us say you need two internal countersignatures for a deal with an ACV >= 100k:

- Create two queries that return data only if ACV < 100k or ACV >= 100k, respectively

- Add TS/TE tags to your template, with one set surrounding a signature block that includes 2 signers and then another set around a signature block that includes 3 signers:

<b>Internal Signer</b>		<b>Client</b>	
<b>Signature</b>		<b>Signature</b>	
<b>Typed Printed Name</b>		<b>Typed Printed Name</b>	
<b>Title</b>		<b>Title</b>	
<b>Date</b>		<b>Date</b>	
<b>Internal Signer 1</b>			
<b>Internal Signer 2</b>		<b>Client</b>	
<b>Signature</b>		<b>Signature</b>	
<b>Typed Printed Name</b>		<b>Typed Printed Name</b>	
<b>Title</b>		<b>Title</b>	
<b>Date</b>		<b>Date</b>	

- [Conditionally Remove a Row in a Table that has No Data](#)

## Conditionally Remove a Row in a Table that has No Data

Rowburner is to be used when you have a row on a table that may not always return data. If there is no relevant data populated the entire row should not appear on the merged output.

This feature is to be used when you have a row on a table that may not always return data, such as the table with different Product Families below. If there is no relevant data populated (no products from that family in the opportunity) that entire row should not appear on the merged output. For additional information see [Dynamically show content with TableStart/TableEnd and Rowburner](#).

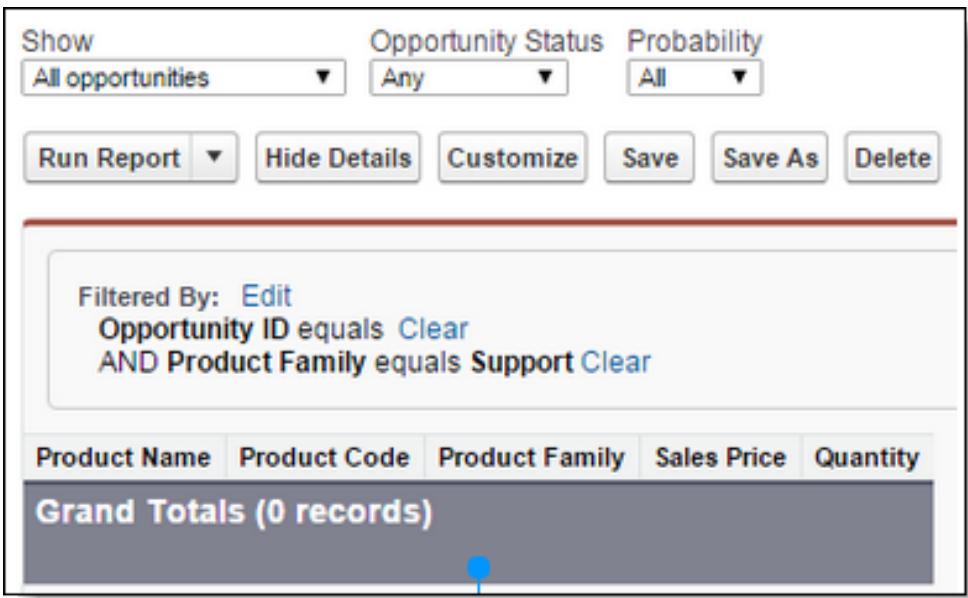
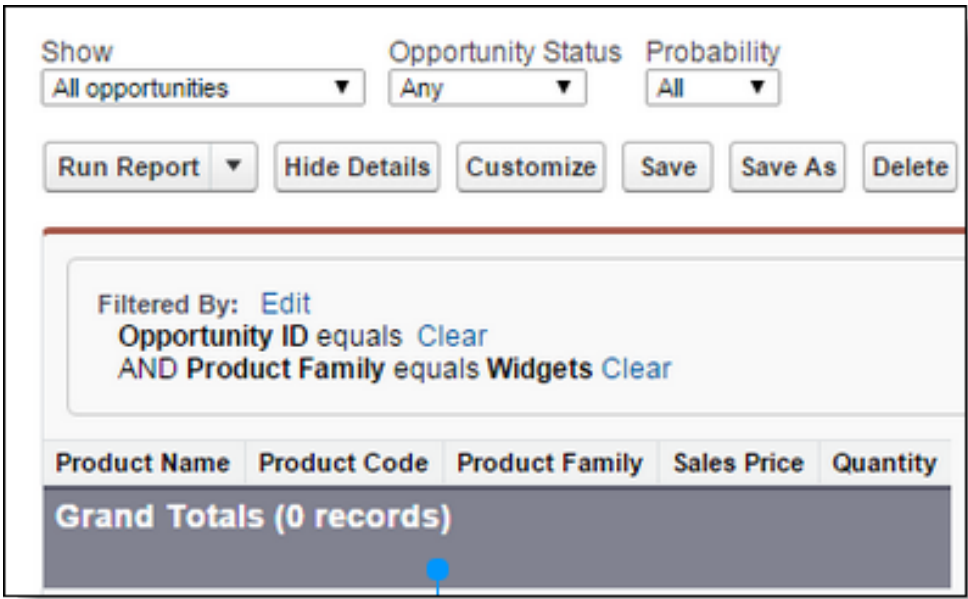
For similar information, see [Using Detail Regions with If Statements](#).

**What Is Needed:**

1. A report or query that will return data if that row should populate. In this example, the button has three reports, one for each product family.

```
• &ReportID=[Widgets]00000000004DUqa, [Support]00000000004DURJ, [Generators]00000000004DUqk
```

2. The Report or SOQL Query should have filters that dynamically return data based on the record. These are the reports from the Button URL above, filtered by Product Family.





ⓘ Each Dataset must have its own row with a TableStart:ReportData field in the first column and a TableEnd:ReportData field in the last column.

Template:

{{OPPORTUNITY_NAME}}			
Product Name	Product Code	Sales Price	Quantity
{{TableStart:Widgets}}{{Product Name}}	{{Product Code}}	{{Sales Price \# Currency}}	{{Quantity \# 0}}>{{TableEnd:Widgets}}
{{TableStart:Support}}{{Product Name}}	{{Product Code}}	{{Sales Price \# Currency}}	{{Quantity \# 0}}>{{TableEnd:Support}}
{{TableStart:Generators}}{{Product Name}}	{{Product Code}}	{{Sales Price \# Currency}}	{{Quantity \# 0}}>{{TableEnd:Generators}}
Total Owed:		\$	0.00

Output:


If we merge from an opportunity with products from all three families, we'll see data from each dataset:

Support and Generators and Widgets			
Product Name	Product Code	Sales Price	Quantity
Deluxe Large Widget	W-004	\$850.00	1
Extra Large Ultra-Luxe Widget	W-005	\$1,500.00	1
Large Widget	W-003	\$565.00	1
Medium Widget	W-002	\$375.00	1
Mystery Widget	M-999	\$10.00	1
Small Widget	W-001	\$150.00	1
Tier 1 Support	SR-001	\$100.00	1
Tier 2 Support	SR-002	\$200.00	1
Tier 3 Support	SR-003	\$300.00	1
15,000 GW Generator	G-999	\$500,000.00	1
1500 KW Generator	G-001	\$100.00	1
Total Owed:		\$504,150.00	

Now let's merge from an opportunity that only has products from the 'Support' and 'Generator' Product families:

Support and Generators			
Product Name	Product Code	Sales Price	Quantity
Tier 1 Support	SR-001	\$100.00	1
Tier 2 Support	SR-002	\$200.00	1
Tier 3 Support	SR-003	\$300.00	1
15,000 GW Generator	G-999	\$500,000.00	1
1500 KW Generator	G-001	\$100.00	1
Total Owed:		\$500,700.00	

The rows for Support and Generators were merged, and the rows with no data (Widgets) were neatly removed by the Row Burner.

 You must have at least two columns in your row for row burner to work.

# Requirements for Sending Automatic Emails with Conga

## Sending Automatic Emails with Conga

When describing the automatic sending of emails using Conga products, we're referring to any method that delivers the email over the API rather than via the user interface. These API delivery methods are:

- Composer in background mode (DS7=12 parameter value)
- Conga Batch
- Conga Trigger


## Required Email Parameters

- **&EmailTold=[Contact ID]**
- **&EmailTemplateId=[Salesforce Email Template ID] or &CongaEmailTemplateId=[Conga Email Template ID] (the &CETID)**
  - If you're using a Salesforce email template, you must also use the **&TemplateId** parameter to specify a document template.
  - If you're using a Conga Email Template, a document template is optional.

## Optional Email Parameter

- **&EmailRelatedTold=[Object ID]**
  - If using a Salesforce Email Template, the merge fields that are available for use on that template are fields from the Object referenced in the **&EmailRelatedTold** parameter. Contact fields also merge if you've included the **&EmailTold=[Contact Id]**.
  - Specifies which record to relate the Salesforce email task, not the task that Conga may create as part of activity logging. The Salesforce email task always starts with "Email:" and can be related to two records: 1) the Contact to which the email was sent, and 2) the record to which the **&EmailRelatedTold** parameter is set.
  - This parameter defaults to the Master record if not specified otherwise in the URL button/formula.
  - Can't Specify a Contact as the Email Recipient? Use this workaround for not using **&EmailTold**:

- If you are unable to specify a Contact ID as the primary recipient of the email using &EmailTold, the work-around solution is to omit the email template and send the recipient an email with a Subject, Attachment, and no Body (in other words, the body of the email is blank but has a subject line and a document attachment, created by Conga Composer). The recipient is specified using an email address rather than a Contact ID.
- You can omit &EmailTold if these parameter instructions are followed:
  - EXCLUDE:
    - &EmailRelatedTold=[Object ID]**
    - &EmailTemplateId=[Salesforce Email Template ID] or &CongaEmailTemplateId=[Conga Email Template ID] (a.k.a. &CETID)**
  - INCLUDE:
    - &EmailSubject=[Text]**
    - &EmailAdditionalTo=[Email Address(es)]**
- Composer can customize the From: address of an email with two different parameters:
  - **&EmailFromId=[Org-wide email address]**
    - Set to the Salesforce ID of an active Org-wide email address, that's it!
  - Set to the Salesforce ID of an active User (not a Contact!)
    - Alters two properties of the email message: 1) the DisplayName (the name that is associated to the "from" address), and 2) the Reply To (the email address to which replies are sent)
    - Your email appears to come from the User specified (because we change the Display Name)
    - If the recipient clicks Reply in their inbox, they reply to the User specified (because we changed the ReplyTo)
    - The DisplayName property is not editable when using SendGrid.
    - This parameter does not affect the From attribute of the message, which is the actual email address from which the email is sent.
    - The actual From property of the message contains the email address of the person who: clicked the Composer button, scheduled Conga Batch, or is the User to Send As of the workflow outbound message specified in Conga Trigger.**&EmailReplyTold=[User ID]**

 The &EmailFromId and &EmailReplyTold parameters cannot be used together because the functionality of each conflict with one another (they both alter the Display Name). You must choose one or the other.



## Salesforce API Email Limitations

- 24-hour email limit: Salesforce limits you to a certain number of emails which can be sent through the API per day. **\*\*\*Contact Salesforce Support for details on the limitation of your instance\*\*\***
- Once the daily API email limit is exceeded, the Composer/Conga Batch/Conga Trigger process will return an error, " **Problem: ERROR: Email delivery failed: SINGLE\_EMAIL\_LIMIT\_EXCEEDED .Failed to send email.** "
- In order to work around these Salesforce limits, a customer should integrate [SendGrid](#) email delivery service in their Conga Composer solution(s).

## Streamlining Template Selection in Conga Composer

You can use several methods to select a template more quickly in a Conga Composer solution.

- [The TemplateID Parameter](#)
  - The TemplateID parameter selects a specific template when the solution runs. Additionally, you can specify multiple template IDs to select multiple templates. A template ID can be a Conga Template record ID, a Salesforce Document record ID, or a Salesforce Attachment record ID.
- [The TemplateGroup Parameter](#)
  - Templates can be assigned to custom categories using the TemplateGroup feature. There is no limit to the number of categories that can be created, and individual Composer buttons can be configured to only use templates that belong to certain groups.
- [The SelectTemplate Parameter](#)
  - The SelectTemplates parameter allows the running user to select more than one template and arrange them as necessary. SelectTemplates can be used in conjunction with TemplateGroup in order to limit the templates that can be selected for a merge process. It can also be used with the TemplateID parameter to specify templates, and allow users to add additional templates during a merge.
- CASE or IF in Button URL
  - Salesforce formulas, such as IF or CASE formulas, can be leveraged within a Composer button to [dynamically select templates](#). Most commonly, these formulas are used with the &TemplateID parameter to include or exclude certain templates conditionally. These could also work with the &TemplateGroup parameter to conditionally define template categories.
- Formula fields on Object

- Using standalone Salesforce Formula fields is arguably the most flexible way to drive template selection for Composer. Business-specific formulas can be created to evaluate a record (or related records) in order to determine which templates are selected under any given circumstance. This is definitely the best approach for applying business-specific logic and rules to the document generation process with Composer.
- Conditional formatting in a template
  - Depending on the type of template, there are different features available for merging or displaying content dynamically based on logical rules. When using PowerPoint templates, the following two features are available:
    - Report-based SlideHide – A slide (or slides) can be hidden if a Salesforce Report does not return any records.
    - Field-based SlideHide – A slide (or slides) can be hidden if a field has a certain value. This also works when evaluating if a field has a null value.

## Use Case for QueryStringFields

[Salesforce.com](https://www.salesforce.com) limits formula fields to 1300 characters. Once you hit the 1300 character limitation you have to split up your formulas into two to four separate fields.

You can do this by using the API Field Name of another formula field which contains Conga Parameters. Using the QueryStringField parameter, you will reference your QueryString formula field by its API field name in your Conga Trigger formula.

### Formula Field

Account Custom Field  
**CN Acct Brief\_QueryStringField**  
[Back to Account Fields](#)

**Custom Field Definition Detail** Edit Set Field-Level Security View Field Accessibility

**Field Information**

Field Label	CN Acct Brief_QueryStringField
Field Name	CN_Acct_Brief_QueryStringField
API Name	CN_Acct_Brief_QueryStringField__c
Description	
Help Text	
Created By	Pete Castro, 9/25/2013 2:38 PM

**Formula Options**

Data Type	Formula
	"&id=" + Id + "&QueryStringField=CN_Acct_Brief_QS0__c" + "&QueryStringField1=CN_Acct_Brief_QS1__c" + "&TemplateId=a01E000000CBzqk" + "&DefaultPDF=1" + "&OFN=" + TEXT(TODAY()) + "+Account+Brief+for" + Name + "&QMode=1"

**Conga Parameters: &QueryStringField=**  
**API Field Name: CN\_Acct\_Brief\_QS0\_\_c**

### Query String Field 1

Home Leads Accounts Contacts Opportunities Cases Reports Conga Mail Merge Conga Templates (v5) Conga Email Templates

Quick Find

Expand All | Collapse All

**Force.com Home**

**System Overview**

**Personal Setup**

- My Personal Information
- Email
- Import
- Desktop Integration
- My Chatter Settings
- My Social Accounts and Contacts

**App Setup**

- Customize
  - Tab Names and Labels
  - Home
  - Activities
  - Campaigns
  - Leads
  - Accounts

Fields

Account Custom Field  
**CN Acct Brief\_QS0**  
[Back to Account Fields](#)

**Custom Field Definition Detail** Edit Set Field-Level Security View Field Accessibility

**Field Information**

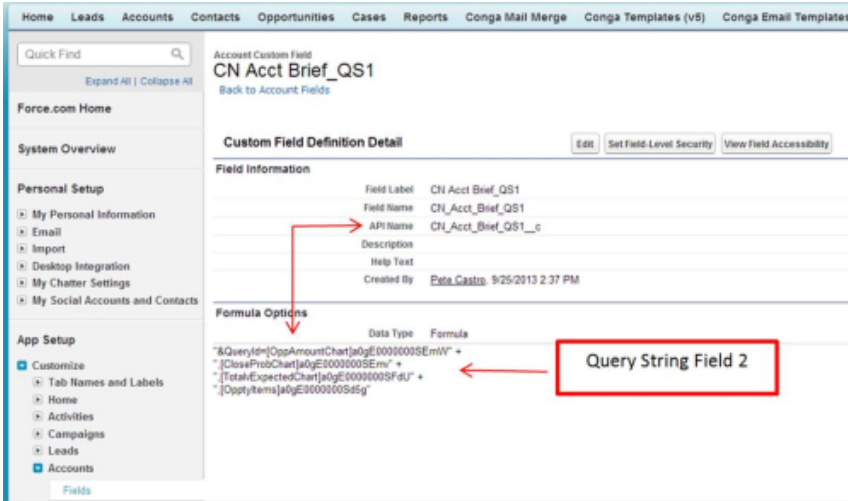
Field Label	CN Acct Brief_QS0
Field Name	CN_Acct_Brief_QS0
API Name	CN_Acct_Brief_QS0__c
Description	
Help Text	
Created By	Pete Castro, 9/25/2013 2:35 PM

**Formula Options**

Data Type	Formula
	"&ReportId={Contacts}000E000000hZkY" + "&Opportunities}000E000000hZkY" + "&Cases}000E000000hgVf" + "&Activity}000E000000h8d"

**Query String Field 1**

### Query String Field 2



The Query Strings connect the separate formula fields and pieces them together to work around the 1300 Salesforce character limit.

## Use Salesforce Functions (IF, OR, ISPICKVAL) in a Conga Composer Button URL

Using functions in Salesforce custom button or link URLs provides a way to dynamically determine behavior for Conga Composer parameters. This allows a Composer button or link to perform different functions based on certain criteria, including the values of fields in Salesforce. This logic can be applied using different Salesforce functions for various parameters or conditions within the Composer button URL. See Salesforce Help's Formula Operators and Functions Overview to familiarize yourself with the various functions available.

**i** Using the methods described below, the functions are NOT selecting which Composer parameter to use but simply selecting the value to provide for a certain Composer parameter.


### Example 1 - Simple IF Function

Let's say that you want to generate invoices from Opportunity records, but want to lock the file output format to PDF for all users with a profile other than System Administrator.

Logically, we want to set the value of the &FP0 parameter to either 0 or 1 depending on the profile of the current user.

1. Create the parameter you'd like to provide a conditional value for, but leave its value blank.

2. Insert the IF function (from the Salesforce drop-down).
3. Define the logical test - The 'logical\_test' clause will define the condition for what our parameter value will be set to. For this example, we'll insert the merge field for the User Profile ID, and specify that we want to define what value is provided when that equals "System Administrator."

 Within any Salesforce function, literal text is enclosed in either single (e.g. 'System Administrator') or double quotations (e.g. "System Administrator").

4. Define the true and false values - The next step is to define what is displayed if the condition evaluates to "true." For our "true" value, if the user's profile is "System Administrator," we want to set FPO equal to 0 so that the output file format is not locked. For our "false" value, the user's profile is not "System Administrator," we want to set FPO equal to 1 so that the output file is locked.
5. Test your solution to ensure your IF function is controlling the behavior as desired.

## Example 2 - Evaluating Multiple Field Values - IF / OR / ISPICKVAL Function

The management team has dictated that an invoice can be generated with Conga Composer only if the Stage on the Opportunity record is one of three values: Proposal Made, Due Diligence, or Closed Won. We'll use the Composer DC parameter to specify this condition.

1. Start with the DC parameter and the IF function:

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&DC={!IF (logical_test, value_if_true, value_if_false)}
```

2. Next, because Opportunity Stage is a picklist field, which are only supported in certain functions, we must use the ISPICKVAL function to determine the value of the Stage field.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&DC={!IF( ISPICKVAL(Opportunity.StageName, "Proposal Made"), value_if_true,
```

```
value_if_false)}}

```

3. Replace value\_if\_true and value\_if\_false with the correct values for the DC parameter. In this example, the value\_if\_true is "0" because if the Stage is Proposal Made, we want DC=0, which allows Composer to run. The value\_if\_false is "1" because we want to disable Composer (DC=1) if the Stage is not Proposal Made.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&DC={!IF( ISPICKVAL(Opportunity.StageName, "Proposal Made"), "0", "1")}}

```

4. Now we add the OR function because we have three approved values for Stage (if we only had one approved value, we could simply use the above IF / ISPICKVAL functions from step 3).

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&DC={!IF (OR ( ISPICKVAL(Opportunity.StageName, "Proposal Made"),
logical2,...), "0", "1")}}

```

5. Replace logical2 and ... with the second and third Stage values, remembering to include the ISPICKVAL functions. Note the double closed parentheses )) after the "Closed Won" value to indicate the end of the OR function.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&DC={!IF (OR ( ISPICKVAL(Opportunity.StageName, "Proposal
Made"),ISPICKVAL(Opportunity.StageName, "Due
Diligence"),ISPICKVAL(Opportunity.StageName, "Closed Won")), "0", "1")}}

```

6. Use the Check Syntax button at the bottom of the Salesforce button page to ensure you don't have any syntax errors. Now, go test your solution!

## Example 3 - Evaluating Whether a Field Is Blank - IF / OR / ISNULL Function

If either the Case Principal, Case MRF, or Case Monthly Rate fields are blank (the ISNULL function evaluates whether a field is populated or not), we don't want Conga Composer to run (and would therefore want to enable the DC parameter to disable Conga Composer -- &DC=1). The following functions would accomplish this:

```
&DC={!IF( OR( ISNULL(Case.Principal__c), ISNULL(Case.MRF__c),
ISNULL(Case.Monthly_Rate__c)), 1, 0)}
```

## Use a Salesforce Attachment or Salesforce Files as a Conga Template

In addition to pre-selecting a Conga Template, Conga Composer can also gather attachments from the Notes and Attachments object related to the master object from which you are running your Conga Composer solution. This is helpful if there are additional documents that should be included with your merged document. To learn more about this process please review this Conga Blog post under number 2, "Using a Salesforce Attachment as a Conga Template."

### Important

The Composer Lightning component does not support it.

For example, you may want Conga Composer to generate a proposal document from an Opportunity record but also need to include the Terms and Conditions document that is stored as an attachment (under Notes and Attachments) on the Opportunity.

1. Are you still using Notes and Attachments? If so, create a SOQL Query to retrieve the attachment(s) that looks like this: `SELECT Id FROM Attachment WHERE ParentId = '{pv0}'`  
Have you switched to Salesforce files? Use this query: `SELECT ContentDocument.id FROM ContentDocumentLink WHERE LinkedEntityId = '{pv0}'`
2. Next, reference this query via its ID using the &QVar0Id (Query Variable) parameter.  
For example: `&QVar0ID=a0Ai0000009jpZH`
3. The SOQL Query from Step 1 will return the ID of the attachment, which we now have available in the Conga Composer button URL to use as a variable in another parameter, &AttachmentID. Reference the Query Variable, as {QVar0}, as the value for the &TemplateID parameter.  
For example: `&TemplateID=a0Bi000000MrkC8, {QVar0}`

Below is an example of what your button URL might look like when the &QVar0Id and &TemplateID parameters are added.

In this example, {!Account.Id} is being passed into the pv0 filter of the query in order to retrieve the attachment(s) from the Account record related to the Opportunity. Furthermore, we're using the &QVar0Format parameter to format the output of the results retrieved in the Qvar0 query; this is necessary for situations where more than one attachment may be retrieved.

```
https://composer.congamerger.com?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&QVar0ID=a0Ai0000009jpZH
&TemplateID=a0Bi000000MrkC8,{QVar0}
&QVar0Format=10010
&DefaultPDF=1
```

**i** The above method is supported by both TemplateID and AttachmentID. AttachmentID will simply append the file. TemplateID will process the file.

## Using Composer to Update Fields in Salesforce

Composer can be used to update fields in Salesforce. There are two ways to do this: in the Composer button URL or on a Conga Template Record. Both methods are described below.

**✓** The primary difference between the two methods is whether the fields update every time the Composer button runs or every time a specific template is used in a merge.

### Method 1: Update Fields in the Composer URL

With this method, the field updates are handled entirely in the Composer button URL, which means the updates take place every time the Composer button runs.

To update fields in the Composer URL:

1. Add the following three parameters to your button URL. See Adding Parameters to a Solution for more information on this process.
  - UF0 - enables Update Fields feature.
  - MFTS - identifies the field to update.
  - MFTSValue - defines the value to update the field with.



To update the Member Status field with a value of Patron every time your Composer button runs, add the following parameters to your button

URL: `&UF0=1&MFTS0=Member_Status__c&MFTSValue0=Patron`

To update multiple fields simultaneously, increment the MFTS and MFTSValue parameters (MFTS1, MFTSValue1, MFTS2, MFTSValue2).

## Method 2: Update Fields on the Conga Template Record

With this method, the field updates are handled in the Conga Template record and with a parameter, which means the field updates take place when a specific template is used in a merge.

To update fields on the Conga Template Record:

1. Add the UF0 parameter to your button URL. See Adding Parameters to a Solution for more information on this process.
2. Update the Master Field to Set fields on the Conga Template record.

To update the Member Status field with a value of Patron when the specific template is used in a merge, do the following:

1. Add the UF0 parameter to your button URL.
2. Add Member\_Status\_\_c=Patron to Master Field to Set 1 on the Conga Template record of the template you want to enable the field update.

With this method, you can update three fields at once using Master Field to Set 1, Master Field to Set 2, and Master Field to Set 3. This method is only available for document templates (Word, Excel, PowerPoint, PDF); it does NOT work with Conga Email templates.

## Using Live Edit to a Solution

Live Edit allows you to preview and edit your document before it is delivered according to the specified method like email and download.

## Considerations

The following limitations apply:

- Microsoft Word templates are the only supported template type.
- If a non-Word template is predefined in the URL or selected in the interface, the UI presents a warning message and disables the Live Edit checkbox.
- Only single Word templates are supported.

- The template must be saved in Salesforce (local templates do not work).
- Text boxes are not supported.
- Resized images may not carry over into the output document as expected.
- Some advanced formatting might not carry over into the output document as expected.
- There is no "write back" functionality between Salesforce and Live Edit. For example, if you change information like a phone number or name in Live Edit that same information still needs to be updated in Salesforce in order to match.
- Headers are not supported with Live Edit.
- Live Edit might negatively impact Watermarks.
- When IF statements are present in the Word template, it is advised to use the parameter, `wordStripIfs=1`.

Live Edit is a simple copy/paste of the content out of the Word document, edited in Rich Text, and pasted back to the Word document.

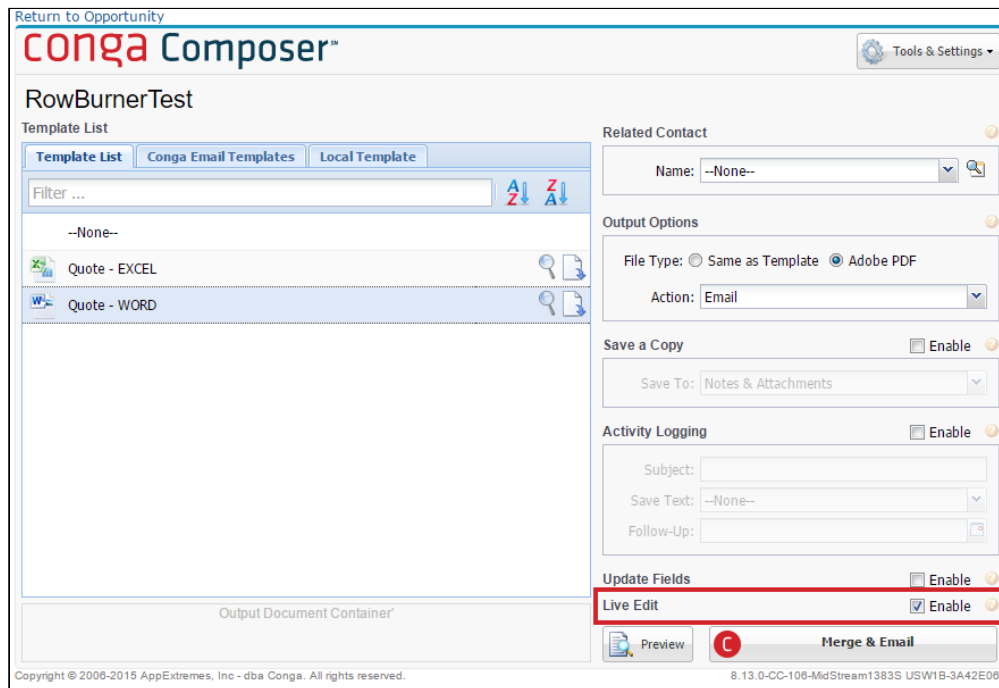
❗ Live Edit does not support all Composer processes such as headers, footers, watermarks, or background colors. For example, headers and footers do not display in the Live Edit window and consequentially do not display in the final merged document. Live edit does not support multiple templates in a Composer solution.

## To use this feature

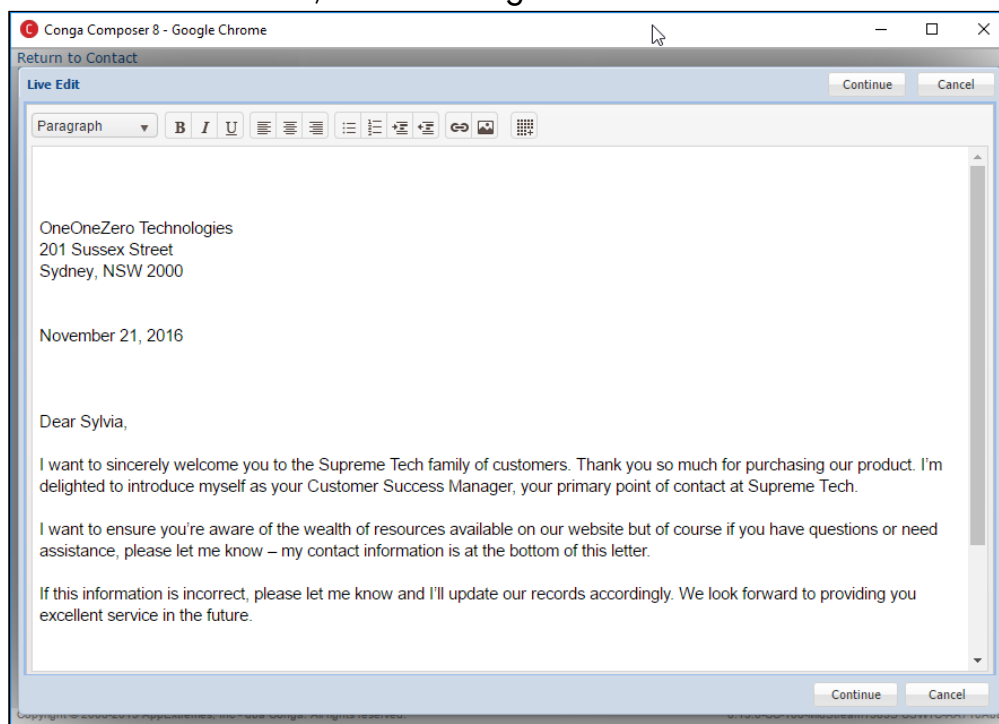
1. Add the following parameters to your button or link URL:
  - `&LiveEditVisible=1`, which enables the Live Edit option on the Composer user interface.
  - (Optional) `&LiveEditEnable=1`, which checks the Live Edit option by default.
2. If you have not already, add your button to a page layout.


## To test this feature

1. Click the *Composer Solution* button to launch your solution. When the Conga Composer user interface loads, you will see an option for Live Edit. If you used the `LiveEditEnable` parameter, the option will be checked as well (see screenshot below).



2. With Live Edit checked, click the **Merge** button. The *Live Edit* window launches.



 Some templates do not display accurately in the Live Edit window. Trigger a full merge to see the final results.

In the Live Edit window, you can make changes to your document. The toolbar (shown below) allows you to make formatting changes, including adding links, images, and tables. Be aware that formulas and other data have already been calculated when using Live Edit, so edits to calculations need to be made in all applicable locations.

For example, changing a value in a list does not automatically update the total amount; that value needs to be updated manually as well.



3. After making changes, click the **Continue** button. The merge process and deliver the document finalizes.

## Using Qvars in a Conga Composer Solution

Use QVARs in your Conga Composer solutions.

### QVars

QVars (Querystring Variable) allow users to reference the results of Conga Queries as the value of other parameters in the same Composer solution. The benefit of QVars is referencing gathered data for use in a Composer solution's button URL or formula field.

### QVar Parameters

There are two QVar parameters used in Composer solutions:

- QVar0ID
  - The QVar0ID parameter designates a specific Conga Query record for Composer to execute.
    - The parameter value is a Conga Query's 15 digit record ID.
  - This parameter is required to use QVars in a Composer solution.
  - The maximum amount of QVar0ID parameters in a Composer solution is six.
    - Each QVar is represented as an individual parameter, such as QVar0ID, QVar1ID, and QVar2ID.
    - Users can relate only one Conga Query record ID for each QVarID parameter.

**i** The Conga Query must return only one column of data and must not retrieve more than **50** rows of data.

- QVar0Format
  - The QVar0Format parameter formats the results of the QVar0ID parameter's executed SOQL Query.
    - The parameter value is a 5 digit code.

- This parameter is not required for all QVar use cases, but it is highly suggested to use for QVar solutions that return multiple data values.
- Users can use the QVar0Format parameter up to six times in a Composer solution.
  - Each QVar0Format parameter corresponds to a QVar0ID parameter, such as QVar0Format, QVar1Format, and QVar2Format.
- QVar0Format is useful for separating multiple retrieved values by delimiters, enclosing retrieved values by apostrophes and quotes, and truncating retrieved record IDs to 15 digits.
  - For more information on formatting retrieved data with the QVar0Format parameter, see [QVar0Format](#).

## Using {Qvar} as a Parameter Value

{Qvar0} is the syntax used to reference the result of a QVar0ID parameter's related SOQL Query. Each {Qvar} correlates to a specific QVarID parameter.

For Example, users would use {Qvar0} as a parameter value to reference QVar0ID's retrieved query results. If a user references the results of Qvar0ID as the EmailToID parameter's value, the syntax is &EmailToId={Qvar0}.

Use QVarID parameters in sequential order. For Example, add the QVar0ID before QVar1ID in a Composer button url or formula field. Using the QVarID parameters out of sequential order may result in unstable QVar performance.

## QVar Use Cases

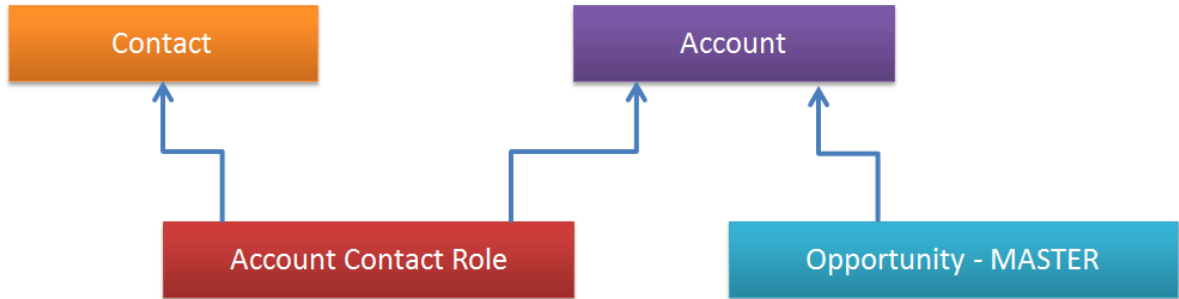
The examples below depict common use cases for using QVars in Conga Composer solutions.

### Using a Contact ID as an Email Recipient

- The EmailToID parameter is a useful way to specify an email recipient for a Conga Email Template. The EmailToID parameter requires a Contact, User or Lead record ID as the parameter value. Salesforce's custom buttons allow users to reference Contacts through lookup fields, or with the Primary Contact Role, but there are many scenarios where users must traverse through related records to find the right email recipient.
- Instead of finding the correct related Contact record in Composer's email user interface, use a QVar to select an email recipient automatically.
- For example, a user needs to send a proposal from an Opportunity using Conga Composer and the email recipient is always going to be the related Account's Primary Contact. There is not a way to reference that contact's record ID directly from the

Composer button on the Opportunity. To resolve this issue, use a QVar to reach up to the Account, locate its Primary Contact, and retrieve the Primary Contact's record ID.

- Example Object Schema:



- Example SOQL Query for QVar:

**Conga Query**  
CMQ-0032

**Conga Query Detail** Edit Delete Clone Conga Query Builder

Conga Query Name	CMQ-0032
Owner	<a href="#">Robert Boyd [Change]</a>
Name	QVar - Account Primary Contact
Description	
SOQL Select Statement	<b>SELECT ContactId FROM AccountContactRole WHERE AccountId = '{pv0}' AND IsPrimary = TRUE</b>
Created By	<a href="#">Robert Boyd</a> , 12/5/2013 9:02 AM

Edit Delete Clone Conga Query Builder

- The Account ID is the pv0 variable in the above SOQL Query and the query is filtered to return only Primary Contact Roles.
- The next step is to take the Conga Query's 15 digit record ID and use it as the value of a QVar. The necessary QVar parameter in the Composer button is QVar0ID. This parameter makes the result of the query accessible as a variable in the Composer solution. Once defined, the variable is available for use by referencing it as {QVar0}.
- Example parameter configuration:

**1** **2**

```
&QVar0Id=a04E0000004S1Zr?pv0={!Opportunity.AccountId}
&EmailToID={QVar0}
```


**3**

- a. The Conga Query ID is referenced with &QVar0Id.
- b. The Opportunity's related Account ID is passed to the first filter in the query.

- c. The {QVar0} result is used as the value for the &EmailToID parameter.
- When a user launches the Composer button and the query executes, the actual 15 digit Salesforce ID of the desired Contact record replaces {QVar0} and the email recipient is automatically selected.

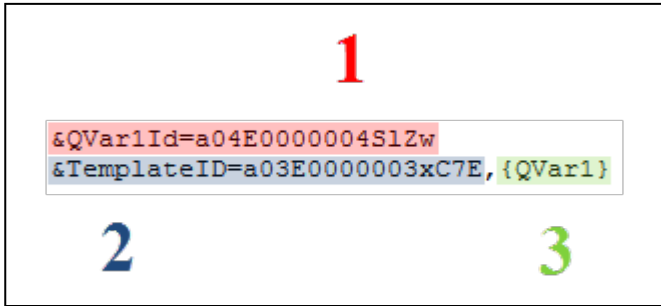
### Using a Salesforce Attachment as a Conga Template

- Conga Composer Templates are typically stored and referenced by uploading the file to the Conga Template Manager. The TemplateID parameter can reference the record IDs of these Conga Templates to predefine template selection.
- In many situations, it makes sense to include a file that is attached to a Salesforce record as a template for a Composer solution. This is not usually a template with merge fields included - more often it is a document such as one of the examples below:
  - Custom terms and conditions added to a proposal document.
  - A candidate's resume included in a job application document.
  - A contract addendum.
- A QVar is useful to retrieve the record ID of a Salesforce Attachment record and then reference that Attachment record as the value of the TemplateID parameter. In this use case, the SOQL Query requires some additional custom filtering to direct the QVar on what type of Attachment to select.
- For example, a user wants to select the file attached to an Opportunity that represents custom terms and conditions for a proposal.
  - Example SOQL Query:

```
SOQL Select Statement  SELECT Id FROM Attachment WHERE ParentId = '{pv0}' AND Name LIKE '%Terms%'
```

- Similar to the previous example, there are two filters used in the query. The first filter looks for a dynamic parent ID and the second filter looks for Attachments with the word "Terms" in the file name. The second filter is entirely customized to the type of file a user needs to select. A user could also filter for a specific file extension, a particular keyword, or any other unique identifier.
 

Tip: Establishing a file naming convention is a recommended best practice when using this type of QVar. Setting up QVars to retrieve Attachments is only worthwhile if it is used dynamically across records for the same purpose.
- Example parameter configuration:



- a. The Conga Query ID is referenced with &QVar1Id.
  - b. The proposal template is hard-coded as the first template merged using the &TemplateID.
  - c. The {QVar1} result is used as the second value for the TemplateID parameter, separated by a comma.
- When a user launches this Composer button, it replaces {QVar1} with an Attachment record ID and then Composer merges both templates. The merged templates may be combined into a single output file or merged separately in a .zip file.

### Using Multiple Email Addresses as CC Recipients

- The previous examples both used QVars to retrieve single data values (Contact ID and Attachment ID). However, QVars are highly useful for returning multiple data values, such as a handful of email addresses that need to be included as additional recipients on an outbound email.

When working with multiple values in a QVar, it is almost always necessary to format the result to separate each value by commas or surround each value with quotations. The QVar0Format parameter simply allows users to format a QVars resulting data.

In this scenario, a user is on an Opportunity record and needs to email the related Account's Primary Contact (accomplished with the QVar in the first example) in addition to CCing related Contacts with Opportunity Contact Roles of "Decisions Maker".

- Example SOQL Query for QVar:

```
SOQL Select Statement SELECT Contact.Email FROM OpportunityContactRole WHERE OpportunityId = '{pv0}' AND Role = 'Decision Maker'
```

- Example of Decision Makers (Contact Roles) on an Opportunity:

Action	Contact Name	Account Name	Email	Phone	Role
Edit   Del	Aaron Lopinko	SP Salesworks	alopinko@democlient.com	(555) 543-1234	Decision Maker
Edit   Del	Mike Smith	SP Salesworks	robertbovdsfcdemo@gmail.com	(555) 444-5640	Decision Maker
Edit   Del	Tina Yurchovski	SP Salesworks	tyurchovski@democlient.com	(555) 987-6543	Decision Maker

- The above SOQL query will return all three email addresses. To separate each email address with commas, it requires the QVar0Format parameter.
- Example parameter configuration:



```

1
2 &QVar2Id=a04E0000004S1a1
  &QVar2Format=10000
  &EmailCC={QVar2}
3

```

- The Conga Query ID is referenced with &QVar2Id.
- The results of the query are formatted with &QVar2Format. The parameter value of 10000 separates each data value with a comma (appropriate for multiple email addresses).
- The {QVar2} result is used as the value for the EmailCC parameter.

### Using All Three QVar Use Cases to Send an Email - Bringing It All Together

- Using all three QVars described above together creates an elegant and streamlined Composer solution:


The screenshot shows the 'Edit Email' interface. A red arrow labeled '1' points to the 'To' field containing 'Robert Boyd'. Another red arrow labeled '2' points to the 'CC' field containing a list of email addresses. A third red arrow labeled '3' points to an attachment in the 'Attachments' section titled 'Proposal for SP Salesworks - 12\_5\_2013.pdf'. The 'Subject' field contains 'Your Proposal from Demo Studios'.

- The first QVar retrieves the Account's Primary Contact and defines it as the main email recipient.
- The second QVar retrieves the Decision Maker Contact Roles and includes them as CC recipients.
- The third and final QVar retrieves the attached terms and conditions file and combines it with a proposal template as a single PDF attachment.

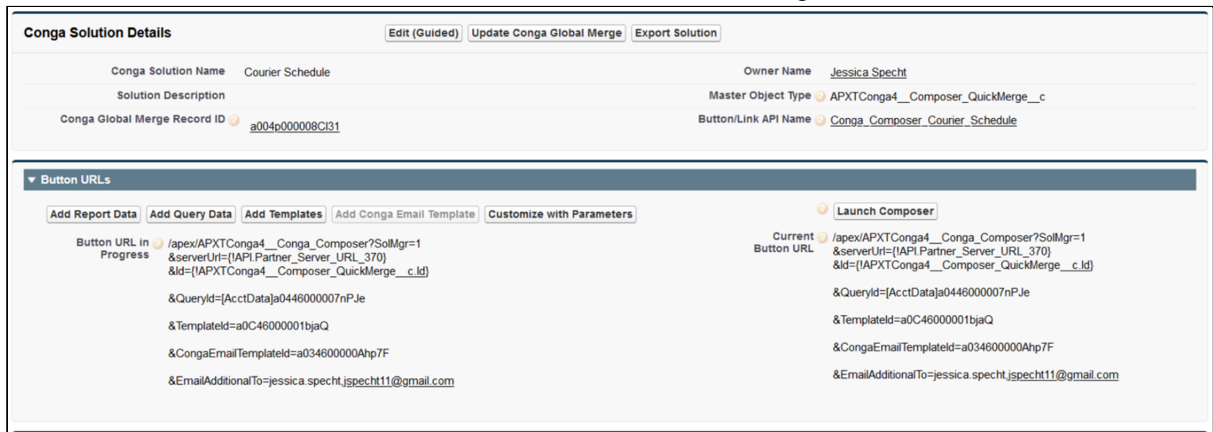
# Transitioning from Courier to a Composer and Batch Solution

To transition from a Courier to a Composer and Batch solution, take the following steps:

1. [Convert Data Gathering Reports to Conga Queries](#).
2. [Build a Conga Global Merge Solution](#).
3. Add the query you created from Step 1 to the solution and [use the template builder to build a template](#).

 We recommend using an Excel template for this step.

4. [Create a Conga email template](#) to use with Global Merge.
5. Determine from the Courier schedule who the recipients are and add those email addresses to the Global Merge Solution using the [EmailAdditionalTo](#) parameter.
6. Add the Template and Email Template to Global Merge along with any additional desired behavior parameters (for example: defaultpdf, field update, activity logging, etc). The final solution should look similar to the following screenshot:



The screenshot displays the 'Conga Solution Details' page for a solution named 'Courier Schedule'. It shows the solution's configuration, including the Conga Global Merge Record ID (a0046000008C31) and the Button/Link API Name (Conga\_Composer\_Courier\_Schedule). Below this, the 'Button URLs' section is expanded, showing two buttons: 'Add Report Data' and 'Launch Composer'. Both buttons have identical URLs with parameters for server URL, record ID, query ID, template ID, Conga email template ID, and email recipient.

Button Name	Button URL
Add Report Data	/apex/APXTConga4__Conga_Composer?SolMgr=1 &serverUrl={!API.Partner_Server_URL_370} &Id={!APXTConga4__Composer_QuickMerge_c.Id} &QueryId={AcctData}a0446000007nPJe &TemplateId=a0C46000001bjaQ &CongaEmailTemplateId=a034600000Ahp7F &EmailAdditionalTo=jessica.specht.jspecht11@gmail.com
Launch Composer	/apex/APXTConga4__Conga_Composer?SolMgr=1 &serverUrl={!API.Partner_Server_URL_370} &Id={!APXTConga4__Composer_QuickMerge_c.Id} &QueryId={AcctData}a0446000007nPJe &TemplateId=a0C46000001bjaQ &CongaEmailTemplateId=a034600000Ahp7F &EmailAdditionalTo=jessica.specht.jspecht11@gmail.com

7. [Create a Conga Batch Record](#) and use the Record Id from your Global Merge as the Master Object Source.
8. Convert your Global Merge Solution to a formula field using the [Formula Builder](#) and set Qmode to Send Email.
9. Go to **Salesforce Setup** > **Conga Global Merge** to find the field you created and set Field Level Security to visible as shown in the screenshot below.

 This step is required. Not setting Field Level Security to visible results in an error when Batch runs.

Set Field-Level Security  
Courier Schedule [Help for this Page](#)

Field Label	Courier Schedule
Data Type	Formula (Text)
<b>Field-Level Security for Profile</b>	
	<input checked="" type="checkbox"/> Visible
	Read-Only
Analytics Cloud Integration User	<input checked="" type="checkbox"/> ✓
Analytics Cloud Security User	<input checked="" type="checkbox"/> ✓
Authenticated Website	<input checked="" type="checkbox"/> ✓
Authenticated Website	<input checked="" type="checkbox"/> ✓
Contract Manager	<input checked="" type="checkbox"/> ✓
Cross Org Data Proxy User	<input checked="" type="checkbox"/> ✓
Custom: Marketing Profile	<input checked="" type="checkbox"/> ✓
Custom: Sales Profile	<input checked="" type="checkbox"/> ✓
Custom: Support Profile	<input checked="" type="checkbox"/> ✓
Customer Community Login User	<input checked="" type="checkbox"/> ✓
Customer Community Plus Login User	<input checked="" type="checkbox"/> ✓
Customer Community Plus User	<input checked="" type="checkbox"/> ✓
Customer Community User	<input checked="" type="checkbox"/> ✓
Customer Portal Manager Custom	<input checked="" type="checkbox"/> ✓
Customer Portal Manager Standard	<input checked="" type="checkbox"/> ✓

- Enable the Conga Batch Master Switch and create a Salesforce token through Conga Batch setup and then return to the Batch record to [set the schedule](#).

## Composer Integrations

Information on how to integrate Composer with other applications.

Click on the links in the list to the left for more information on how to integrate Composer with other applications.

- [Integration Partners](#)
  - Parameters for partner and Salesforce integrations.
- [Configuring Box with Conga Composer](#)
  - How to configure Box to work with Conga Composer.
- [Configuring OneSpan Sign with Conga Composer](#)
  - How to configure OneSpan Sign to work with Conga Composer.
- [Configuring Page Layouts for Use with Box](#)
  - How to configure page layouts to use with Box.
- [Creating Box Tokens](#)
  - How to create Box Tokens for Composer and its suite of products.
- [Google Drive Integration Parameters](#)
  - Conga Composer integrates with Google Drive, making it possible to create and deploy Conga Composer solutions completely in the cloud.
- [Integrating Conga Composer with Sharepoint](#)
  - How to configure Sharepoint with Conga Composer using Microsoft Flow.
- [Modify the Adobe eSign Application to Disable the Signature Order Field](#)

- After installing Adobe eSign from the AppExchange, you must make the following changes to the Adobe eSign application.
- [SendGrid Integration Parameters](#)
  - Conga Composer offers integration with SendGrid, an email delivery service. These parameters affect how Composer works with SendGrid.
- [Setup the Sertifi Integration](#)
  - How to setup the Composer Sertifi Integration.
- [Set Up the Salesforce Content Integration](#)
  - We recommend that you create a Lookup Relationship field on Salesforce Content for your Master Object type so that you can relate new Content records to the object from which Conga Composer is initiated.
- [Supported Culture Codes for Adobe Sign](#)
  - This documentation reviews supported culture codes (languages) for Adobe Sign in Salesforce.
- [Using Composer with Salesforce Shield Encryption](#)
  - Merged data encrypted with Salesforce Shield Encryption.
- [Merge and Box integration](#)
  - Using Conga Composer to save content to Box.

## Integration Partners

Parameters for partner and Salesforce integrations.

### Integration Partners

- Adobe Sign
- Box
- DocuSign
- [Google Drive](#)
- OneDrive
- [SendGrid](#)
- Sertifi
- SpringCM
- Amazon S3
- [Amazon SES](#)
- Salesforce Integrations
- Salesforce Files (Content)
- Salesforce Chatter
- Salesforce1

- Salesforce Email
- Salesforce Notes and Attachments

## Configuring Box with Conga Composer

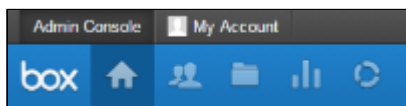
You can configure Box to work with Conga Composer by following the steps below.

To use Box in an automated fashion with Conga Composer, you will also need to [create and save a Box token](#).

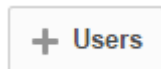
To configure Box with Conga Composer:

For proper integration, you need to send collaborator invites to users from [Box.com](#) so the Salesforce integration can properly manage permissions. See [Box.com](#) for instructions on sending collaborator invitations. You also need to [Configure Page Layouts for Use with Box](#).

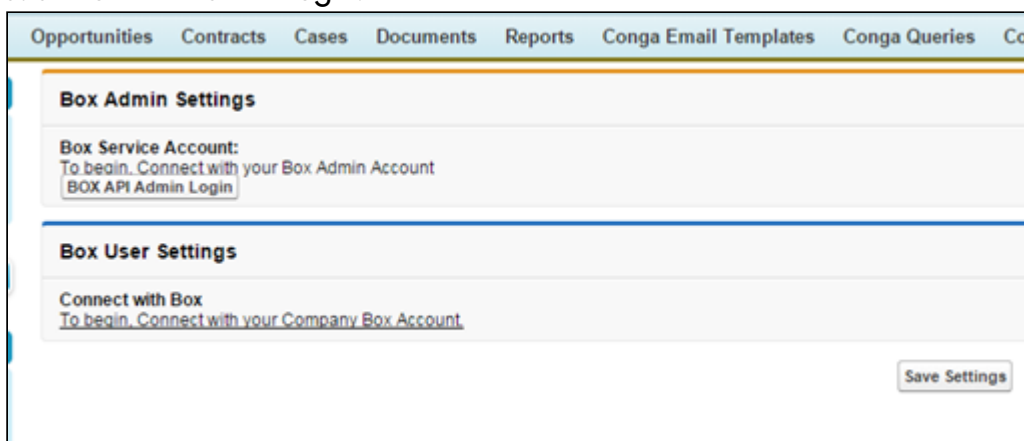
1. Log in to a [Box.com](#) account which has access to the Admin Console.
2. Click **Admin Console** in the top left.



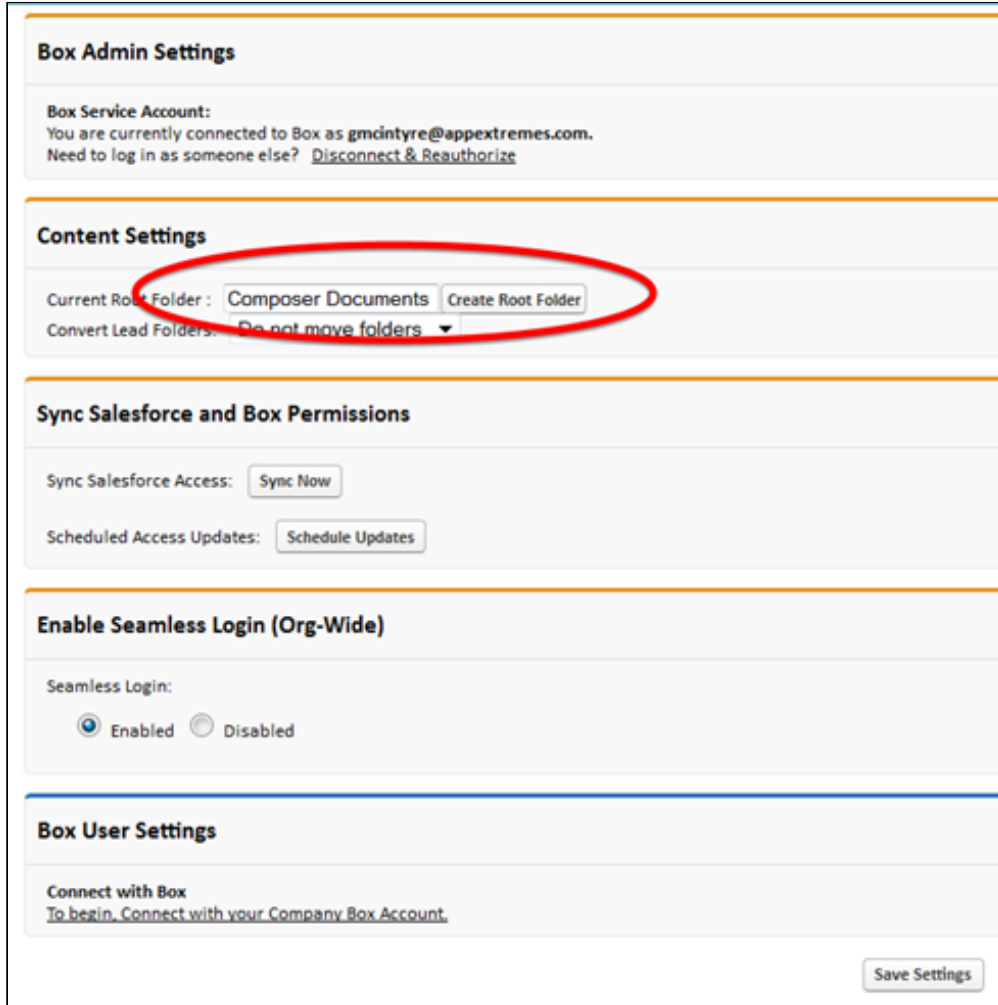
3. Click on the two heads (Users and Groups) icon.
4. Click on **+Users**.



5. Enter a Name and Email. It is essential that the email for this user and the email of the main account are different. You might want to indicate in the Name that this is the non-admin account. For example: Bobby NonAdmin.
6. Click **Add User**.
7. Install Box from the AppExchange.
8. After Box is installed, in Salesforce, click **All Tabs**, then **Box Settings**.
9. Make sure you are logged in as your Admin user. Then, on the Box Settings page, click **Box API Admin Login**.



10. On the next page, click **Grant access to Box**. (If prompted, enter the credentials for your Admin user and click Authorize.) You will be taken back to the Box Settings page.
11. Type a name (Composer Documents, for example) and click Create Root Folder to create the root folder for this organization.



12. Log out of your [Box.com](https://www.box.com) Admin account.
13. Log into your [Box.com](https://www.box.com) Non Admin Account.
14. Switch back to [salesforce.com](https://www.salesforce.com).
15. Click on **To begin, Connect with your Company Box Account**.
16. Click **Grant access to Box** in the pop-up window.
17. Click **Save Settings**.

**⚠** If your Box account does not have external links to files enabled, you cannot be redirected to the Box document after the merge. To enable this setting go to **Box Account settings** → set **Enable external links** → Select **Files Only** or **Folders and Files**.

Conga Composer cannot overwrite existing folders in your Box Interface.

## Configuring OneSpan Sign with Conga Composer

How to configure OneSpan Sign to work with Conga Composer.

OneSpan Sign by Vasco is an electronic signature service that allows you to easily and securely sign Composer documents over the internet. You can configure OneSpan Sign to work with Conga Composer.

### Pre-requisites for OneSpan Sign and Composer integration:

- Install the latest OneSpan Sign for Salesforce package from the Appexchange.
- Once it is installed, ensure that OneSpan Sign is connected by navigating to the OneSpan Sign Admin tab in Salesforce and logging in with your OneSpan Sign credentials.

### To integrate OneSpan Sign with Composer:

1. Go to <https://www.onespan.com/> and click **OneSpan Sign Login** to log in to your account.
2. Click **Admin**.
3. Click **Integration**.
4. Copy the API Key.
5. Go to the **Composer Setup tab** in Salesforce.
6. Click into the **Integration Partners → eSignature tab**.
7. Click on the **OneSpan Sign tab**.
8. Enter the API Key in the OneSpan Sign API Key field.
9. Select a OneSpan Sign Endpoint from the dropdown menu. Ensure that the endpoint URL matches the OneSpan Sign endpoint that is connected on the OneSpan Sign Admin tab.
10. Click **Save Changes**.
11. Add the following parameters to a Composer solution. For more information, see [Adding Parameters to a Solution](#).
  - eSignLiveVisible
  - eslPackageName
  - eslSignerRIID
  - DS7 (optional)

Conga Composer is now integrated with OneSpan Sign and ready for use.

## Configuring Page Layouts for Use with Box

How to configure page layouts to use with Box.

To configure page layouts for Box:

**i** Before You Begin, you must be logged into Salesforce. You must also have [Configured Conga Composer for Use with Box](#).

1. In Salesforce, navigate to and edit the page layouts for each object you wish to use with Box.
2. From the layout menu, click **Visualforce Pages**.
3. Click and drag Sections to the appropriate place on the page layout.
4. For Section Name, enter Box and select a 1-Column layout.
5. Click **OK**.

## Creating Box Tokens

How to create Box Tokens for Composer and its suite of products.

To use Box in an automated fashion (background mode in Composer, Conga Batch, or Conga Trigger), you need to create and save a Box token. To create a box token for each of the Conga Composer suite products, see the following:

- [Creating a Box Token with Conga Composer](#)
- [Creating a Box Token with Conga Batch](#)
- [Creating a Box Token with Conga Trigger](#)

## Creating a Box Token to Use with Conga Composer

How to create a Box token to use with Conga Composer, Conga Batch and Conga Trigger.

To use Box in an automated fashion (background mode in Composer, Conga Batch, or Conga Trigger), you need to create and save a Box token.

To use Box in an automated fashion with Conga Composer, you need to create and save a Box token.

1. Go to the **Composer Setup tab**.
2. Click **Integration Partners tab**.
3. Click the **Box tab**.
4. Click **Create Box Token**.



5. Enter the login details for the Box Service user (which can be found in the Box Settings tab).
6. Run your Composer solution in background mode (using &DS7=600) to automatically send the merged documents to Box.

## Creating a Box Token to Use with Conga Batch

How to create a box token to use with Conga Batch.

To use Box in an automated fashion with Conga Batch, you need to create and save a Box token by taking the following steps:

1. Go to the **Conga Batch Setup tab**.
2. Click **Create Box Token**.
3. Enter the login details for the Box Service user (which can be found in the Box Settings tab).
4. Run a batch operation to automatically send the merged documents to Box.

## Creating a Box Token to Use with Conga Trigger

How to create a box token to use with Conga Trigger.

To use Box in an automated fashion with Conga Trigger, you need to create and save a Box token by taking the following steps:

1. Go to the **Conga Trigger Setup tab**.
2. Click **Create Box Token**.
3. Enter the login details for the Box Service user (which can be found in the Box Settings tab).
4. Trigger the Workflow Rule to automatically send the merged documents to Box.

## Google Drive Integration Parameters

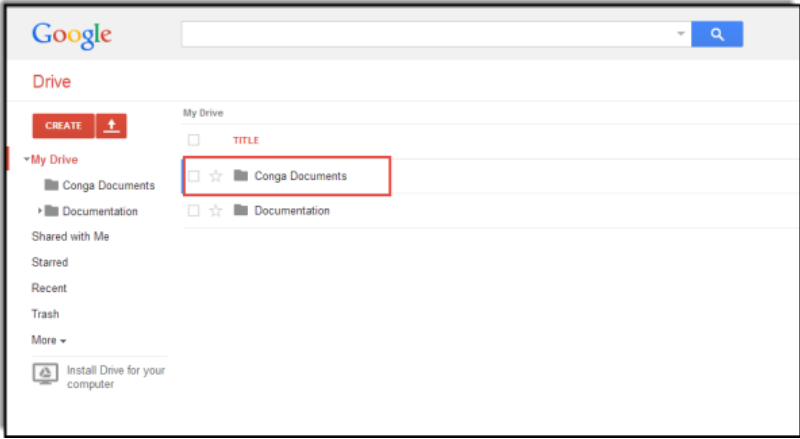

Conga Composer integrates with Google Drive, making it possible to create and deploy Conga Composer solutions completely in the cloud. With the Google Drive integration, you can:

- Create Google Docs, Google Sheets or Google Slides with Composer
- Store any generated output file (Microsoft Office, PDF, or Google Docs/Sheets/Slides) directly in Google Drive
- Create templates in Google Drive (which must then be downloaded as Microsoft Office file type before being uploaded to Salesforce to use as a Conga Template)

To use Google Drive with Conga Composer, ensure that Google Drive is deployed in your Google account. The first time you run a Conga Composer solution using the Google integration, you are prompted to login to your Google account and allow Conga to access Google Drive to store your documents.

## Google Drive Integration Parameters

Parameter	Behavior
&GoogleVisible=1	Enables Google as an output option on the Conga Composer user interface
&SC0=[0 or 1]	Enables the Save a Copy feature <ul style="list-style-type: none"> <li>• 0 = Save a Copy is disabled (default)</li> <li>• 1 = Save a Copy is enabled</li> </ul>
& SC1=Google	Defines Google Drive as the location where a copy of the merged output file will be stored.  Requires: &SC0=1
&DS7=9	Conga Composer's Background Mode is enabled and the merged document (Microsoft Office version) is saved to Google Drive
&DS7=19	Conga Composer's Background Mode is enabled and the merged document (Google Docs version) is saved to Google Drive

Parameter	Behavior
&OFP=[Path]	<p>Output File Path Specifies the file path within Google Drive where the merged output files will be saved. Each folder name should be separated with a forward slash "/"</p> <p>Requires: &amp;GoogleVisible=1 or &amp;SC1=Google</p> <p>Example: &amp;OFP=Corporate+Docs/{!Opportunity.Name}</p> <p>This example would create a folder named with the Opportunity Name nested inside a folder called "Corporate Docs"</p> <p>By default, Conga Composer will create a folder in Google Drive called Conga Documents if the OFP parameter is not used.</p>  <p> Google Team Drives is not supported with the OFP parameter.</p>

## Integrating Conga Composer with Sharepoint

How to configure Sharepoint with Conga Composer using Microsoft Flow.

Conga Composer files can be stored in Sharepoint by setting up a Flow in Microsoft Office 365 that copies all files in a folder from OneDrive for Business to a specified Sharepoint folder. OneDrive must first be integrated with Conga Composer for the Flow to work.

To integrate Composer with Sharepoint:

1. Go to the **Conga Composer Setup** tab.
2. Click the **Integration Partners** tab.
3. Click the **OneDrive** tab.

4. Click **Create OneDrive Token**.
5. Enter login credentials or select the appropriate Microsoft Office 365 account.
6. Valid Token appears next to Create One Drive Token to confirm a token was successfully generated.
7. Add the following OneDrive parameters to a Composer solution.
  - DS7
  - OneDriveVisible
  - OFP
  - QMode
  - SC1
8. Go to the [Copy files between OneDrive for Business and Sharepoint template](#) on the Microsoft Flow website.
9. Click on **Use this template to create a Flow between OneDrive and Sharepoint**.
10. Enter login credentials or select the appropriate Microsoft Office 365 account.
11. Select which folders in OneDrive and Sharepoint are used for the Flow.
12. Click **Save flow**.
13. When running the Conga Composer solution, files saved to OneDrive are now automatically copied into the specified Sharepoint folder.

## Modify the Adobe eSign Application to Disable the Signature Order Field

After installing Adobe eSign from the AppExchange, you must make the following changes to the Adobe eSign application:

To disable the Signature Order field in Adobe eSign:

 Before you begin, you must first download Adobe eSign from the AppExchange.

1. Navigate to the **Salesforce Custom Objects for Adobe Document Cloud eSign Services** and click **Edit**.
2. In the Signatures area, click **Edit**.
3. In the Picklist Values area, click **Edit** to edit the Recipient Signs, then I sign picklist.
4. In the Signature Order field, select **Recipient signs, then I sign**.
5. Clear the Default box: Make this value the default for the master picklist.
6. Click **Save**.

## SendGrid Integration Parameters

Conga Composer offers integration with SendGrid, an email delivery service. These parameters affect how Composer works with SendGrid.

### Background

SendGrid is an email delivery service with features that assure customers an extremely high deliverability rate at a low cost. Delivering emails via SendGrid means customers can circumvent Salesforce email limits. SendGrid is available separately from their [website](#). You must subscribe to that service to use this integration.

### Setup the SendGrid Integration

1. Subscribe to a plan on the SendGrid website.
2. Configure your SendGrid account according to their instructions.
3. Open the Conga Setup page (located on the lower-right corner of the Conga Composer dialog box).
4. On the Integration Partners tab, enter your SendGrid API Key. For more information see [Entering a SendGrid API Key](#).
5. Save your changes.

Once Conga is configured with your SendGrid credentials, all email generated through Conga will be delivered via SendGrid. Conga Composer will also automatically create a completed Task record for each email sent through SendGrid.

From the Conga Composer dialog box, when the Email button is clicked, Conga Composer generates the output document (as usual) and then presents an editable email, pre-addressed to the email address configured in the Conga Composer button.

To enable the use of merge fields in your email templates use a Conga Email Template OR configure a SendGrid Template ([from their website](#)). Do not use Salesforce email templates, merge fields are not compatible with Salesforce email templates and SendGrid.

#### Parameters

- Conga Composer's Email Parameters and Conga Email Templates work with SendGrid.
- DS7 enables merge operations to occur in the background.
- LGAttachOption sets the Salesforce object to which attachments generated during Activity Logging are stored.
- SC1 defines where a copy of the merged output file will be stored.

## Entering a SendGrid API Key

You can integrate Conga Composer and Twilio SendGrid through the use of an API Key. First, you must obtain an API Key through Twilio SendGrid, and then you must enter the key you generated in Conga Composer.

### **Warning:**

You must update your authentication method with Twilio SendGrid to API keys exclusively before SendGrid deactivates the functionality on December 9th, 2020 in order to ensure uninterrupted service and improve the security of your account

## Obtaining a SendGrid API Key

- Log in to your Twilio SendGrid account.
- Follow the steps in Twilio SendGrid's documentation for [Creating an API Key](#).
- Use the API Key you created for the next steps taken in Conga Composer.

## Entering a SendGrid API Key in Conga Composer

To enter your Twilio SendGrid API Key, take the following steps:

1. Go to **Conga Composer Setup**.
2. Under Integration Partners, click **Email > SendGrid**.
3. Enter the SendGrid API Key you created in the SendGrid API Key field, and click **Save Credentials**.

**SendGrid**

Conga apps seamlessly integrate with SendGrid\* to provide a scalable and reliable email deliverability solution that goes beyond the Salesforce email limits. SendGrid is available as a separate application.

SendGrid is available separately from their [website](#). You must subscribe to that service to use this integration. For more information on the Composer integration to SendGrid, visit our [Knowledge Base](#).

**Reliable Delivery of Important Documents:**

- SendGrid handles all the worries associated with email delivery rates, spam folders, ISP blacklists and scaling to meet high user volume

**SendGrid Settings:**

Please supply your SendGrid credentials for Conga to use to deliver email. The credentials you enter are protected with the Advanced Encryption Standard (AES) block cipher, a widely adopted encryption standard.

Using these values, Conga will deliver email through SendGrid.

\* SendGrid API Key:

[Remove Credentials](#) [Save Credentials](#)

## Setup the Sertifi Integration

How to setup the Composer Sertifi Integration.

**Pre-requisite:** Install Sertifi for Salesforce from the [AppExchange](#).

Create a New Sertifi Account

1. Navigate to the **Sertifi Getting Started tab**.
2. Input a value for the Account Title field.
3. Input a value for the Account URL field.
4. Copy your API Code – you will need it in the next step.
5. Navigate to **Setup → Develop → Custom Settings**.
6. Find the Sertifi Application Settings and click **Manage**.
7. Edit the custom setting called Settings.
8. Find the APICode field and paste your API Code from Step 4.
9. Click **Save**.

## Integrate with Composer

After your Sertifi account is setup, you are ready to integrate it with your Composer solution. This is done using Sertifi Integration parameters.

For example, if you want to specify the recipient of the merged document to be delivered via Sertifi, you would add the `&SertifiSignIEmail` parameter to the Composer button URL, as shown below:

```
/apex/APXTConga4__Conga_Composer?id={!Opportunity.Id}&SertifiSigner1Email={!Contact.Email}
```

If not working correctly, try removing the plus sign from the email address (example: [robeaston+alias@gmail.com](mailto:robeaston+alias@gmail.com))

## Set Up the Salesforce Content Integration

We recommend that you create a Lookup Relationship field on Salesforce Content for your Master Object type so that you can relate new Content records to the object from which Conga Composer is initiated.

To set up Salesforce content integration

1. In Salesforce, navigate to the Salesforce CRM Content screen.
2. Create a new Lookup Relationship field for the type of field that calls Conga Composer.
3. Save your changes.

Refer to this field using the [ContentCustomFieldName](#) and [ContentCustomFieldValue](#) parameters. You must also specify a [ContentWorkspaceld](#) via that parameter.

## Supported Culture Codes for Adobe Sign

This documentation reviews supported culture codes (languages) for Adobe Sign in Salesforce. Adobe Sign supports the languages listed in the table below. Reference a language's corresponding Culture Code as the value for the ESLanguage parameter in a Composer solution to set the Adobe Sign language field.

For more information on Adobe Sign supported languages, see [System requirements for Adobe Sign](#).

ENGLISH NAME	ABBREVIATION
For all supported locales as well as examples of display formats for each, visit <a href="https://help.salesforce.com/apex/HTViewHelpDoc?id=admin_supported_locales.htm">https://help.salesforce.com/apex/HTViewHelpDoc?id=admin_supported_locales.htm</a>	
English (United States)	en-US
English (United Kingdom)	en-GB
French (France)	fr-FR



ENGLISH NAME	ABBREVIATION
German (Germany)	de-DE
Spanish (Spain)	es-ES
Catalan (Catalan)	ca-ES
Basque (Basque)	eu-ES
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-PT
Japanese (Japan)	kn-IN
Chinese (China)	zh-CH
Korean	ko-KR
Indonesian (Indonesia)	id-ID
Malay (Malaysia)	ms-MY
Thai (Thailand)	th-TH
Vietnamese (Vietnam)	vi-VN
Croatian (Croatia)	hr-HR
Czech (Czech Republic)	cs-CZ
Polish (Poland)	pl-PL
Russian (Russia)	ru-RU
Slovenian (Slovenia)	sl-SI
Turkish (Turkey)	tr-TR
Ukrainian (Ukraine)	uk-UA

ENGLISH NAME	ABBREVIATION
Danish (Denmark)	da-DK
Dutch (Netherlands)	nl-NL
Finnish (Finland)	fi-FI
Hungarian (Hungary)	hu-HU
Icelandic (Iceland)	is-IS
Italian (Italy)	it-IT
Norwegian (Norway)	no_NO
Swedish (Sweden)	sv-SE

## Using Composer with Salesforce Shield Encryption

Merged data encrypted with Salesforce Shield Encryption.

### Using Composer with Salesforce Shield Encryption

Conga Composer users can successfully merge data in Salesforce orgs with Salesforce Shield Encryption enabled. To merge fields encrypted with Salesforce Shield Encryption, create a new View Encrypted Data Permission Set and assign it to user, or assign the View Encrypted Data system permission to User Profiles.

The recommended method is to create and assign a Permission Set with the View Encrypted Data permission enabled.

#### Create and Assign a View Encrypted Data Permission Set

1. Navigate to **Salesforce Setup**.
2. In the Quick Find search box, type in Permission Sets.
3. Click **Permission Sets**.
4. Click **New** to create a new Permission Set
5. Name the Label View Encrypted Data and the API Name View\_Encrypted\_Data.
6. Click **Save**.
7. Under the System Permission section, click System Permissions.

8. Click **Edit**.
9. Check the View Encrypted Data checkbox
10. Click **Save**.
11. Click **Manage Assignments**.
12. Click **Add Assignments**.
13. Select all users that need to merge encrypted data with the checkbox field.
14. Click **Assign**.

### Assign the View Encrypted Data Permission

 You can only assign the View Encrypted Data permission to a custom profile.

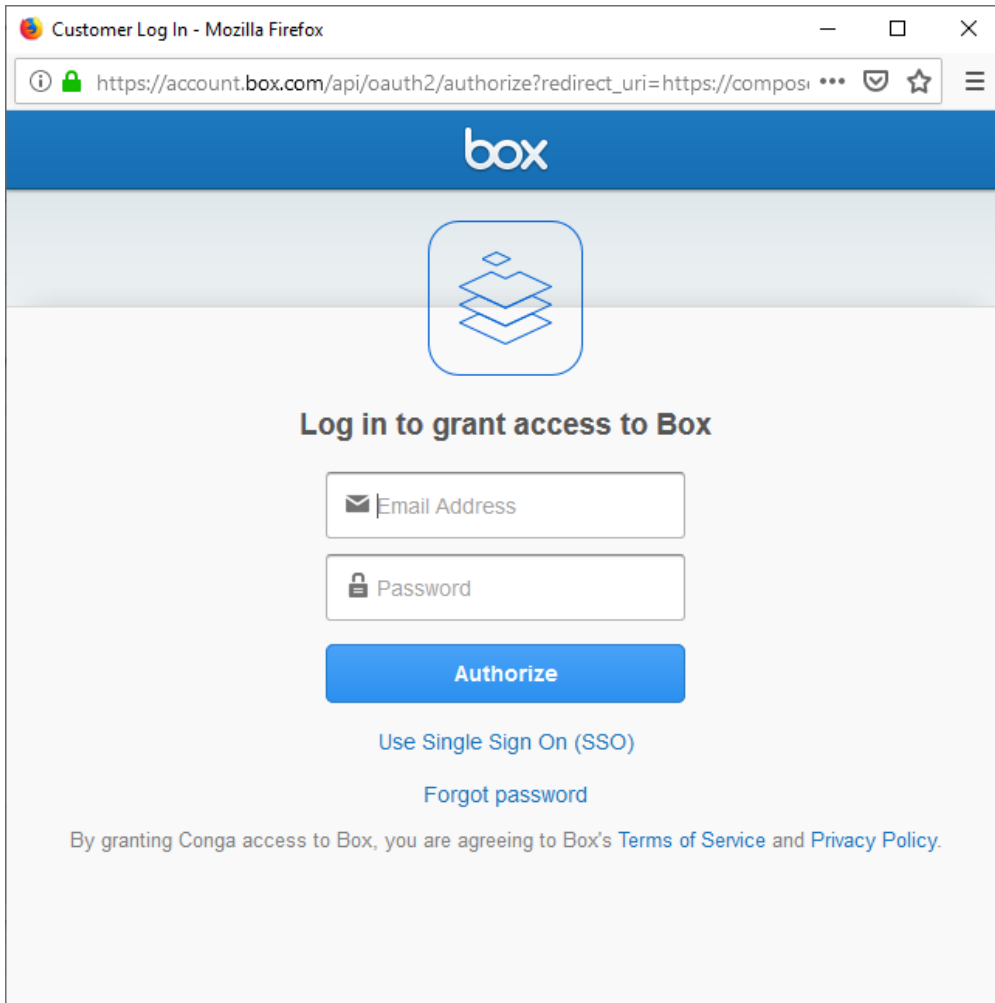
1. Navigate to **Salesforce Setup**.
2. In the Quick Find search box, type in Profiles
3. Click **Profiles**.
4. Select the User Profile that needs to merge encrypted data.
5. Click **System Permissions**.
6. Click **Edit**.
7. Check the View Encrypted Data checkbox.
8. Click **Save**.

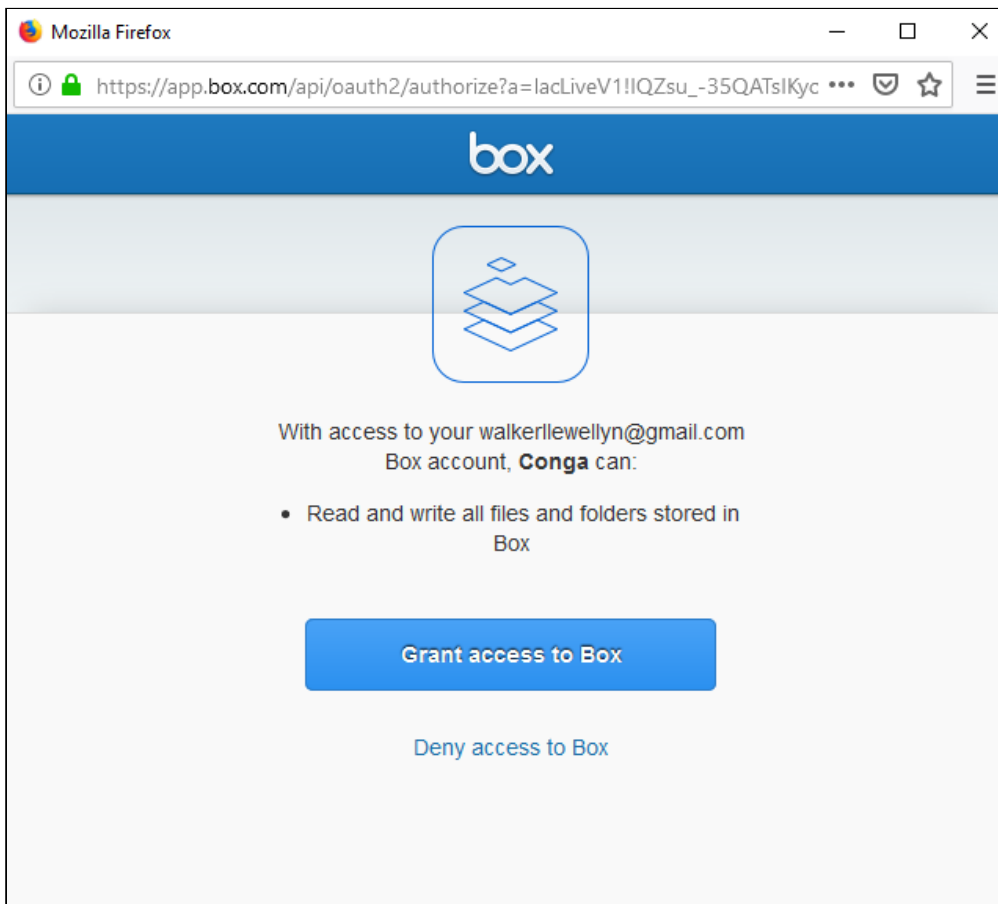
## Merge and Box integration

Using Conga Composer to save content to Box.

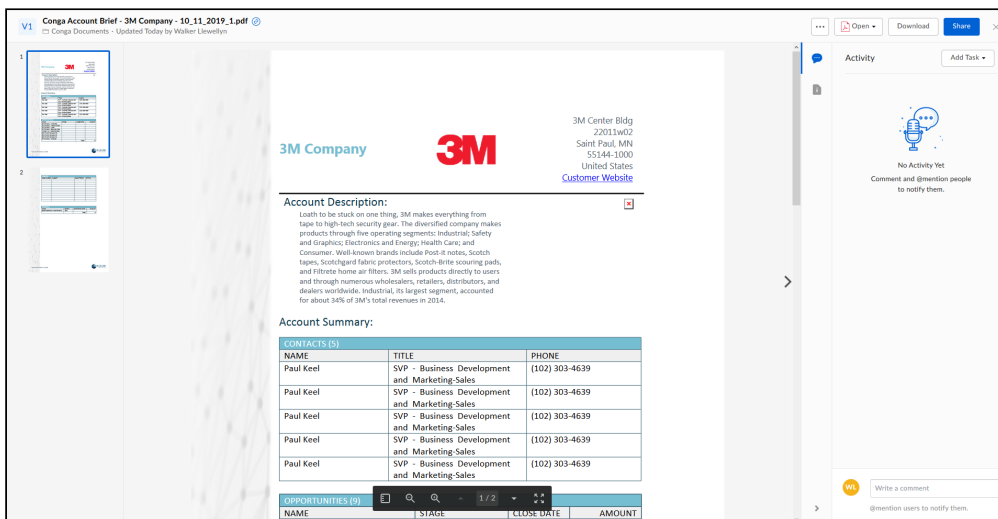
When creating the button URL for use with a Box solution, it is critical that the OFP parameter specifies the same path as the Box for Salesforce integration. This file path is case sensitive in the OFP parameter. If these do not match, the Salesforce permissions are not properly applied. For more information on parameters, see [Merge and Box integration](#).

Clicking **Merge & Box** presents you with standard OAuth screen. If you have not yet logged in to Box, you are prompted with a login screen followed by an Authorize screen.





When the merge process completes, you are directed to the Box preview screen.




Closing the preview screen takes you to the record page.

## Amazon SES Integration

Conga provides integration to send emails with Conga Composer utilizing a customer's Amazon SES (Simple Email Service).


Amazon SES provides users with additional security features such as filtering technologies, dedicated IP addresses, and a reputation dashboard. For more detailed information Amazon SES, see [Amazon Simple Email Service](#).

 You must have an active Amazon SES account to utilize the Amazon SES integration with Conga Composer.

To set up the Conga Composer integration with Amazon SES, see [Setting Up Amazon SES Integration for Conga Composer](#).


## Setting Up Amazon SES Integration for Conga Composer

Set up Amazon SES integration in Conga Composer Setup.

 You must have an active [Amazon SES](#) account to utilize the integration with Conga Composer.

To set up Amazon SES integration for Conga Composer:

1. Navigate to Conga Composer Setup.
2. Select **Email** under the Integration Partners section.
3. Click the Amazon Web Services SES logo to access the configuration for Amazon SES.
4. Complete the following fields with the credentials and information from your Amazon SES account.

-  • The information and credentials are located in the Using SMTP to Send Email with Amazon SES page in your Amazon SES account. If you have not created the credentials yet, use the Create in My SMTP Credentials on that page to create your credentials.
- Amazon SES Integration for Conga Composer only supports SMTP Port 587.

- a. AWS SES Username
- b. AWS SES Password
- c. AWS SES Address
- d. AWS SES Port

- e. (Optional) AWS SES Config Set
- 5. Click **Save** after completing the fields above.
- 6. Click the Amazon Web Services SES toggle to change the value from Disabled to Enabled.

After entering the necessary credentials and enabling Amazon SES in Conga Composer Setup, Conga Composer will send all emails with Amazon SES.

- ✓ Before sending an email using Amazon SES, Conga recommends validating a domain or email address that you use to send or receive emails. For more information on how to validate your Amazon SES configuration, see [Sending test emails in Amazon SES](#).

## Global Reporting with Conga Composer

Information about Global Reporting with Composer.

- [Adding a Global Merge Custom Link to Lightning Home Page](#)
  - How to add a Global Merge link to a Lightning home page.
- [Create a Dashboard Report using Conga Composer and Conga QuickMerge](#)
  - Step by Step Process on how to create a Dashboard Report using Conga Composer and Conga QuickMerge.
- [Creating a Home Page Link for Conga Composer in Salesforce Classic](#)
  - Conga Composer can be launched from a custom link on the Home page.
- [Global Merge in Solution Manager](#)
  - In Composer 8.7 and newer, the QuickMerge global reporting functionality has been built into the Global Merge feature of Solution Manager.
- [How to Include Dashboard Graphics in a Conga Composer Merge Document](#)
  - You can include an image from a dashboard in a Conga Composer merge document by using an IMAGE: field.

## Adding a Global Merge Custom Link to Lightning Home Page

How to add a Global Merge link to a Lightning home page.

- ⚠ The integration of Conga with Visualforce, Apex, Javascript (or other programming languages), and custom code is not supported.

How to add a Global Merge link to a Lightning homepage:

1. Take any current Composer solution URL that's been built for a Custom Link and remove any spaces or carriage returns so it's on a single line.

The initial URL:

```
/apex/APXTConga4__Conga_Composer
?SolMgr=1
&serverUrl={!$Api.Partner_Server_URL_370}
&Id={!&APXTConga4__Composer_QuickMerge__c.Id}
&QueryId=[ClosedOpps]a1w1a000000Ooo8,[NegotiateOpps]a1w1a000000Oool,
[ProposalOpps]a1w1a000000OooN,[CloseLostOpps]a1w1a000000OooS,[OppInfo]a1w1a000000Ooom
&TemplatId=a241a000000shZh
&DS7=13
```

Becomes this URL:

```
/apex/APXTConga4__Conga_Composer
?SolMgr=1&serverUrl={!$Api.Partner_Server_URL_370}
&Id={!&APXTConga4__Composer_QuickMerge__c.Id}
&QueryId=[ClosedOpps]a1w1a000000Ooo8,[NegotiateOpps]a1w1a000000Oool,
[ProposalOpps]a1w1a000000OooN,[CloseLostOpps]a1w1a000000OooS,[OppInfo]a1w1a000000Ooom
&TemplatId=a241a000000shZh
&DS7=13
```

2. Remove the merge field from the Id parameter's value and replace it with a hard-coded Global Merge record Id

The URL below

```
/apex/APXTConga4__Conga_Composer
?SolMgr=1&serverUrl={!$Api.Partner_Server_URL_370}
&Id={!&APXTConga4__Composer_QuickMerge__c.Id}
&QueryId=[ClosedOpps]a1w1a000000Ooo8,[NegotiateOpps]a1w1a000000Oool,
[ProposalOpps]a1w1a000000OooN,[CloseLostOpps]a1w1a000000OooS,[OppInfo]a1w1a000000Ooom
&TemplatId=a241a000000shZh
&DS7=13
```

Becomes this URL:

```
/apex/APXTConga4__Conga_Composer
?SolMgr=1&serverUrl={!$Api.Partner_Server_URL_370}
&Id=a061U000000Bs9A
&QueryId=[ClosedOpps]a1w1a000000Ooo8,[NegotiateOpps]a1w1a000000Oool,
[ProposalOpps]a1w1a000000OooN,[CloseLostOpps]a1w1a000000OooS,[OppInfo]a1w1a000000Ooom
&TemplatId=a241a000000shZh
&DS7=13
```

3. Anywhere the link is referencing a System field like `{!user.Id}` above will need to be updated to have a dollar sign following the exclamation point like this `{!$user.Id}`.



- Once your URL is doctored up you can throw it into this snippet for the Visualforce page:

```
<apex:page showHeader="false" sidebar="false"
lightningStylesheets="true"> <div>
<apex:outputLink value="YOUR URL GOES HERE" id="WHATEVER MAKES SENSE">THE DISPLAY NAME OF
THE LINK</apex:outputLink>
</div></apex:page>
```

So your final snippet would look like this:

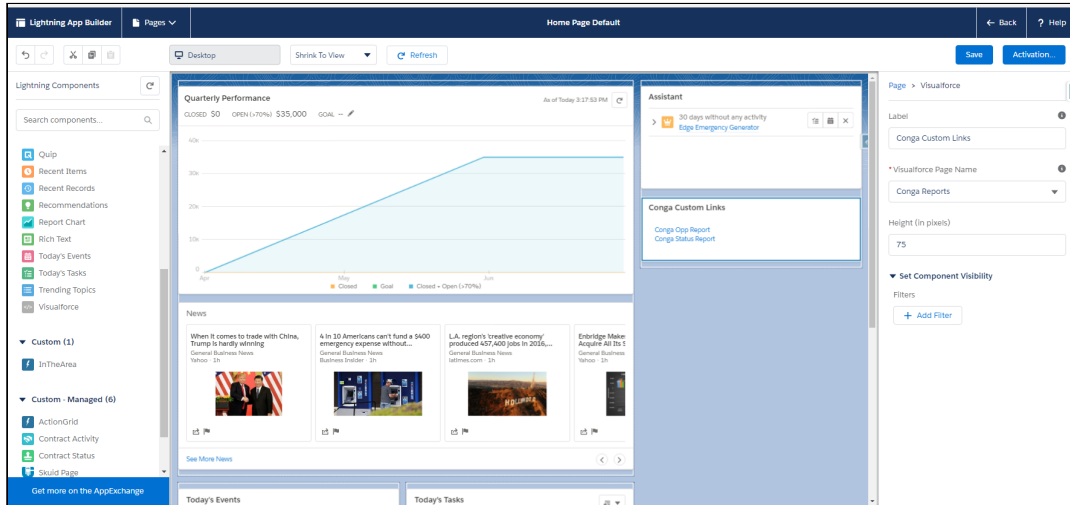
```
<apex:page showHeader="false" sidebar="false"
lightningStylesheets="true"> <div>
<apex:outputLink value = "/apex/APXTConga4__Conga_Composer?id={!$User.Id}
&QueryId=[ClosedOpps]a1w1a000000Ooo8, [NegotiateOpps]a1w1a000000Oool,
[ProposalOpps]a1w1a000000OooN,[CloseLostOpps]a1w1a000000OooS,
[OppInfo]a1w1a000000Ooom&TemplatelD=a241a000000shZh&DS7=3" id="oppReport">
Conga Opp Report</apex:outputLink> </div></apex:page>
```

- You could wrap any other Custom Links in their own <div> tags to get multiple links on the same page.
- Now copy and paste the whole snippet into the body of the VisualForce page, ensuring 'Available for Lightning Experience, Lightning Communities, and the mobile app' is checked TRUE. Your page should look something like this:

The screenshot shows the 'Visualforce Page' editor. The 'Page Information' section is visible, with 'Label' set to 'Conga Reports' and 'Name' set to 'Conga\_Reports'. The 'Available for Lightning Experience, Lightning Communities, and the mobile app' checkbox is checked. Below this, the 'Visualforce Markup' section shows the XML code for the page, including the apex:outputLink components.

- Once you have your page created it's time to add it to the homepage! Navigate to your Lightning Home Page click the **Gear Icon** → **Edit Page**.
- From the Lightning App Builder menu, select a Visualforce and drag and drop it onto the page wherever you would like the links to appear.
- Give the component a Label, select the Visualforce page you just created and set the Height (75 is ideal if you only have one or two links).

## 10. Save your page, activate it and assign to profiles and you're done!



# Create a Dashboard Report using Conga Composer and Conga QuickMerge

Step by Step Process on how to create a Dashboard Report using Conga Composer and Conga QuickMerge.

Customers on Conga Composer Release 8 can use Global Merge to create Dashboard Reports.

1. Create a custom Conga Composer home page link.
2. Create a QuickMerge link and customize the page.
3. Create reports to gather the Data and Image URLs.
4. Update the home page link to tie it all together.
5. Click **Setup** → **Customize** → **Home** → **Custom Links**. Click **New** for a New Custom Link.
6. Name your link then paste this into the Button or Link URL section and save:

```
https://composer.congamerge.com?SessionId={!API.Session_ID}
&ServerURL={!API.Partner_Server_URL_220}&Id={!User.Id}
```

**i** We used the User.Id for this home page link. Don't worry - you'll get all the data you request in your report. We use the User.Id so that this solution is not tied to one object. It really has no bearing on your data.

7. Now create a Home Page Component (if you wish or you can add the link to an existing Component).
8. After you create your Home page component, add the Dashboard Report link to show.

9. Next add your new component to your layout. Click **Home Page Layouts** and check the box next to your new component.
10. On the next page you can order your Components. This is how they will show on your Home page.
11. Click **Save**. You now have a basic Composer Home page link, and a new Component that is available on the home page. The next step is to create your QuickMerge link.

To create a QuickMerge link:

1. Click **All Tabs** → **QuickMerge Links** → **New**.
2. Select **PointMerge**.
3. Give the QuickMerge Link a name and description.
4. Click **Save**.

We now need to customize the QuickMerge layout. We need to create a few custom fields to access your images for the dashboard.

To customize the QuickMerge layout:

1. Go to **Setup** → **Create** → **Objects** → **QuickMerge Link**.
2. Under the Custom Fields & Relationships section, click **New** and create 6 fields (or however many you need for your solution). 3 Text fields for the name of the Dashboard and 3 Long Text Area fields that will hold the URL of the Dashboard image. I call these:
  - a. Image 1 Title
  - b. Image 1 URL
  - c. Image 2 Title
  - d. Image 2 URL (and so on)

Now that your fields have been created, add a section to your QuickMerge Link object and add the new fields to it.

To add a section and fields to the QuickMerge Link object:

1. Click **Edit Layout** on the QuickMerge object.
2. Add a section to your layout and give it a name.
3. Move all your new Image Title and URL fields to this section.
4. Click **Save**.

Now we need to add the Image URL from each of your dashboards to the QuickMege Page:

1. Go to your dashboard, right click and select **Copy Image Location** (or the relevant selection depending on which browser you are using).
2. Now paste this URL into the correct Image field in QuickMerge and repeat for the rest of your dashboards.

Now you could simply merge the dashboard images into a simple Word document. To customize this even further, we can add reports from these dashboards to show any specific fields and the relevant data just as you would with any Conga Composer Button/Link.

1. Click on a dashboard.
2. Clone this report, remove all fields, and add in just the fields you need in your output.
3. Next customize the "Home Page Link" and add in any specific parameters for your solution. See the [Conga Composer Parameter Guide](#) for a complete list.

✔ **Make sure you add in the OCNR parameter so your solution doesn't refresh to the user record.**

You'll need to add "?pv0=" to the end of each of your report Id's in your button so that Conga engine doesn't filter for a specific record.

This solution is showing how to merge this solution from a Home Page Link. You could instead use the QuickMerge record to execute your solution.

## Creating a Home Page Link for Conga Composer in Salesforce Classic


Conga Composer can be launched from a custom link on the Home page.

✔ **This works well for Conga Composer version 7. For Composer version 8, try using Solution Manager.**

To create a custom link on the homepage:

1. Native to **Setup** → **Customize** → **Home** → **Custom Links**.
2. Click **New** to create a new Custom Link. Label it according to the solution you're running (ex: Weekly Sales Report).
3. Choose **Display in new window** for the Behavior.
4. Select **URL** for the Content Source.
5. Enter the URL for your custom link, including parameters to customize the behavior and data retrieved.
6. Enter the Conga Composer base URL. This may be copied from any Conga Composer button or link.
7. Change the value for your &Id= parameter {!user.Id} merge field.

Running Conga Composer from a Home Page Link results in no Master Object, so we use the User ID of the running user as the Master Object instead.

 Reference reports or queries in your home page link, but be aware of what values you're passing to the filters on each. The addition of "?pv0=" after each report or query ID prevents Conga Composer from passing any values to the report and query filters.

The &OCNR parameter is very useful for home page links. If this parameter is not included, the user is redirected to the master object record when the Composer window is closed. For home page links, the Master Object is the User detail page (see number 7 above for further information). Using &OCNR=1 will override that behavior and prevents the redirect so that the user will remain on the Salesforce Home page after the merge process runs.

8. Click **Save**.

Ensure that the correct users have access to the Home page layout that houses your new Conga Composer custom link.


## Global Merge in Solution Manager

In Composer 8.7 and newer, the QuickMerge global reporting functionality has been built into the Global Merge feature of Solution Manager for Conga Composer.

To read more about how to perform org-wide reporting with Conga Composer, please visit [Global Merge](#).

## How to Include Dashboard Graphics in a Conga Composer Merge Document

You can include an image from a dashboard in a Conga Composer merge document by using an `IMAGE:` field.

 To merge Dashboard graphics in Salesforce Lightning, you must complete the steps below in Salesforce Classic and then merge the Classic Dashboard element with an [OAuth button](#) in Lightning. Dashboard images merge in PPTX templates if the `photoURL` field is on the master object only. Additionally, only internal users can see these images due to the nature of the dashboard images in Salesforce (which are formula fields, not hard-coded image files).

You can include an image from a dashboard in a Conga Composer merge document by using an `IMAGE:` field.

IMAGE special merge fields are supported in Word and PPTX templates. (The instructions below are for Firefox, which is easier to use for accomplishing this task).

1. Open the Dashboard. Right-click on the dashboard metric that you want to use. Select **Copy Image Location**.
2. Create a long text area field in Salesforce in which to store this value.
3. The value you copied in step 1 can now be pasted into this long text area field on a record.
4. Use an IMAGE: merge field to grab the value of the URL.

## Conga Composer and DocuSign for Salesforce Integration

Information about integrating Composer and DocuSign for Salesforce. For a complete list of Composer and DocuSign functionality, see DocuSign Integration Parameters.

- [Overview of Composer and DocuSign for Salesforce](#)
- [Installing DocuSign for Salesforce](#)
- [Setting Up DocuSign for Salesforce](#)
- [Using Conga Composer and DocuSign for Salesforce](#)
- [Using Conga Composer with DocuSign In Background Mode for Multiple Signers](#)
- [Use Custom DocuSign Tags to Writeback to Salesforce](#)
- [In-Person Signing with Conga and DocuSign](#)
- [Integrating Composer with DocuSign for Salesforce 7](#)
- [Integrating Composer with DocuSign App Launcher Package](#)

## Overview of Composer and DocuSign for Salesforce


DocuSign for Salesforce is an eSignature solution that integrates with Salesforce. When you have integrated DocuSign and Conga Composer you can use Conga Composer to gather information from any object in Salesforce.

[Read more about DocuSign for Salesforce.](#)

## Installing DocuSign for Salesforce

These instructions are provided as a courtesy to Conga customers. For the most up-to-date directions on installing DocuSign for Salesforce, visit [DocuSign's site](#).

1. Visit DocuSign for Salesforce's page on the [Salesforce appexchange](#) and click **Get It Now**.
2. Log into the AppExchange with your Salesforce username and password.
3. Choose to either **Install in production** or **Install in sandbox** as appropriate.
4. Select the **I have read and agree to the terms and conditions** box.
5. Click **Confirm and Install**.
6. Log into your Salesforce account.
7. Select **Install for All Users**.

 This does not grant DocuSign licenses to all users, it simply installs the necessary Apex classes, VisualForce pages, and field-level permissions.

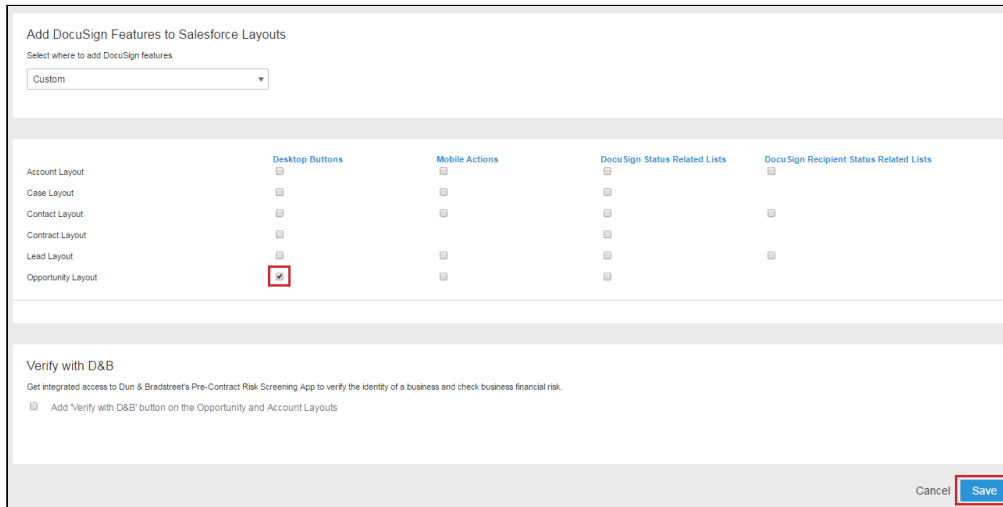
8. Click **Install**.
9. Select the **Yes, grant access to these third-party web sites** box.
10. Click **Continue**.
11. The app will begin to install.
12. You will likely receive a notification that the app is taking a long time to install.
13. You can click **Done** or wait to receive an email.
14. From the Installed Packages page, click **Configure**.
15. Enter the email address and password for your DocuSign account.
16. Select the appropriate DocuSign Environment. In most cases, this will be Production.
17. Click **Login**.
18. You will then be prompted to enter your Salesforce username and password.
19. Select the appropriate Salesforce Environment.
20. Select the check box by I agree to add the remote site and network access for communication with DocuSign.
21. Click **Connect**.

At this point, the Admin who completed the above steps is assigned a DocuSign license and is ready to use DocuSign. Other users who need to send DocuSign Envelopes through Conga Composer need to be setup according to the steps on the [Setting Up DocuSign for Salesforce](#) article.

To test your installation, do the following:

1. Click **Layouts**.
2. Select the check box for **Desktop Buttons** on an **Opportunity Layout**.

3. Click **Save**.



4. You will receive the following message.



5. Navigate to the Opportunity.

6. Click the button on the layout and see if you can get to the DocuSign envelope page. If you do, success! If not, contact DocuSign.

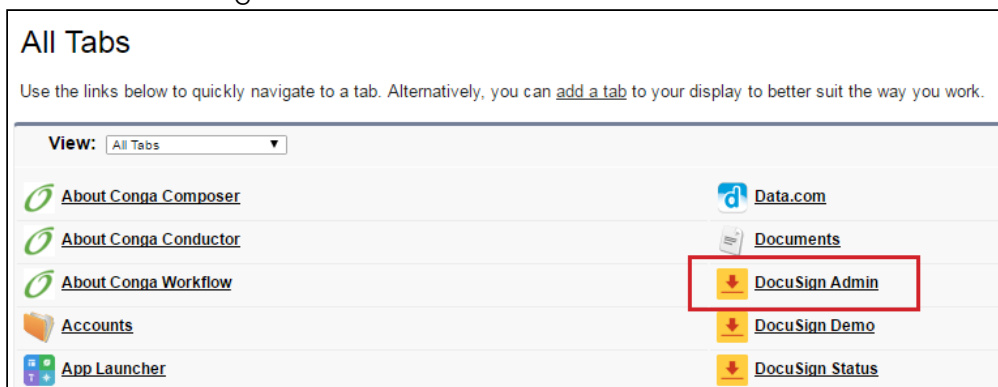
## Setting Up DocuSign for Salesforce

How to setup DocuSign for Salesforce.

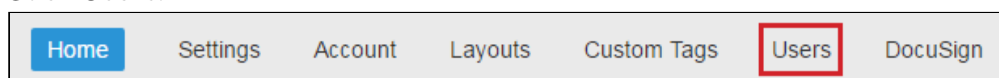
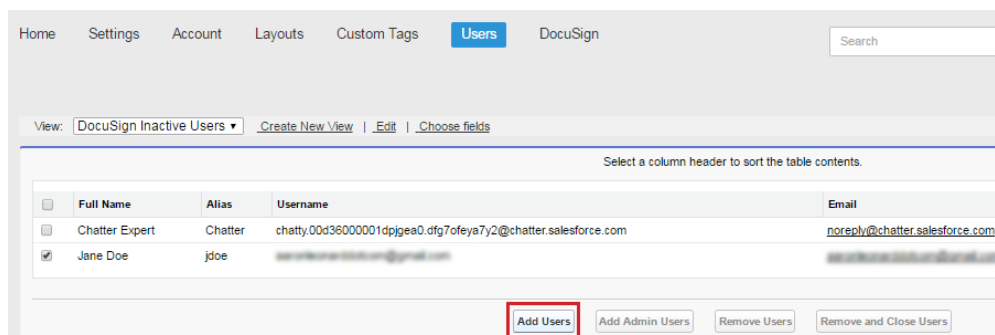
You can integrate Conga Composer with DocuSign for Salesforce. The configuration steps for integrating with DocuSign for Salesforce 7 are different. For more information, see [Integrating Composer with DocuSign for Salesforce 7](#) for more information.

To set up DocuSign for Salesforce:

1. Go to the DocuSign Admin tab.





2. Click **Users**.3. Select users and click **Add Users**. This will assign a DocuSign license to the defined users.

You are now ready to integrate DocuSign for Salesforce and Conga Composer!

## Using Conga Composer and DocuSign for Salesforce

How to create a Composer and DocuSign for Salesforce solution.

1. Create a custom button to integrate Conga Composer with DocuSign for Salesforce.
  - Create the button manually
  - Create the button with Solution Manager (Recommended)
2. Create your Conga Composer template with all the necessary merge fields.
3. Enter **automatic anchor text**, `\s1\`, where you want the tag information for the first signer to appear.
4. Color that tag information white so it does not appear in the document.
5. In Salesforce, store the document as a new Conga Template record to make it available for merging.
6. Add the following parameters to your button URL:
  - &Template ID= The ID of the Conga Template record you created in steps 2-4.
  - &DocuSignVisible=1
  - &DocuSignR1Id=[The record ID of a Salesforce Contact, Lead, or User. This can be done with Salesforce fields, typically a lookup or formula field, with Contact Roles, or with more advanced options].
  - &DocuSignR1Type=Signer
  - &DocuSignR1Role=Signer+1

The above parameters represent a single-signer solution. If you need multiple signers, see the DocuSign Integration Parameters article for additional information.
7. Add the button to the Page Layout.
8. Click your button. When working in the Salesforce Console, if the DocuSign window doesn't open, ensure the Behavior of the button is set to Display in the existing window with Sidebar and the Content Source is set to URL.

9. Verify that your template was selected in the Template List and that DocuSign is the Action under Output Options is DocuSign.
10. Click **Merge & DocuSign**.
11. In the new envelope window, you will see that the document you selected has been merged and selected as the document for signing. In Recipients and Routing, the selected contact is set to be the first recipient.
12. Click **Send Now** to send your document to the signer.

## Using Conga Composer with DocuSign In Background Mode for Multiple Signers

How to use Composer and DocuSign for Salesforce in background mode with multiple signers.

1. Create a custom button to integrate Conga Composer with DocuSign for Salesforce.
  - Create the button manually
  - Create the button with Solution Manager (Recommended)
2. Create your Conga Composer template with all the necessary merge fields.
3. Enter **automatic anchor text**, `\s1\`, where you want the tag information for the first signer to appear. Use `\s2\` where you want the second signer to sign.
4. Color that tag information white so it does not appear in the document.
5. In Salesforce, we need to store the document as a new Conga Template record to make it available for merging.
6. Add the following parameters to your button URL:

`&TemplateID=[The template ID of the document you uploaded]`

`&DocuSignVisible=1`

`&DocuSignR1Id=[The record ID of a Salesforce Contact, Lead, or User. This can be done with Salesforce fields, typically a lookup or formula field, with Contact Roles, or with more advanced options.]`

`&DocuSignR1Type=Signer`

`&DocuSignR1Role=Signer+1`

`&DocuSignR1RoutingOrder=1` ( Specifies that recipient 1 will sign first.)

`&DocuSignR2Id=[The record ID of a Salesforce Contact, Lead, or User. This can be done with Salesforce fields, typically a lookup or formula field, with Contact Roles, or with more advanced options.]`

`&DocuSignR2Type=Signer`

`&DocuSignR2Role=Signer+2` (Note, that this will allow the individual specified as R2 to take advantage of all tags set up for Signer 2.)


`&DocuSignR2RoutingOrder=2` ( Specifies that recipient 2 will receive the document after signer 1 has completed.)

`&DS7=7` or 17 (Bypasses the Conga Composer user interface (background mode) and automatically sends. Salesforce buttons can only use DS7=7.)

7. Add this button to the page layout where you want it to run, for example the Opportunity,
8. Click your button.
9. Your solution will run in Background Mode and will be automatically delivered to the first recipient.

## Use Custom DocuSign Tags to Writeback to Salesforce

Use custom DocuSign tags to populate fields in Salesforce from Conga Composer documents.

 This is an advanced feature that can be integrated with the Conga Composer but not supported by the Conga Technical Customer Success Team. You need to connect with the DocuSign technical support team.

One of the key features of DocuSign is the ability to add tags to a document that will define where a recipient will input information like their signature, initials, name, etc. However, DocuSign also offers [the ability to use custom tags](#) to capture other types of information from signers.

 Custom tags can be related to Salesforce, which gives users the ability to push data from a DocuSign Envelope back into specific Salesforce fields.

The first step is to establish DocuSign custom tags, which is done in the preferences of the DocuSign Administration Console:

The screenshot shows the DocuSign Composer interface. The top navigation bar includes 'Home', 'Manage', 'Send', 'Dashboards', and 'Reports'. The left sidebar is divided into 'Member Profile' and 'Member Options'. Under 'Member Options', 'Custom Tags' is highlighted with a red box. A red arrow points from this box to the 'Shipping Method' tag in the main content area. The 'Custom Tags' section contains a table with the following data:

	Type	Label <sup>⤴</sup>	Tool Tip
<input type="checkbox"/>	Date	Ship Date	
<input type="checkbox"/>	List	Shipping Method	Air,Ground;Slow Boat,Carrier Pigeon

Below, you can see how the Shipping Method custom tag has been configured:

**Member Profile**

- Personal Info
- My Account Address
- Sharing
- Connected Apps
- Names Available
- Manage Identity
- Electronic Notary Public
- Manage Email Notifications
- Time Zone
- Change Password
- Change Email

**Member Options**

- Permissions
- Address Book
- Custom Tags
- Template Matching

**Account Administration**

- Address
- Branding
- Billing
- Invoices
- Connect
- Envelope Custom Fields
- Features
- Reminders & Expirations
- Users
- Groups
- Manage Signers
- Permission Profiles
- Envelope Publish
- Locked Out Users

**Custom Tags**

Salesforce Field:  Relate to Salesforce  
 Opportunity > Shipping\_Method\_\_c Edit  
 -- Select --  
  
  
 Writeback  
 Allow Sender to Edit  
 Cancel

Label: Shipping Method \* Type: Drop Down List

Tool Tip: Air;Ground;Slow Boat;Carrier Pigeon  
 (semi-colon delimited list of items)

Initial Value:

Font: Arial  
 Font Size: 12  
 Font Style: Default (Regular)  
 Font Color: Default

Conceal entered value as \*\*\* characters

Width (pixels): 1 Height: 0

Max Characters:

Regex Pattern:

Validation Error:

Anchor: \\Shipping\_Method\_{r}

The Form Field is:  Required  Locked  Fixed Size  Shared

Recipient Note:  Include Note in Email

Save Cancel

1. This tag is related to a Salesforce field – specifically, the Shipping\_Method\_\_c custom picklist field on the Opportunity object. Checking the Writeback checkbox allows data from the custom tag to overwrite the field in Salesforce. Lastly, the Salesforce picklist values are included in the Tool Tip field here in the DocuSign Custom Tag definition.
2. DocuSign offers a number of formatting options for the data that will populate this field. This provides a handy way to match the formatting style of a custom tag with the rest of the content in your document.
3. Arguably the most important part of defining a custom tag is the Anchor. This is a string of text that an administrator defines, and represents what will actually be added to a document template before merging it with Composer and sending it via DocuSign.

Take careful note of the {r} placeholder value in the anchor. This is a variable – when you actually place the anchor in your template, you’ll replace this value with a number to indicate which signer role will complete this field. For example, to add this particular

custom anchor tag to a document template for the first signer role, type `\Shipping_Method_1\`.

Finally, the custom tag needs to be Shared for other DocuSign users (not just the administrator) to use it.

Below is a contract template that includes standard anchor tags to capture the signature of two recipients, but also two custom anchor tags for the first signer to input information:

*By signing and accepting below you are acknowledging that you have read and agree to the specific terms outlined in this document and wish to proceed with the implementation of the aforementioned products and services.*

<code>\s1\</code>	<code>\s2\</code>
<i>Customer Signature</i>	<i>{{ORG_NAME}} Signature</i>

Shipping Method: <code>\Shipping_Method_1\</code>	Ship Date: <code>\Ship_Date_1\</code>
---	---------------------------------------

✔ It's highly recommended to apply white color to DocuSign anchor tags in document templates. Otherwise, the literal text of the original anchor tag (like `\s1\`) will show up in the output file. Coloring the tags white makes it so only the fillable tag shows in the output file, not the original anchor text.

Next, create a Conga Composer solution that runs from a custom button on the Opportunity. On the Opportunity record from where the contract is being created, the Shipping Method and Ship Date fields already have values:

Opportunity Owner	<b>Robert Boyd</b> [Change]
Opportunity Name	<b>Starter Package Restock</b>
Account Name	<b>OneOneZero Technologies</b>
Type	<b>Existing Business</b>
Shipping Method	<b>Ground</b>
Ship Date	<b>4/10/2015</b>

When Composer generates the contract and sends it through DocuSign for signature, the first recipient (in this case, the customer) is presented with this DocuSign Envelope:



Notice how the DocuSign custom tags display the current values of the Shipping Method and Ship Date fields from Salesforce. If the customer wanted ground shipping and April 10th was the correct date, they wouldn't need to make any changes.

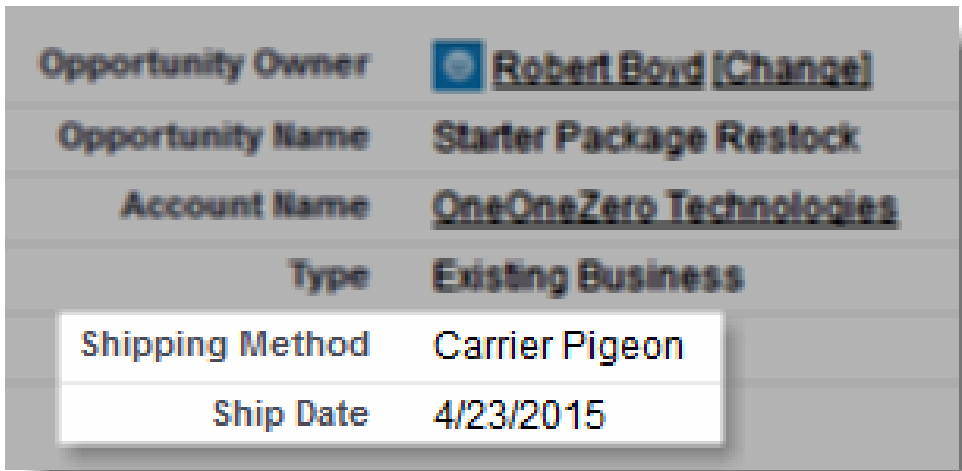
However, if the customer wants to change the delivery method and also wants to push the shipment date a little farther into April, they can modify these fields right here:




After making these changes, the customer could simply DocuSign the contract and it would be automatically routed to the second signer:



After the first recipient modifies the shipping method and ship date and signs the contract, the corresponding fields on our Opportunity in Salesforce are updated:



The image shows a screenshot of a Salesforce Opportunity record. The record is displayed in a list view with a grey background. The fields and their values are as follows:

Opportunity Owner	 Robert Boyd <a href="#">[Change]</a>
Opportunity Name	Starter Package Restock
Account Name	<a href="#">OneOneZero Technologies</a>
Type	Existing Business
Shipping Method	Carrier Pigeon
Ship Date	4/23/2015

With the use of custom tags, you are able to capture up-to-date information from your signers and populate fields in Salesforce. Conga customers have leveraged this DocuSign feature to do things such as help normalize their data, capture key information to accelerate the time it takes to close a deal and even onboard new employees.

## In-Person Signing with Conga and DocuSign

With Conga Composer and DocuSign you can generate any document (Contract, Agreement, Application, etc.) and utilize DocuSign "In-Person Signing" to handle that requirement.

In-Person signing with Conga and DocuSign allows you to generate your document with Composer and pull up the DocuSign signature block on your device for signing in person via DocuSign.

### Steps to Implement

1. Enable DocuSign's Sign in Person Functionality by contacting DocuSign support directly.
2. Create a Conga Solution record or create a new Conga button manually.
3. Turn on the DocuSign Integration - This is done by adding `&DocuSignVisible=1`.
4. Add an in-person signing host, which is required for in-person signing sessions. The signing host must be an internal DocuSign for Salesforce user, and will be specified in the `DocuSignR1Id` parameter. This can reference a static user id or merge the record owner or button's running user's record id.



5. Add the DocuSign sign in person parameters - DocusignR1SignNow and DocusignR1SIPName.
6. Add the Docusign role, type, and routing order parameters.
7. Repeat these steps for additional sign in person recipients.

#### Example URL:

```
https://composer.congamerger.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id{!!Opportunity.Id}
&DocuSignVisible=1
&DocuSignR1Id=00536000002tSDo
&DocuSignR1SignNow=1
&DocuSignR1SIPName=John+Doe
&DocuSignR1Type=Sign+In+Person
&DocuSignR1Role=Signer+1
&DocuSignR1RoutingOrder=1
&DocuSignR2Id=00536000002tSDo
&DocuSignR2SignNow=1
&DocuSignR2SIPName=Jane+Doe
&DocuSignR2Type=Sign+In+Person
&DocuSignR2Role=Signer+2
&DocuSignR2RoutingOrder=2
```

- &DocuSignVisible=1 --> Turns on Docusign Integration
- &DocuSignR1Id=00536000002tSDo ---> This is the in-person signing host and is an internal Salesforce user &DocuSignR1SignNow=1 --> Enables Signing in Person for this signer &DocuSignR1SIPName=John+Doe --> In Person Signer's Name
- &DocuSignR1Type=Sign+In+Person --> Specifies that this signer will be signing in person
- &DocuSignR1Role=Signer+1 --> Standard first signer role value, which can be replaced with a custom signer role &DocuSignR1RoutingOrder=1 --> Indicates this is the first signer in the routing order

**⚠ A Docusign in-person signing session might continue after all sign-in-person recipients have completed signing the document. You might experience this when:**

- A sign-in-person recipient is immediately followed by a standard signer who should receive the agreement through email.

- For example: &DocuSignR1Type=Sign+in+Person and &DocuSignR2Type=Signer
- The sign in person host for the first, in-person signer and subsequent signer are the same user in Salesforce. For example: &DocuSignR1ID=00541000000bK9y and &DocuSignR2ID=00541000000bK9y.
  - To fix the issue, create a Contact record in Salesforce with the same name and information as the internal Salesforce user and reference the Contact record ID for the signer that follows the in-person signer.

## Integrating Composer with DocuSign for Salesforce 7

Conga Composer now integrates with DocuSign for Salesforce 7. For steps to integrate Conga Composer with versions of DocuSign for Salesforce previous to DocuSign for Salesforce 7, see [Setting Up DocuSign for Salesforce](#).

**⚠ Customers using an earlier version of DocuSign for Salesforce (6.9 and earlier) do not need to make changes. This is only applicable to DocuSign for Salesforce 7.**

Please follow the configuration steps below to integrate Conga Composer and DocuSign for Salesforce 7.

- Composer Integration with DocuSign for Salesforce 7 requires Composer version 8.167 or later.

**i** Upgrade Conga Composer to the current version available in the AppExchange.

- If you are using DocuSign for Salesforce 7 and Composer with the Composer parameter &DS7=17 (background mode), an upgrade is not required.
- Enter your [Connect User](#) DocuSign Account credentials into the Integration Partners section on the Conga Composer Setup object.

**i** **Connect User:** Salesforce user that authenticates on behalf of all DocuSign eSignature for Salesforce users to the Salesforce service. Also, updates or creates Salesforce records to the configured Salesforce org.

- DocuSign for Salesforce 7 integration requires you to enter the DocuSign API Account ID, Connect User DocuSign Account Email, and Connect User > DocuSign Password in the aforementioned section.
- The DocuSign API Account ID is found by navigating to the DocuSign Admin tab in Salesforce and clicking on the Account sub-tab.

- For additional details on connecting your DocuSign account to your Salesforce account, please see this [DocuSign support article](#).
- If you customized the default Recipient Role settings in your DocuSign 7 account, a comma-delimited list of the role names is also required in this section.
- A screenshot of the DocuSign integration section in Conga Composer Setup is listed below.

✓ Please verify the DocuSign Account Credentials that are entered are for the Connect User. There are no security controls in place that will prevent entering in credentials for a different DocuSign Account User.

ⓘ When updating the Recipient Roles, it is required to re-enter the Connect User DocuSign account password before saving.

## Integrating Composer with DocuSign App Launcher Package

The standalone (legacy) DocuSign eSignature for Salesforce app has been de-listed from the Salesforce App Exchange. This includes the DocuSign eSignature for Salesforce CPQ app.

The legacy app has been replaced by the new **DocuSign Apps Launcher** package of apps that includes:

- DocuSign eSignature for Salesforce
- DocuSign Gen for Salesforce
- DocuSign Negotiate for Salesforce
- DocuSign CLM for Salesforce

### ⚠ Note

- If you install any one of these apps in your Salesforce organization, the entire package of apps installs.
- If you already have the legacy DocuSign 7 package installed in your Salesforce Org, you can also install the new DocuSign App Launcher package.
- With Composer, you can use both DocuSign 7 and DocuSign App Launcher.


Perform the following steps to send the Composer documents for signature using the DocuSign App Launcher package:

1. Create a **Send With DocuSign** button in the *Conga Solution* layout as follows:

### Important

Because the *Send With DocuSign* button does not generate documents, you must first create a *Composer* button that generates the document and saves it to the *Files* related list, then send the document for signature using the *Send with DocuSign* button. For more information on how to create a *Composer* solution, refer to the [Document Automation Wizard \(DAW\)](#) topic.

- a. From the **App Menu**, search and select **DocuSign Apps Launcher**.
  - b. Go to the **DocuSign Setup** tab.
  - c. Click the **eSignature** option under the *Configuration* in the left panel.
  - d. Click the **Button** tab.
  - e. Select the **desired object** (for example, Account) > select **Layouts** > click the **Update Layouts** button.
2. Navigate to the object you want to use for merging the document and send it for signature.
  3. To merge and generate the document, click the *Composer* button that you have created.
  4. Click the **Send With DocuSign** button. The *DocuSign Prepare & Send* page is displayed.
  5. Ensure that the desired *Composer* merged documents are included. You can also include additional documents.
  6. Ensure that the desired Recipients are included. You can adjust the signing order if necessary.
  7. Click the **Send** button after review/edit the document.

 Visit *DocuSign Documentation* to know more about [DocuSign eSignature Prepare & Send](#) page.

## Transition from SpringCM to the Conga Product Suite

Options for replacing SpringCM-integrated *Composer* solutions with the Conga Product Suite.

Conga is no longer making updates or delivering new enhancements to the SpringCM integration with Conga *Composer*. While SpringCM-integrated Conga solutions may still function, continuing to use the integrated solutions creates potential risks such as solution instability, performance issues, and loss of functionality.


It is highly suggested to replace any SpringCM-integrated Conga solutions with the Conga Product Suite. Conga offers powerful Salesforce-based products that can seamlessly replace SpringCM-integrated Conga solutions and increase functionality in use cases.

See the chart below to examine how the Conga Product Suite matches SpringCM products.

SpringCM Product	Conga Product
Document Generation	Conga Composer, Trigger, Batch, and Conga Collaborate
Contract Management	Conga Contracts and Conga Contracts for Salesforce
Document Management	Conga Contracts and native Salesforce document repository
Workflow	Conga Orchestrate and Conga Trigger
DocuSign	Conga Sign

## Composer Enterprise Edition

Information about features specific to the Enterprise Edition of Composer, including third-party data encryption services, external data, and API.

 The Composer Enterprise Edition package does not include Conga Contracts for Salesforce (CFS) Negotiator Edition. It must be installed separately in order to work with Composer Enterprise Edition.

Select one of the following topics for more information:

- [Advanced Features](#)
- [CipherCloud for Salesforce](#)
- [Symantec Cloud Data Protection \(fka PerspecSys\)](#)
- [External Data](#)
- [Custom Branding](#)
- [Negotiate, Redline, and True-Up in Composer Enterprise](#)
- [Increased Conga Template Limit](#)

## Advanced Features

Composer Advanced Features enables users to send documents directly from Composer and securely save them in Salesforce. It gives you access to the Conga File Repository, the Composer Link Service, Notification Emails, and the Conga PDF Viewer.

[Composer Link Service](#) allows users to use Composer to easily generate and send documents in an email directly from Salesforce. Each document generated with Composer Link Service is given a unique, shareable link and is associated with a Document History record in Salesforce.

Documents generated with Composer Link Service are securely stored in the proprietary [Conga File Repository](#).

Users and Recipients can then easily view documents generated by Composer Link Service in the [Conga PDF Viewer](#). The Conga PDF Viewer provides users and recipients access to documents without having to log into Salesforce.

Composer Link Service also provides users with insightful [Notifications Emails](#) to inform users when recipients interact with documents sent with Composer Link Service.

- [Composer Link Service](#)
  - Composer Link Service allows users to generate and send documents with Conga Composer.
- [Notification Emails](#)
  - Notification emails are emails sent to Composer users when recipients interact with documents sent from Conga Link Service.
- [Conga File Repository](#)
  - The Conga File Repository is a proprietary repository that stores documents generated with the Conga Link Service.
- [Enabling Composer Link Service and Conga File Repository](#)
- [Document History and Document History Detail Objects](#)
  - The Document History object is a custom object that contains the link to a document generated with Composer Link Service.
- [Conga PDF Viewer](#)
  - View documents generated with Composer Link Service in the Conga PDF Viewer

## CipherCloud for Salesforce

CipherCloud for Salesforce addresses data privacy, residency, security, and compliance concerns allowing organizations to confidently deploy their Salesforce solutions.

- [CipherCloud for Salesforce](#)
  - CipherCloud for Salesforce addresses data privacy, residency, security, and compliance concerns so that organizations can confidently deploy Salesforce, [Force.com](#) applications, and Chatter.
- [Integrate Composer with CipherCloud](#)
  - After configuring CipherCloud, the integration with CipherCloud utilizes Conga Composer's Custom Setting, Server Endpoint URL.

## CipherCloud for Salesforce Overview

CipherCloud for Salesforce addresses data privacy, residency, security, and compliance concerns so that organizations can confidently deploy Salesforce, [Force.com](#) applications, and Chatter.

The CipherCloud solution understands Salesforce metadata and provides organizations with the ability to easily tag any standard and custom fields, files, and attachments as sensitive. Once configured, the CipherCloud solution secures this data and prevents it from being accessed by unauthorized users, while working seamlessly with Salesforce.

This revolutionary technology, based on function-preserving encryption and advanced tokenization, provides full protection and control, and allows customers to retain complete control over keys. The solution is transparent to the end user, and all native Salesforce functionality, such as searching and sorting, is preserved.

CipherCloud is available separately on the AppExchange. You must subscribe to that service to make use of this integration.

## Integrate Composer with CipherCloud

After configuring CipherCloud, the integration with CipherCloud utilizes Conga Composer's Custom Setting, Server Endpoint URL.

As the administrator of your Salesforce instance, you would enter the valid host + domain name as directed by CipherCloud by following the steps below:

1. Navigate to **Setup** → **Develop** → **Custom Settings**.
2. Click **Manage** for Conga Composer Settings.
3. If no previous settings exist, click **New**; otherwise, **Edit**.
4. For the Name field, enter Conga Settings.
5. For the Server Endpoint URL, enter the CipherCloud proxy server address
6. Click **Save**.

No further changes are necessary in individual Composer buttons or links.

- ✔ Use only Conga Queries, not Reports. We recommend in building Composer solutions that you use Conga Queries rather than Salesforce Reports as data sources. This is because Queries run through the Salesforce API, and therefore, will be channeled through CipherCloud.

## Symantec Cloud Data Protection (fka PerspecSys)

The PerspecSys Cloud Data Protection Gateway for salesforce.com allows organizations to maintain control of their sensitive business data while using salesforce.com and [Force.com](#) applications.

- [Symantec Cloud Data Protection Gateway for Salesforce](#)
- [Integrate Composer with Symantec Cloud Data Protection \(fka PerspecSys\)](#)

## Symantec Cloud Data Protection Gateway for Salesforce

The PerspecSys Cloud Data Protection Gateway for [salesforce.com](#) allows organizations to maintain control of their sensitive business data while using [salesforce.com](#) and [Force.com](#).

Any sensitive information that leaves a business's environment and enters the [Salesforce.com](#) cloud can be tokenized or encrypted, ensuring that data remains undecipherable when it is being processed or stored outside of the enterprise's environment.

This PerspecSys solution is available separately on the Salesforce AppExchange. You must subscribe to that service to take advantage of this integration.

## Integrate Composer with Symantec Cloud Data Protection (fka PerspecSys)

After installing and configuring PerspecSys in your Salesforce organization, utilize Conga Composer's Custom Setting, Server Endpoint URL. As the administrator of your Salesforce instance, you would enter the valid host + domain name as directed by PerspecSys. Follow the steps below.

1. Navigate to **Salesforce Setup** → **Develop** → **Custom Settings**.
2. Click **Manage** for Conga Composer Settings.
3. If no previous settings exist, click **New**; otherwise, **Edit**.
4. In the Name field, enter **Conga Settings**.



5. For the Server Endpoint URL, enter the PerspecSys proxy server address, such as:  
`https://myprsserver.mycompany.com`.

The actual endpoint URL will be provided by PerspecSys during the configuration of your system.

No further changes are necessary in individual Composer buttons or links.

- ✔ Use only Conga Queries, not reports. We recommend in building Composer solutions that you use Conga Queries rather than Salesforce Reports as data sources. This is because Queries run through the Salesforce API, and therefore, will be channeled through PerspecSys.

## External Data

Retrieve data from external systems, like an ERP, and combine it with Salesforce data in a single Composer document.

Select one of the following topics for more information:

- [Overview of External Data](#)
- [Setup and Configure External Data Sources](#)
- [Build Your Conga Query to Retrieve External Data](#)
- [Build Your External Data Composer Solution](#)

## Overview of External Data

By leveraging the [OData protocol](#), users can now import data from sources outside of Salesforce into their Conga Composer solutions.

- ⚠ Data from multiple instances of Salesforce can also be combined.

With minimal setup and configuration, existing Composer tools can be used to integrate external data with Salesforce data for document creation.

### **i** Important

Prior to starting this process, you must have external OData data sources configured and available for use. If you do not have your own data sources, the list of OData connectors and services below will help you get started.

## Components of a Composer External Data Solution

Using External Data with Conga Composer requires three components.


### 1. Conga Composer Solution

- Setup an OData endpoint in [Conga Composer Setup](#).
- The Composer Solution must include a [Conga Query that retrieves external data](#) (identified with the [QueryID](#) parameter) and the [xid](#) to identify the external data source.

 The xid parameter supports both [Conga Keys](#) and standard record IDs.

### 2. OData Service

- Customers must build and configure the OData service to act as a data source for Conga Composer.

 Conga Composer uses OData version 2.0. You may use a newer version to build your OData service, but some features in newer OData versions may not be available for use.

- Prior to starting this process, you must have external OData data sources configured and available for use. See the list of resources below for more information about building and configuring the OData service.
  - <https://www.odata.org/ecosystem/>
  - <https://www.odata.org/documentation/>
  - <https://www.progress.com/odata>
  - <https://www.cdata.com/odata/>
- The OData service must support the necessary URL components query syntax. For more information, see [Sections 2 and 5 in OData's documentation](#).

### 3. Data Source

- The data source is a third-party data source or repository hosting the data queried by Conga Composer.

## Setup and Configure External Data Sources

This article explains setting up and configuring Composer for use with your external data sources.


1. Log in to Salesforce and navigate to the Conga Composer Setup tab.
2. Click the External Data Sources under the *Enterprise* section.
3. Click **Create New Connection**.

4. Select Basic or No Auth from the Authorization Type field.

 The Oauth Authorization type is not yet supported.

5. Enter the following information for your OData connection:

- Alias: The name you'll use to reference this OData connection in your Composer template. Alphanumeric characters, no spaces. (Referenced later as ExternalConnection)
- Authorization Type: "Basic" authorization requires the Username and Password to your OData connection. The username and password will resolve into a Basic Authorization base64 header in the HTTP request to the endpoint. The header parameter will look like: Authorization=Basic {base64EncodedKey} "No Auth" requires only the endpoint and will not use any authorization method.

 Regardless of Authorization Type, all requests from Conga Composer to your OData endpoint will be secured over HTTPS.

- (Optional) - Username and Password: Enter your Username and Password if "Basic" authorization is selected for the Authorization Type field.
- Endpoint: The url to your OData source.
  - Example Endpoint: `https://<your-domain>.com/yourODataEndpoint/`

6. Click **Save**.

7. (Optional) Repeat steps 3 – 6 for each external data source.

 Additional data sources can be added at a later time.

## Build Your Conga Query to Retrieve External Data

This article walks through creating a Conga Query using OData syntax to query your external data sources. Each OData Conga Query you create will be appended to an individual external data source [previously created](#).

### Important

The OData service must support the necessary URL components query syntax. For more information, see [Sections 2 and 5 in OData's documentation](#).

Behind the scenes, Conga Composer will combine each OData Endpoint with a Composer OData query to gather your data. The resulting URL used to call your OData data source will take the form:

`https://<your-domain>.com/yourODataEndpoint/Products?$orderby=Name&$format=json`

**i Important**

The query syntax for the OData query must contain `$format=json` to retrieve and merge data with Conga Composer.

## Example External Data Queries

External Data query syntax is different than the SOQL Query syntax used to gather data in Salesforce. See the examples below for example External Data query syntax.

**⚠** Use pv functionality to dynamically filter the results of your query. When using pv functionality in an OData query, you must define the pv in both the query syntax and in the xid parameter value. For more information on using pv functionality with queries, see [Dynamically Filter Reports and Queries with pv Values, xid](#), and the examples with pv functionality below.

- Query that returns Products ordered by Price
  - `Products?$format=json&$orderby=Price`
- Query that returns only the first 10 Product records ordered by Price
  - `Products?$format=json&$orderby=Price&$top=10`
- Query that returns the Name and Price of all Products
  - `Products?$format=json&$select=Name,Price`

**⚠** The SOQL equivalent of this query is `SELECT Name, Price FROM Products`

- Query that returns Products that have more Units in Stock than a specific field value on an Account record
  - `Products?$format=json&$filter=UnitsInStock gt {pv0}`
  - syntax for the xid parameter in the Conga Solution:
    - `&xid=ExternalProduct:CongaData:a045000001DjTva?pv0={!Account.AverageStock}`

**✓** For more information on the external data query syntax, see [Section 5 \(Query Options\) of OData's documentation](#). For information on additional OData query filters and functionality, see the Querying Data section of [OData's Basic Tutorial](#).

**⚠** Conga's External Data feature is currently set up to retrieve a list of object records (Entity Collections) only, as opposed to just a single object record (Individual Entity by ID, Individual Property, and Individual Property Raw Value). Do not use query


syntax, such as (1) or (3) to attempt to return 1 or 3 specific records. Instead, use additional filters in the query syntax to filter the returned list of data. For more information, see [OData's Basic Tutorial](#).

## Creating an External Data Query

1. Navigate to the Conga Queries tab.
2. Click **New**.
3. Enter the information for your Conga Query using the following:
  - a. Name: A name for this Conga Query.
  - b. Description: (Optional) A description for this Conga Query.
  - c. SOQL Select Statement: This is where your OData query will reside. Use OData query syntax here instead of SOQL. The OData query you enter here will be related to an endpoint when you build your Composer solution in Step 3.
    - OData Query Example: `Products?$orderby=Name&$format=json`
4. Click **Save**.

Repeat steps 1 - 4 for each OData query you'd like to integrate.

## Build Your External Data Composer Solution

 This documentation assumes you have familiarity with building a Conga Composer solution. For additional information, see [Getting Started with Conga Composer](#). This step also introduces a new parameter, **xid**.

Follow the steps below to create an External Data Composer solution.

1. Add the xid parameter to your Composer solution in the following format:
 

```
&xid=QueryAlias:ExternalConnection:QueryId
```

  - xid - Indicates an external data source
  - QueryAlias - Choose an alias for your query and enter it here. Required. Cannot contain spaces or special characters, and must be between 3 and 20 characters long. This QueryAlias becomes the dataset name in Template Builder and in your template.
  - ExternalConnection - ExternalConnection is the alias of the data connection you set up in the [Set up and Configure External Data Sources](#) step.
  - QueryID - Salesforce Id of the Conga Query you set up in the [Build Your Conga Query](#) step.

⚠ Use pv functionality to dynamically filter the results of your query. For more information on using pv functionality with queries, see [Dynamically Filter Reports and Queries with pv Values](#).

2. Add the dataset to your template by referencing the QueryAlias from above.

✔ Information on tables of repeating data in Word can be found [here](#).

### Example xid Parameter

```
&xid=MyOracleDataAlias:MyOracleExternalConnection:a045000001DjTva
```

### Example External Data Table with Merge Fields in a Word Template

	Code	Product description	Product name	
{{TableStart:MyOracleDataAlias}}	{{Code}}	{{ProductDescription}}	{{ProductName}}	{{TableEnd:MyOracleDataAlias}}

## Custom Branding

You can customize the Composer® interface with your own logo.

Select one of the following topics for more information:

- [Custom Branding for Conga Composer](#)
- [Your Logo on the Composer User Interface](#)

## Custom Branding for Conga Composer

How to customize the Composer® interface with your own logo.

You can customize the Composer® interface with your own logo.

To Add a Custom Logo to the Composer UI

- Image must be stored at a publicly accessible URL.
- Image must be 300 x 32 pixels (images of different sizes are scaled).
- Use the BrandingCode parameter in your button or link URL.
- Add to all branded solutions.

The image and button is tied to your Salesforce® Org.

## Your Logo on the Composer User Interface

The Composer interface branded with your logo.

After following the steps in [Custom Branding for Conga Composer](#), the Composer interface will be branded with your logo for all of the users in your community.


## Negotiate, Redline, and True-Up in Composer Enterprise

Conga Contracts for Salesforce Negotiator edition provides negotiate, Redline, and True-Up capabilities in the Composer Enterprise package.

- [About Conga CLM \(Contracts for Salesforce\)](#)

## Increased Conga Template Limit

Composer Enterprise Edition features an increased template limit that allows users to merge 40 Conga Templates at once in a Composer solution. The maximum size for Composer-generated output files is 40 mb.

 Composer Business Edition allows users to merge a maximum of 20 Conga templates at once.

## Composer Troubleshooting Help

Troubleshooting help for Composer.

- [Caching Error on the Conga Composer Setup Menu](#)
- [Chrome Redirect Issues when using Composer](#)
- [Composer Not Redirecting to Adobe Esign in Community](#)
- [Composer Troubleshooting Tips](#)
- [Conditionally Show or Hide Data in Word Templates](#)
- [Conga Composer Debug URL](#)
- [Conga Word templates - When there is no data in a field, spacing and font size are altered](#)
- [Connection to Salesforce Failed](#)

- [CreateEnvelope in DocuSignAPI UserID does not have a valid membership in Account](#)
- [Create a Conga Composer List View Button \(Lightning Experience\)](#)
- [Critical Update Actions for Composer Versions](#)
- [Data appears on the QueryDataX dataset for some users, but not others. Why?](#)
- [Data appears on the ReportDataX dataset for some users, but not others. Why?](#)
- [Differences between Conga Composer versions](#)
- [Disable Composer 7 Triggers](#)
- [Document not Merging to SpringCM](#)
- [Removing Duplicate Conga Composer Setup Tab](#)
- [Duplicate solutions record created where solution pack installed](#)
- [Edit My Own Posts Permission Required to Post to Chatter](#)
- [Error Using Google Drive Integration with Composer Templates](#)
- [Excel to PDF - Different Values](#)
- [Fields not Copying to the Clipboard in Conga Composer](#)
- [Fields Won't Display Even After Permission Is Granted in Salesforce to Access that Field \(Release 8\)](#)
- [Field Not Merging in PowerPoint Template](#)
- [How do I know what version of Composer I'm using?](#)
- [How do I move my Conga Composer subscription from one Salesforce instance to another?](#)
- [How to use Extended Ping and Trace Route Instructions for Conga](#)
- [Images in Word](#)
- [Insufficient Privileges Error when a System Admin launches an OAuth button as a user](#)
- [Log Into One Salesforce Instance or as One User per Browser](#)
- [Merge Fields Displaying Unexpected Results](#)
- [One or more documents failed to convert to PDF](#)
- [Opplineltems Dataset Missing in Template Builder](#)
- [Remote Site Error in Conga Solution Manager](#)
- [Removing duplicate tabs](#)
- [Shipping and Billing Merge Fields](#)
- [Templates not displaying or showing fewer than expected in Composer \(Release 8\)](#)
- [Template is not populating in template list on the Conga dialog box or is not visible in Conga Template Manager](#)
- [Troubleshooting Font Styles](#)
- [Using Send through External Email Services with Conga Composer](#)
- [What is Aspose related to my PDF?](#)
- [Where Did My Conga Mail Merge Tab Go? \(Release 8\)](#)
- [Why didn't my dynamic formula update correctly in my Conga Composer Excel output file?](#)



- [Templates appear for administrators but not end-users](#)

## Caching Error on the Conga Composer Setup Menu

An intermittent caching occurs on the new Conga Composer Setup Menu. This article describes how to resolve the error message. The following error message occurs when users with a large amount of Salesforce browser cookies attempt to access Conga Composer Setup.

### 494 ERROR - The request could not be satisfied.

**494 ERROR**

**The request could not be satisfied.**

---

Bad request. We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner. If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.

---

Generated by cloudfront (CloudFront)  
Request ID: H64eUBagLkbaglpL5F\_46dzG1B0XxUopt8rPBGes149s55h5r6thJA==

Users likely incur this error when logging into multiple Salesforce orgs from the same browser.

## Resolution

To resolve this error, clear the cache in your browser.

- [Clear cache in Google Chrome](#)
- [Clear cache in Firefox](#)
- [Clear cache in Safari](#)
- [Clear cache in Internet Explorer](#)
- [Clear cache in Microsoft Edge](#)

## Chrome Redirect Issues when using Composer

Current versions of Chrome do not allow redirects to occur as a security feature.

## Question or Issue

When a Composer user in Lightning is running a solution with a DS7 parameter, the merge completes, but the screen never redirects back to Salesforce.

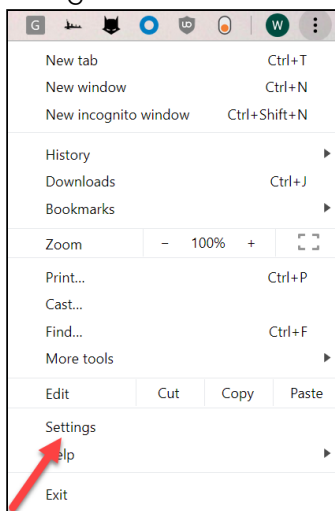
## Causes or Possible Causes

Current versions of Chrome do not allow redirects to occur as a security feature.

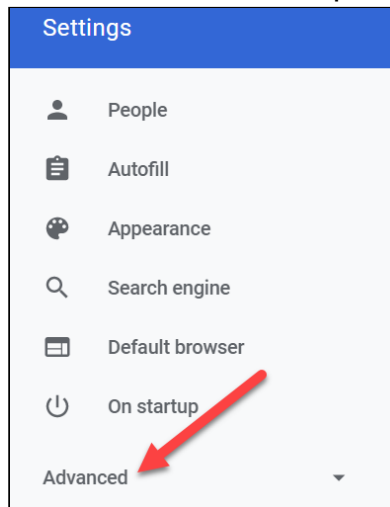
## Related Known Errors and Workarounds

To resolve this issue, enable Chrome to allow redirects when accessing your Salesforce URL by taking the following steps:

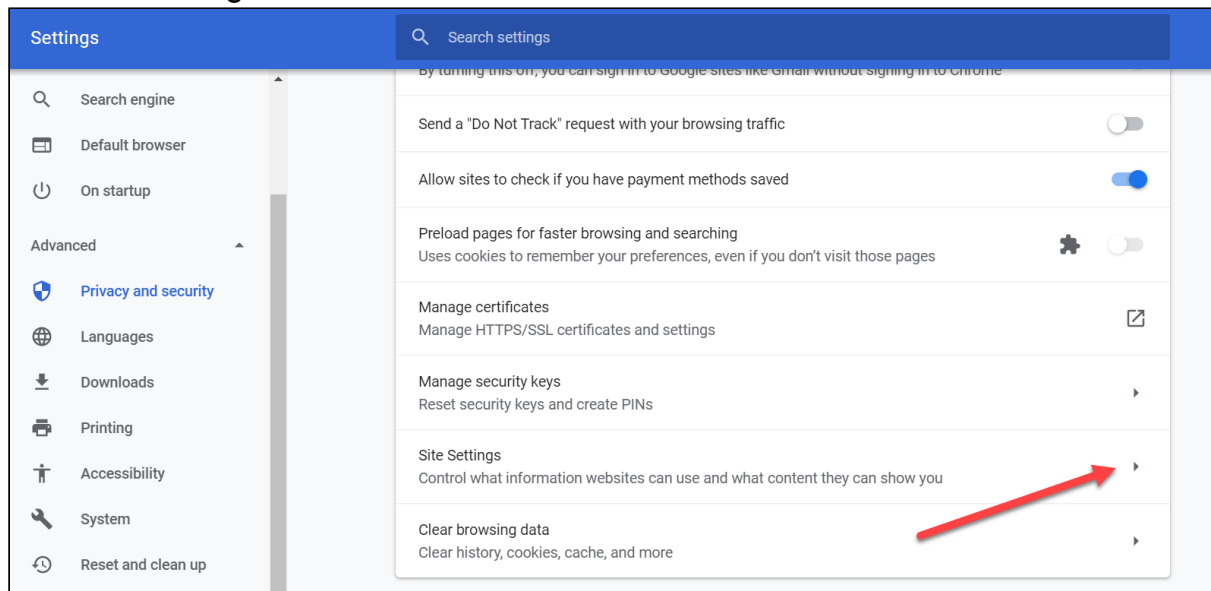
1. Navigate to **Chrome Settings**.



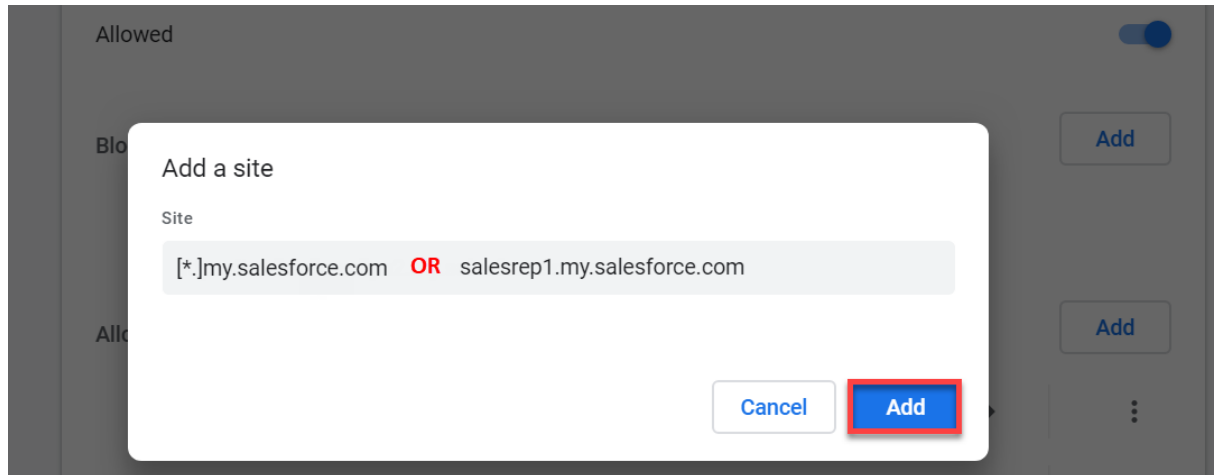
2. Click the **Advanced** dropdown.



3. Click **Privacy and Security**.
4. Click **Site Settings**.



5. In Site Settings, scroll down and click **Pop-Up and Redirects**.
6. Under the Allow section, click **Add** and enter your Salesforce URL (salesrep1.my.salesforce.com) or wildcard Salesforce URL ([\*.]my.salesforce.com).



7. Click **Add**.

## Composer Not Redirecting to Adobe Esign in Community

If you click Merge & Esign on a Composer button in a community, Composer may direct you to a page where you get an error saying the page "is down for maintenance."

To correct this, you'll need to add `&ReturnPath={!$Site.Prefix}` to your Composer button URL to ensure Composer redirects to an Esign URL that corresponds to your community.

## Composer Troubleshooting Tips

The following tips can help you troubleshoot Conga Composer to get you back up and running quickly. Try these tips before doing anything else.

### Clear Your Cache or Try a Different Browser.

The first step in troubleshooting should always be to clear your cache and/or test in a different browser.

To clear your cache in Chrome:

1. Open Chrome.
2. On your browser toolbar, click **More**.
3. Click **History** → **Clear browsing data**.
4. Select the checkboxes for Cookies and site data and Cached images and files.
5. Use the menu at the top to select the amount of data that you want to delete.

## 6. Click **Clear browsing data**.

In Mozilla Firefox:

1. From the History menu, select **Clear Recent History**.
2. From the Time range to clear drop-down menu, select the desired range; to clear your entire cache, select **Everything**.
3. Next to Details, click the down arrow to choose which elements of the history to clear; to clear your entire cache, select all items.

In Internet Explorer:

1. Open Internet Explorer.
2. IE 8: From the Tools menu select **Internet Options**.
3. On the General tab, under Browsing history, click **Delete**.
4. Un-check the Preserve Favorites website data box.
5. Check the Temporary Internet files, Cookies, and History boxes.

In Safari:

1. Click Safari on the upper left-hand side of your screen. In the menu that appears, click **Preferences**.
2. In the window that appears, click the **Privacy tab**. Click the button **Remove All Website Data**.
3. Click **Remove Now** in the pop-up window that appears.

## Test with Different Data or Run on a Different Record.

Sometimes data itself can cause the problem. Test your Composer solution with different data or from a different record to ensure the data you are trying to merge is not the source of the problem.

## Determine Where the Problem Is

If you have reached this point in troubleshooting and have not solved your issue it is time to run a test to determine where the problem is. The problem likely lies in one or more of the following areas: the Composer URL, the Composer Template, or Salesforce permissions.

How do you determine where the problem is? Follow the steps below to run a merge process with a blank template:

1. Save a blank document to your desktop as Test Template.
2. Launch the broken solution.
3. Select the Test Template from the Local Templates tab and click Merge & Download (or Merge & Email).

4. If the Composer window returns an error, the problem likely lies with the Composer URL.
5. If the document merge is successful, the problem likely lies with the Composer template.

If the problem lies with the URL, see the information below about analyzing the Composer URL. If the problem lies with the template, see the information below about deconstructing the template.

## Analyze the Composer URL.

The Composer URL tells you a lot about the solution. [Access the Debug URL](#) and copy it into Notepad or another program where you can view it. Look for illegal characters (spaces, \*, & anywhere other than at the beginning of a parameter, %, pipe (|), ,, etc.), fields that are not updating correctly, or formulas that don't seem to be working. If you cannot solve the problem yourself, you will likely need to provide the Debug URL to your Business Analyst anyway, so you are moving in the right direction.

## Deconstruct the Template.

With larger templates, we recommend removing one page/slide of the template at a time and running the merge process again. For smaller templates, you can remove a paragraph at a time. Keep doing this until you run a successful merge. At that point, you have likely identified the source of the error in the template and you can work on correcting it.

## View Data Workbook

If the problem with your template is that data are not merging, the View Data Workbook is a handy tool to help identify the issue. From the Tools & Settings drop-down on the Composer user interface, you can select the View Data link, which downloads an Excel workbook that contains all of the data Composer has retrieved for your solution.

A simple rule of thumb: if the data you expect to see in your template is not in the View Data Workbook, they are not going to merge into your template.

If data are not in the View Data Workbook: the problem likely lies with the data source retrieving the data (i.e. the Salesforce report or Conga Query). Check the reports or queries for your solution to ensure they are returning data properly.

If data are in the View Data Workbook: the problem likely lies with the template. Check the detail region to ensure it is properly formed with TableStart and TableEnd merge fields (and make sure the proper dataset name, i.e. the alias, is referenced).

## Check Salesforce Permissions for All End Users

In addition to the above, the problem could pertain to Salesforce permissions. Please see [Permissions Needed to Use Composer](#) for more information.

If you are a System Administrator, you can use the Login as User functionality in Salesforce to run solutions, download the View Data Workbook, and review error messages as end-users, which will be helpful in identifying the permissions issue.

✔ **Field Level Security (FLS) is the most likely culprit when it comes to Salesforce permissions.**

## Investigate Error Messages

Lastly, if you are receiving an error message, search for your error message on the support site. You can review all documented error messages in [Composer Error Messages](#). Error message articles contain information about what is causing the issue as well as steps to address it.

If all else fails, view options for [Contacting Support](#). Experienced Business Analysts are on hand in all regions to help you with your issue.

⚠ **Do not use a colon in document file names.**

## Additional Troubleshooting Tips

- Passing an ampersand (&) as data in a Conga Batch or Conga Trigger solution is a known problem.
- Ensure you are not logged into multiple Salesforce instances.
  - Being logged into multiple instances of Salesforce (Sandbox, Production, Development, and so on.) in the same browser is always a bad idea. Try logging out of all instances except the one in which you are working and run the solution again.
- Be sure you have not exceeded product data limits.

- If you are experiencing an issue, it is possible you have exceeded the limits of what the product is able to do. Check the Product Data Limits to ensure you are within the bounds of what the application is able to do.

## Conditionally Show or Hide Data in Word Templates

There are multiple ways to conditionally show or hide data in Composer Word templates. This includes:

- [Table Hide](#)
- [Conditional Table Hide](#)
- [Row Burner](#)
- [IF Statements in a Microsoft Word Template \(Release 8\)](#)

We recommend reviewing articles in Advanced Word Templates to learn how to use these different techniques.

## Conga Composer Debug URL

### About the Debug URL

When you encounter an issue requiring support, you will be asked to provide a URL to access your solution and instance. The Debug URL will produce a URL that you can send to support that will allow them to access your solution without having to request access. You can revoke access anytime under My Connections.

There are several ways to access the Debug URL link:

1. In the Composer UI under Tools & Settings, you will see the Debug URL option.
2. A Debug URL link also appears in error messages.
3. You can also copy the URL from the address bar of the browser after launching a Composer solution (recommended for Composer Release 7)

When you click Debug URL, Conga Composer presents a prompt asking for access.

If you allow:

- The system will get a refresh token from Salesforce to allow access. (The Debug URL uses Salesforce OAuth security.)



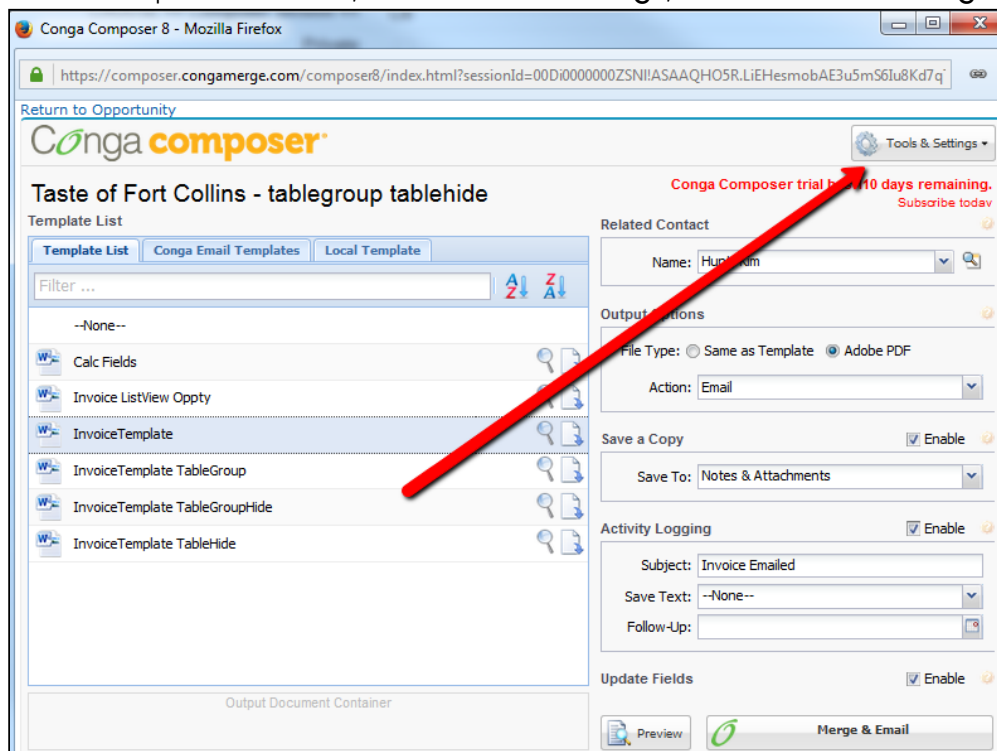
- There are no time limits to how long support can access the solution for troubleshooting.
- Salesforce session timeout settings will not affect access.
- You do not have to remain logged into Salesforce for the URL session to remain accessible.
- You can revoke the access whenever you want via Connections under: **My Settings** → **Personal** → **Connections**

If you deny:

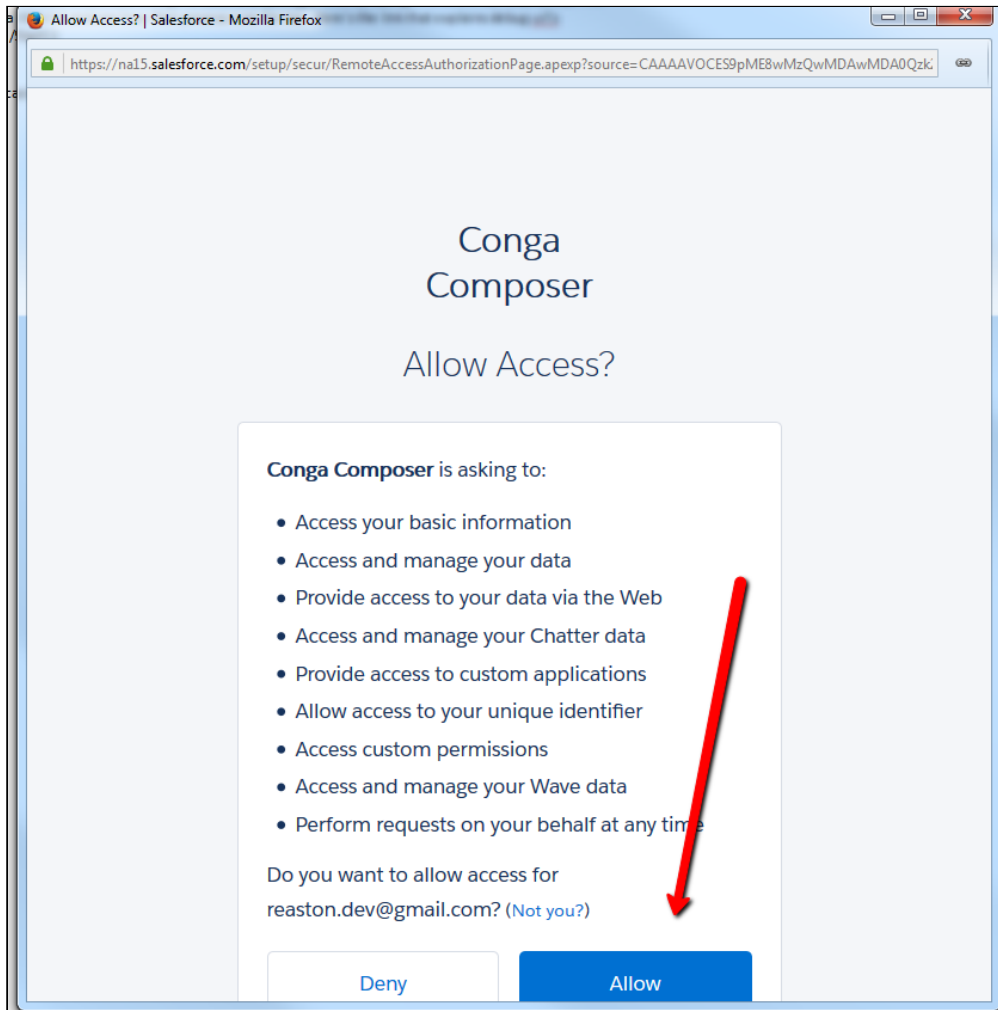
- The URL is generated using the session ID.
- The session will be revoked based on your Salesforce session timeout settings.
- Once you log out of that session the URL will cease to work.
- The URL is not as secure as using the Refresh token.

## Generate a Debug URL (Composer Version 8)

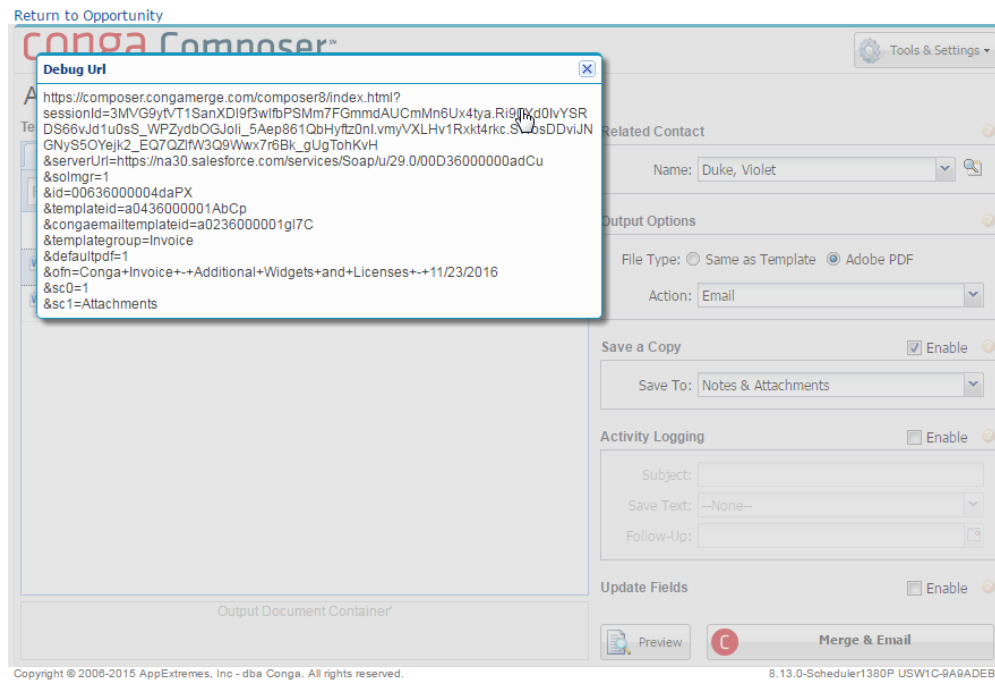
1. In the Composer Window, click **Tools & Settings**, and then click **Debug URL**.



2. Composer presents a prompt asking for access. Click **Allow** to generate a refresh token from Salesforce to allow access. This allows for troubleshooting without time limits. You can revoke this token after your support session. Or click **Deny** to generate the URL using a session ID. This allows for troubleshooting until the current session expires or you log out.



3. Copy the contents of the Debug Url window. This can then be pasted into the support form or emailed to your support representative.



## Generate a Debug URL (Composer Version 7)

1. Click the **Launch** button.
2. The Conga Composer user interface will now display in a new browser tab.

**i** If you have enabled background mode (&DS7), you will need to disable it (&DS7=0) in order to reach the Composer user interface

3. Click on the browser URL, highlight the entire URL, and copy it. The URL you copied from the Composer user interface is the debug URL; it should look something like this:

```

https://www.congamerge.com/apps/Conga/PM.aspx?
sessionId=00Di0000000ktxa!
AR0AQHwx5dfm97QZGlgBVI0cuLpk_jwizj3SmkKU45YhkuWq3yXFYWaKOR8hclBF9_3Mtp__suU
kZZt5SDouNqJy78Pv7Um0
&serverUrl=https://na15.salesforce.com/services/Soap/u/
8.0/00Di0000000ktxa&id=a0Ai000000Lj4uM
&ReportId=[Leads]000i0000005fw1F,[ComplAct]000i00000063IQS,
[ClsdWonOpptys]000i0000005fw1o&TemplateId=

```

## Conga Word templates - When there is no data in a field, spacing and font size are altered

If a field in Salesforce is blank and Conga returns a blank on the Word output file, the output reverts to the default font on the Word template rather than whatever font the merge field was set to.

### Error Message

When there is no data in a field, spacing and font size are altered.

### Cause

If a field in Salesforce is blank and Conga returns a blank on the Word output file, the output reverts to the default font on the Word template rather than whatever font the merge field was set to.

For example, merge fields are set to Calibri font, 6 point size but some of the corresponding fields in Salesforce were blank. Accordingly, Conga outputs a blank on the output file but the font size for the blank is changed to Calibri size 11 because that's the default font for the template.

### What to Do

To remedy this, simply change the default font setting to whatever you need (e.g. Calibri size 6) so that blanks revert to the correct font.

Access the default font from the **Home ribbon** → **Font area** → click **Font flyout** (pop out arrow) → choose your desired font and size → Click the **Set as Default button** and select **For this document only**.

# Connection to Salesforce Failed

## Error Message

This error occurs immediately upon starting Conga Composer and it's caused by the "Lock IP Address" setting being enabled... it needs to be disabled for hosted AppExchange applications.

## Issue

You see an error immediately upon starting Conga Composer.

## Cause

The Lock IP Address setting is enabled. The setting must be disabled for hosted Salesforce AppExchange applications.

## Workaround

To fix the issue:

1. Go to **Salesforce Setup**.
2. **Administration Setup**.
3. **Security Controls**.
4. **Session Settings**.
5. Ensure that the Lock sessions to the IP address from which they originated field is NOT checked.
6. Click **Save**.
7. IMPORTANT! You must restart all instances of the browser for the setting to take effect.

# CreateEnvelope in DocuSignAPI UserID does not have a valid membership in Account

## Error Message

This message is received when using Conga products and DocuSign in a fully automated fashion and the DocuSign Username and Salesforce User Email are different.

## Cause


This message is received when using Conga products and DocuSign in a fully automated fashion (such as using the background parameter/value pair &DS7=17, Conga Trigger, or Conga Batch and the DocuSign Username and Salesforce User Email are different.

## What to Do

The DocuSign Username and Salesforce User Email must be identical.

# Create a Conga Composer List View Button (Lightning Experience)

How to create a Conga Composer List View Button (Lightning Experience).

 Conga does not support customized code. Due to the complicated nature of supporting programmers and programming languages, the integration of Conga with Visualforce, Apex, and Javascript (or other programming languages) is not supported. The example code used below is specific to the Opportunity and will only work for Opportunity list view.

## Integration Method

You can integrate Composer within Salesforce Lightning using the following steps:

1. Create an Apex Class to reference as a set controller.
2. Create a Visualforce page with a set controller referencing the Apex Class.

3. Create a new button and select **Visualforce** for type.
4. Enable Display checkboxes (for Multi-Record Selection) and select Visualforce Page as the Content Source.

The screenshot shows the 'Custom Button or Link Edit' interface. The 'Label' is 'CM Opp List View' and the 'Name' is 'CM\_Opp\_List\_View'. The 'Display Type' is set to 'List Button' with the 'Display Checkboxes (for Multi-Record Selection)' checkbox checked. The 'Content Source' is 'Visualforce Page' and the 'Content' is 'OppCMListView [OppCMListView]'. A green message 'Save was successful.' is displayed at the top right.

5. Add the newly created button to the list in Search Layouts.

**Opportunity Search Layouts**

Search layouts allow you to select the standard and custom fields that are displayed in the following search features:

- Search Results: Columns for search and tagging. [View example](#)
- Lookup Dialogs that pop up when you click the magnifying glass on Lookup fields. [View example](#)
- Lookup Phone Dialogs that pop up when you click the magnifying glass on a SoftPhone dial pad. [View example](#)
- Recently viewed, modified, and created records lists displayed on tab home page. [View example](#)
- Search Results Filter Fields for search. [View example](#)

Search layouts also allow you to select the standard and custom buttons that are displayed in the following search features:

- Search Results for search and tagging.
- List Views for filtering records. [View example](#)

Action	Layout	Columns Displayed	Buttons Displayed
Edit	Search Results	Opportunity Name, Account Name, Account Site, Stage, Close Date, Opportunity Owner Alias	ActionGrid
Edit	Lookup Dialogs	Opportunity Name, Account Name, Account Site	N/A
Edit	Lookup Phone Dialogs	Opportunity Name, Account Name, Account Site	N/A
Edit	Opportunities Tab	Opportunity Name, Account Name, Close Date	N/A
Edit	Opportunities List View	N/A	New, ActionGrid, CM Opp List View, Batch Invoice
Edit	Search Filter Fields		N/A

From Step 1, your Apex Class should look like this:

```
public class OpportunityCMController {
    private final static String urlFieldName = 'Conductor__c';
    private String partnerServerUrl;
    private ApexPages.StandardSetController controller;
    private List<String> opportunityIds = new List<Id>();

    public String CMUrl {get; private set;}

    public OpportunityCMController(ApexPages.StandardSetController controller) {
        this.controller = controller;
    }

    for (Opportunity opportunity : (List<Opportunity>)controller.getSelected()){
        opportunityIds.add(opportunity.Id);
    }

    partnerServerUrl = 'https://' + ApexPages.currentPage().getHeaders().get('Host') + '/services/Soap/u/
    26.0/' + UserInfo.getOrganizationId();
}
```

```

public PageReference prepareCmUrl() {
    CmUrl = 'https://composer.congamerger.com' +
    '?SessionId=' + UserInfo.getSessionID() +
    '&ServerUrl=' + partnerServerUrl +
    '&Id=' + UserInfo.getUserId() +
    '&QueryId=[Opps]a0346000009N2DH?pv0=' + \" + String.join(new List<String>(opportunityIds), '\\\\')+\";

    return null;
}
}

```

And now the Visualforce Page should look like this:

```

<apex:page standardController="Opportunity" extensions="OpportunityCMController"
recordSetVar="opportunities" action="{!prepareCmUrl}">
<apex:outputPanel id="script">
<script type="text/javascript">
window.open("{!CmUrl}","_blank","width=640,height=480");
window.history.back();
</script>
</apex:outputPanel>
</apex:page>

```

The Test Apex Class (if deploying in Production) should look like this:

```

/**
 * @description Test Class for the OpportunityCMController Class
 */
@isTest
public with sharing class OpportunityCMControllerTest {
    /**
     * @description setup - test data setup method
     */
    @testSetup
    public static void setup(){
        List<Account> testAccounts = new List<Account>();
        List<Opportunity> testOpps = new List<Opportunity>();
    }
    /***/ Test Account record creation ***/
    /***/ Fields shown here are required for a Salesforce Developer Org ***/
    /***/ Requires updates for any required fields you may have ***/
}

```



```

Account a = new Account();
    a.Name = 'ABC Company, Inc.';
    a.BillingState = 'CO';
    a.NumberOfLocations__c = 1;
    testAccounts.add(a);
Account acct = new Account();
    acct.BillingState = 'OR';
    acct.NumberOfLocations__c = 2;
    acct.Name = 'XYZ Company, Inc.';
    testAccounts.add(acct);
/**** end test account records ****/
/**** Test Opportunity record creation ****/
/**** Fields shown here are required for a Salesforce Developer Org ****/
/**** Requires updates for any required fields you may have ****/
    Opportunity opp1 = new Opportunity();
    opp1.Name = 'Test Opp 1';
    opp1.CloseDate = Date.newInstance(2020, 01, 01);
    opp1.StageName = 'Negotiating';
    opp1.Amount = 1;
    opp1.AccountId = a.Id;
    testOpps.add(opp1);
    Opportunity opp2 = new Opportunity();
    opp2.Name = 'Test Opp 2';
    opp2.CloseDate = Date.newInstance(2020, 02, 02);
    opp2.StageName = 'Negotiating';
    opp2.Amount = 2;
    opp2.AccountId = a.Id;
    testOpps.add(opp2);
/**** end test Opportunity records ****/
/**** Insert the Test Records ****/
    insert testAccounts;
    insert testOpps;
}
/**
 * @description testMethod1 - Test Happy Path of code
 */
public static testMethod void testMethod1() {
    List<sObject> opps = (List<Opportunity>)[Select ID From Opportunity LIMIT 2];
    System.debug('***** opps: ' + opps);
    ApexPages.StandardSetController ssc = new ApexPages.StandardSetController(opps);
    ssc.setSelected(opps);
    OpportunityCMController ocmc = new OpportunityCMController(ssc);
    ocmc.prepareCMUrl();
/**** Actually run the test ****/
/**** 100% coverage based upon "OpportunityCMController" Class provided on Conga's Support Site ****/
    Test.startTest();
    PageReference pr = ocmc.prepareCMUrl();
    Test.stopTest();

```

```
}  
}
```


Sample Query Dataset referenced in the Apex Class controlled:

```
select name, amount, stagename from opportunity where id IN ({pv0})
```

## Critical Update Actions for Composer Versions

A list of Composer all versions in descending order and the required actions related to the Salesforce Winter '20 Critical Update.

A critical update introduced in the Salesforce Winter '20 Release requires admins to provide users with additional custom metadata types to continue using Composer successfully. Depending on the version of Composer installed in your Salesforce org, admins must complete the action items listed in the table below. For more information about the critical update in relation to Composer, see [Required Profile Permissions for Winter '20](#).

 Not all versions of Composer require users to have access to custom metadata types. Older versions of Composer do not require any action.

## Check What Version of Composer Is Installed


To check what version of Composer is installed in an org:

1. Navigate to **Salesforce Setup**.
2. In the Quick Find search box, type in **Installed Packages**.
3. Click **Installed Packages**.
4. Locate the Conga Composer Installed Package and the version number is display to the right in the Version Number column.



## Required Custom Metadata Types for Your Version of Composer

See the corresponding custom metadata types required for your installed version of Composer. For detailed steps on assigning the required custom metadata types to users, see [Required Profile Permissions for Winter '20](#).

 The table below lists all versions of Composer in descending order (newest to oldest).

Composer Version	Required Custom Metadata Types
8.179	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.178	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.177	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.176	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.175	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.174	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>

Composer Version	Required Custom Metadata Types
8.173	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.172	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.171	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.170	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.169	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.168	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.167	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.166	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>

Composer Version	Required Custom Metadata Types
8.165	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.164	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.163	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.162	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Conga Solution Export Environment</li> <li>• APXTConga4.Conga Solution Export Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.161	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.160	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.159	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.158	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.157	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.156	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.155	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>

Composer Version	Required Custom Metadata Types
8.154	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.153	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.152	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.151	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.150	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.149	<ul style="list-style-type: none"> <li>• APXTConga4.Composer Setting</li> <li>• APXTConga4.Connected App Setting</li> </ul>
8.148	Access to custom metadata types for Composer is not required.
8.147	Access to custom metadata types for Composer is not required.
8.146	Access to custom metadata types for Composer is not required.
8.145	Access to custom metadata types for Composer is not required.
8.144	Access to custom metadata types for Composer is not required.
8.143	Access to custom metadata types for Composer is not required.
8.141	Access to custom metadata types for Composer is not required.

Composer Version	Required Custom Metadata Types
8.140	Access to custom metadata types for Composer is not required.
8.139	Access to custom metadata types for Composer is not required.
8.138	Access to custom metadata types for Composer is not required.
8.137	Access to custom metadata types for Composer is not required.
8.136	Access to custom metadata types for Composer is not required.
8.135	Access to custom metadata types for Composer is not required.
8.134	Access to custom metadata types for Composer is not required.
8.133	Access to custom metadata types for Composer is not required.
8.132	Access to custom metadata types for Composer is not required.
8.131	Access to custom metadata types for Composer is not required.
8.130	Access to custom metadata types for Composer is not required.
8.129	Access to custom metadata types for Composer is not required.
8.128	Access to custom metadata types for Composer is not required.

Composer Version	Required Custom Metadata Types
8.127	Access to custom metadata types for Composer is not required.
8.126	Access to custom metadata types for Composer is not required.
8.125	Access to custom metadata types for Composer is not required.
8.124	Access to custom metadata types for Composer is not required.
8.123	Access to custom metadata types for Composer is not required.
8.122	Access to custom metadata types for Composer is not required.
8.121	Access to custom metadata types for Composer is not required.
8.120	Access to custom metadata types for Composer is not required.
8.119	Access to custom metadata types for Composer is not required.
8.118	Access to custom metadata types for Composer is not required.
8.117	Access to custom metadata types for Composer is not required.
8.116	Access to custom metadata types for Composer is not required.
8.115	Access to custom metadata types for Composer is not required.



Composer Version	Required Custom Metadata Types
8.114	Access to custom metadata types for Composer is not required.
8.113	Access to custom metadata types for Composer is not required.
8.112	Access to custom metadata types for Composer is not required.
8.111	Access to custom metadata types for Composer is not required.
8.10	Access to custom metadata types for Composer is not required.
8.9	Access to custom metadata types for Composer is not required.
8.8	Access to custom metadata types for Composer is not required.
8.7	Access to custom metadata types for Composer is not required.
8.6	Access to custom metadata types for Composer is not required.
8.5	Access to custom metadata types for Composer is not required.
8.4	Access to custom metadata types for Composer is not required.
8.3	Access to custom metadata types for Composer is not required.
8.2	Access to custom metadata types for Composer is not required.

Composer Version	Required Custom Metadata Types
8.1	Access to custom metadata types for Composer is not required.
8	Access to custom metadata types for Composer is not required.
7.2	Access to custom metadata types for Composer is not required.
7.1	Access to custom metadata types for Composer is not required.
7	Access to custom metadata types for Composer is not required.
1.9	Access to custom metadata types for Composer is not required.
1.8	Access to custom metadata types for Composer is not required.
1.7	Access to custom metadata types for Composer is not required.
1.6	Access to custom metadata types for Composer is not required.
1.5	Access to custom metadata types for Composer is not required.
1.4	Access to custom metadata types for Composer is not required.
1.3	Access to custom metadata types for Composer is not required.
1.2	Access to custom metadata types for Composer is not required.

Composer Version	Required Custom Metadata Types
1.1	Access to custom metadata types for Composer is not required.
1	Access to custom metadata types for Composer is not required.

## Data appears on the QueryDataX dataset for some users, but not others. Why?

This issue can have multiple causes:

- Restrictions on the Conga Query Object. Verify that all users have at least Read access to the Conga Query Object.
- Field Level Security issues. Users must have access to all fields referenced on the Conga Query Object.

## Data appears on the ReportDataX dataset for some users, but not others. Why?

This issue can have multiple causes:

- Folder restrictions on the Conga Composer Reports folder.
  - Verify that all users have at least Read access to this folder on the Reports tab.
- Field Level Security issues.
  - Users must have access to all objects or fields referenced in the report.

## Differences between Conga Composer versions

Depending on where you look, you might see two different Conga Composer version numbers. Here is why:

### App Exchange Managed Package

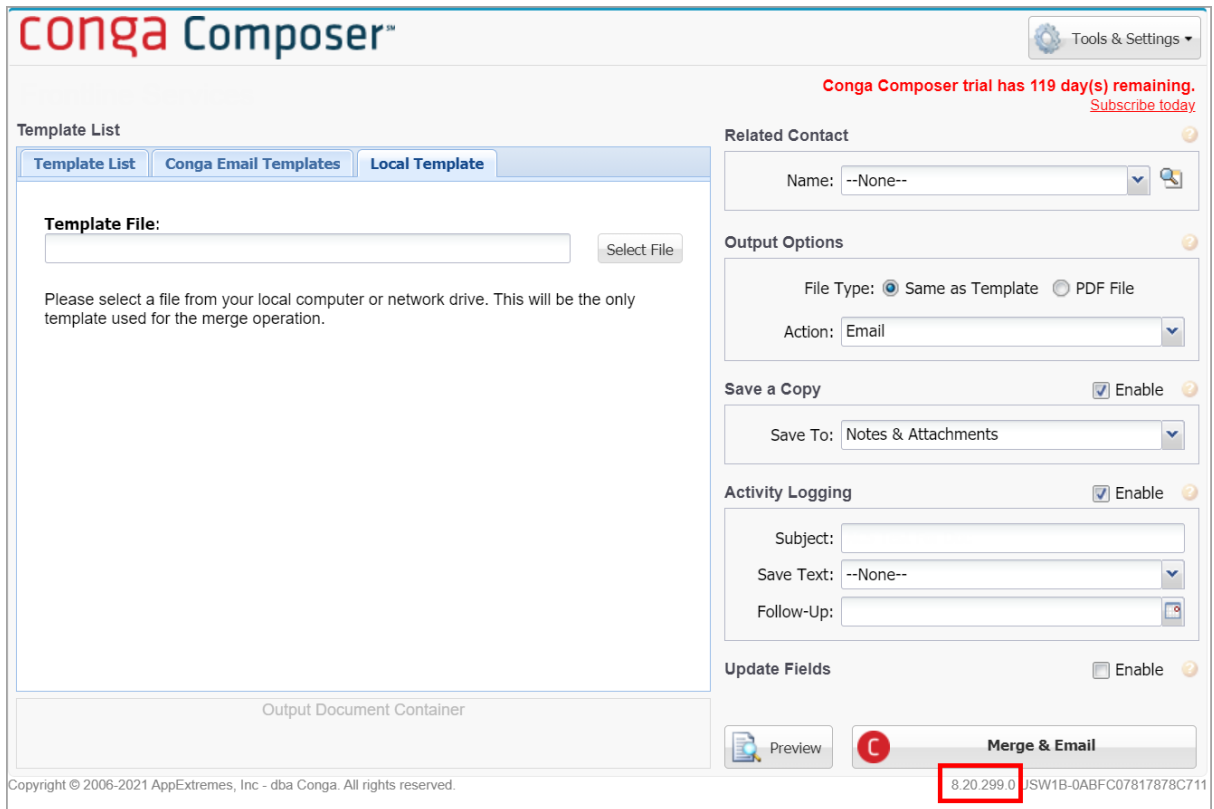
When you download Composer through the Salesforce App Exchange, you are installing a managed package. When Composer version 7 or 8 is referenced, the managed package is called. To learn more, see [Here is how you determine the version number](#) for a managed

package. Managed packages offer new features or changes for Composer, and you have control over when you install it.

## Server Version

Defect fixes and backend changes are automatically released into Production Orgs. These releases are what Release Notes refer to, and why release note numbers might not match the version number for your installed managed package. All users run Composer on the same server version. To check your Conga Composer server version:

1. Launch **Conga Composer Solution**.
2. The current version number is in the lower right-hand corner of the Conga Composer console.



## Disable Composer 7 Triggers

Disable Composer 7 triggers to reduce excessive API calls.

**Enabling the Disable C7 Triggers custom setting is optional and is not required.**

Enable the Disable C7 Triggers custom setting to reduce the total amount of API calls used by Conga Composer.

1. Navigate to **Salesforce Setup**.
2. Type in "Custom Settings" in the Quick Find search box and click **Custom Settings**.
3. Click **Conga Composer Settings**.
4. Click the **Manage button**.
5. Click **New**.
6. Enter a value for the Name field, such as Disable Composer 7 Triggers.
7. Check the Disable C7 Triggers checkbox field.
8. Click **Save**.

## Document not Merging to SpringCM

### Issue

If you click Merge & SpringCM and Composer looks like it ran successfully, but no document gets delivered to SpringCM, there is likely an error occurring in the background where Composer is not able to create a new folder in SpringCM for the merged document.

### What to Do

Configure SpringCM for the Master Object (the object your Composer button or link is on). To do so, follow these steps:

1. Navigate to the SpringCM Setup tab in Salesforce.
2. Check to see if the Master Object is on the Map Salesforce Objects to SpringCM Folders list.
3. If not, click **Map New Object**, type in the name of the object, click **Insert Field**. This wizard allows you to specify the naming convention for the folder Composer will save documents into when merging to SpringCM.
4. When done, click **Next**, and drag into the hierarchy of folders seen in the Your SpringCM Folder Directory section.
5. Click **Add to Page** on the right side of the newly created entry in the Map Salesforce Objects to SpringCM Folders section.
6. Navigate to the Page Layout for your Master Object.
7. Go to the Visualforce Pages section of the page layout editor. Add SpringCM EOS... Visualforce page to the layout.
8. Click **Save**.

These steps will allow you to run Composer on a standard or custom object and merge to SpringCM.

## Removing Duplicate Conga Composer Setup Tab

Remove the duplicate Conga Composer Setup tab.

Upgrading from Composer version 8.148, and all Composer versions preceding version 8.148, to the latest versions creates duplicate Conga Composer Setup tabs in Salesforce. The old Conga Composer Setup tab is no longer necessary and is easily deletable.

**i Duplicate Conga Composer Setup tabs are not created when upgrading from Composer versions newer than Composer version 8.148.**

It is highly suggested to remove the old Conga Composer Setup tab to ensure that users leverage the [new Conga Composer Setup Menu](#).

Follow the steps below to remove the duplicate Conga Composer Setup tab.

1. Navigate to Setup in Salesforce.
2. Type in Tabs in the Quick Find search box.
3. Click **Tags** under the User Interface section.
4. Scroll down to the Web Tabs section.
5. Click **Del** (delete) next to Conga Composer Setup in the Web Tabs section.

**i** Only one Conga Composer Setup tab under the Web Tabs section provides the Del option to delete the tab. This tab is deletable because it no longer managed after upgrading to the latest version of Composer. The new Conga Composer Setup Menu is not deletable because it is part of the managed package.

6. Click **Ok**.

After deleting the unmanaged Conga Composer Setup tab, users can only access the new Conga Composer Setup menu.

## Duplicate solutions record created where solution pack installed

Conga Quick Start is creating duplicate solutions record when creating a pre-built solution in orgs where a Solution Pack has previously been installed.

## Problem

Conga Quick Start is creating duplicate solutions records when creating a pre-built solution in orgs where a Solution Pack has previously been installed. The duplicate Composer solution contains duplicated parameters.

## Causes

There is an issue with duplication of parameters and tabs if you upgrade from a previous version of Composer to version 8.15 or later.

## Workarounds

Delete any duplications.

# Edit My Own Posts Permission Required to Post to Chatter

## Issue

If users do not have Edit My Own Posts enabled on their profile, they will have issues saving a document to Chatter (&SC1=Chatter) when running Conga Composer Release 8 in background mode. A Chatter post will be logged, but there will be an error message instead of a document that says:

```
'Error creating FeedAttachment 0<br />Error code is:  
INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITYError message: insufficient access rights on cross-  
reference id'
```

## Workaround

To correct this, enable Edit My Own Posts in System Wide Permissions (see below) for all affected user profiles.

# Error Using Google Drive Integration with Composer Templates

## Issue

When using the Salesforce Google Drive integration with Conga Composer Templates, users receive an error message saying "Template ID does not exist and will be omitted."

## What to Do

If using a Google Doc as your template, you must download the template in a Microsoft Office file format (such as .docx) before uploading to Salesforce. Do NOT upload the document using the Google Doc button that is part of Salesforce Google Drive integration. You can convert back to a Google Doc post-merge using the &DS7=19 parameter.

## Excel to PDF - Different Values

Error message. Outputting an Excel doc to PDF.

## Error Message

Excel to PDF - Different Values

## Cause

Outputting an Excel doc to PDF and the Excel document provides the correct values, but the PDF version (using FP0 parameter) shows different values in some of the cells.

If you open the Excel document and do not click the Enable Editing, the values will match up between the PDF and Excel output. It's only when you select Enable Editing that Excel runs the calculations and updates.



## What to Do

To have the formulas calculated when outputting into PDF, they will all need to have the prefix of `&=` if the formula only shows once or `&=&` if the formula repeats down the rows.

```
&=SUM(Z15*P49)-P49
```

## Fields not Copying to the Clipboard in Conga Composer

### Issue

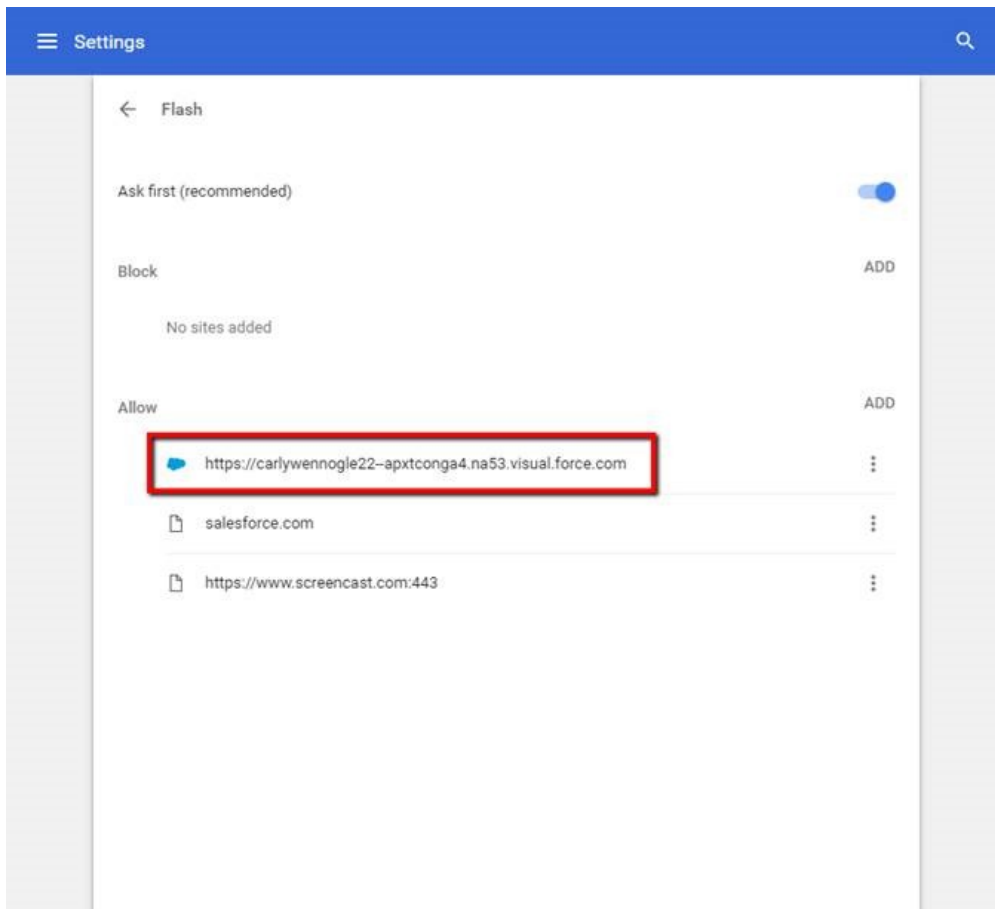
Firefox and Chrome recently updated their security settings to restrict the use of Flash. This disables the copy and paste function in Conga Template Builder.

### Fix

- Use the blue arrows to move fields to the template Canvas and then highlight the field, right-click, and select Copy.
- Install the flash plugin for [Firefox](#) or [Chrome](#).

For Chrome: Add the browser site from the Template Builder pop up window in Flash settings.

For example, *<https://salesforce.com-apxtconga4.na53.visual.force.com>*. Type *chrome://settings/content* into your Chrome browser and navigate to the Flash settings. Add your browser site into the list of allowable sites:



## Fields Won't Display Even After Permission Is Granted in Salesforce to Access that Field (Release 8)

In order to reference a Salesforce field in Conga Composer, a user must have permission (read/visible access at minimum) within Salesforce to access that field.

If a user has access to a field, but that access is later removed, the Conga template may not be able to access that field on the user's behalf. We recommend uploading a new version of the template to force the re-discovery of fields for all users.

## Field Not Merging in PowerPoint Template

A common reason merge fields do not merge at all or do not merge as expected in PowerPoint templates is because there are multiple merge fields in the same text box.

## Cause

A common reason merge fields do not merge at all or do not merge as expected in PowerPoint templates is because there are multiple merge fields in the same text box.

## What to Do

Put all merge fields into their own text boxes. See [Add a Merge Field to a PowerPoint Template](#) for more information.

The GroupRepeat function does merge as expected.

## How do I know what version of Composer I'm using?

Determine your Composer version number.

To determine your Composer version number (in Salesforce classic):

1. Navigate to Salesforce Setup.
2. Click **Installed Packages**.
3. Find Conga Composer on the list. Look at the version number.

## How do I move my Conga Composer subscription from one Salesforce instance to another?

Customers may have to move from one Salesforce instance to another (e.g. if their company is acquired, if they're merging Salesforce instances) and need their Conga Composer, Conga Batch, and/or Conga Trigger subscription(s) to move with them.

To move your Conga subscription from one Salesforce instance to another, please take the following steps:


Do not uninstall Conga Composer (or Conga Batch, or Conga Trigger) from your old Salesforce instance. Do install the desired Conga app in your new Salesforce instance. You can do so from the [Product Installation Links page](#).

 Conga Trigger does not require the installation of an app

Activate Conga Composer in your new Salesforce instance by clicking on the Conga Mail Merge tab in Salesforce (you may need to navigate to your All Tabs view). Activate Conga Batch by clicking the Conga Batch tab.

Contact Support to confirm you've completed steps 3 and 4. Include the Salesforce.com Organization IDs for your old Salesforce instance (where you want the subscription moved from) and your new one (where you want the subscription moved to). The Conga support team will then contact the Provisioning team to move your user licenses and subscription details from your old Salesforce instance to your new one. Conga support can also keep the old instance active with Conga Composer, Conga Batch, or Conga Trigger access.

Conga support will send you an email confirming that your subscription and licenses have been moved to the new instance.

 You will need to recreate all of your Conga Composer, Conga Batch, and Conga Trigger solutions in your new Salesforce instance. It is especially important to keep your old Salesforce instance active so you can refer to the original Conga solutions there while you rebuild them in your new Salesforce instance.

Remember that all Salesforce IDs (report IDs, Conga Template IDs, Conga Query IDs, etc.) will change in the new instance. You will need to update all references to these IDs in your Conga Composer, Conga Batch, and Conga Trigger buttons, links, fields, and general solutions.

Contact Conga Support with questions or if you require assistance.

## How to use Extended Ping and Trace Route Instructions for Conga

### Extended Ping & Trace Route Instructions

In situations where timeout or connection issues are experienced, it may be helpful to run an extended ping (>100 pings) to see whether some packets are dropped. This will create a text file in the current directory that can be emailed to you.

#### Extended Ping Instructions

1. Open a command window from **Windows Start** → **Run** → **CMD** and then hit the Enter key on your keyboard.
2. Type "CD Desktop" and then hit **Enter**.

3. Enter the following text: `ping www.congamerge.com -n 100 > apxt_ping_results.txt` and hit **Enter**.
4. The results from this extended ping test will be in a text file on your desktop called `apxt_ping_results.txt`

## Trace Route Instructions

1. In a Command window (you may use the same one you used in the above instructions for the extended ping), enter the following text: `tracert www.congamerge.com > apxt_tracert_results.txt`
2. Hit **Enter**.
3. The results from this trace route test will be in a text file on your desktop called `apxt_tracert_results.txt`

## Images in Word

You can merge images into Composer templates. Here are two methods for merging images into a Word template:

1. [Image Field](#)
2. [Include Picture \(for publicly accessible images\)](#)

## Insufficient Privileges Error when a System Admin launches an OAuth button as a user

### Issue

System Administrator receives an "Insufficient Privileges" error when launching an OAuth button while logged in as a user (even though the user does not receive the error).

This is due to a Salesforce security update that does not allow System Administrators to log in as a user and launch an OAuth button.

### What to Do

A workaround is to add a session ID to the button only when you need to login as the user. We recommend removing the session ID once you are done.

- Button with session ID: `/apex/APXTConga4__Conga_Composer ?sessionId={!API.Session_ID} &serverUrl={!API.Partner_Server_URL_370}&id=`
- Button without session ID: `/apex/APXTConga4__Conga_Composer ?serverUrl={!API.Partner_Server_URL_370}&id=`

## Log Into One Salesforce Instance or as One User per Browser

Best practice. Log into only one Salesforce instance and as one user per internet browser.

Conga recommends logging into only one Salesforce instance and as one user per internet browser. This behavior optimizes Conga product configuration and data merging, ensuring correct results. If logging into multiple Salesforce instances or as multiple users is required, run each different instance and/or user in a unique browser. For example, Google Chrome runs the production instance while Firefox runs the sandbox instance.

Many Salesforce customers run multiple instances (production, sandbox, dev) of Salesforce in order to properly maintain production data while also running tests. Similarly, many system administrators will utilize the Salesforce function to log in as other users in order to troubleshoot those users' issues.

While these are useful Salesforce features, they can wreak havoc on Conga product configuration. Some of the issues that can result when logged into multiple Salesforce instances within the same browser or logged in as multiple users within the same browser are:

- Incorrect data merging to documents
- Incorrect user data
- Inconsistent privileges (access to reports, objects, etc.)
- Difficulty configuring OAuth tokens

## Merge Fields Displaying Unexpected Results

Due to a recent change to the way Salesforce exports reports, you may experience unexpected behavior when merging data from formula fields retrieved by Salesforce reports.

**How do I know if I am affected?**

In your output document, you may see affected merge fields display "&amp;" or "&quot;" etc. as the merged value. This is due to the change referenced above.

### How do I fix it?

Add an HTML prefix to all formula fields. To do so, follow the steps below.

If using traditional merge fields, such as `<<MergeField>>`, do the following:

1. Toggle the merge field open by right-clicking on the merge field and selecting Toggle Field Codes.
2. Modify the merge field to include HTML: as a prefix to the field name. For example, a custom formula field for the SLA Plan would need to be modified as follows (in bold):

```
{ MERGEFIELD HTML:SLA_PLAN \* MERGEFORMAT }
```

If using text-based merge fields, such as `{{Merge_Field}}`, do the following:

- Modify the merge field to include HTML: as a prefix to the field name. For example, a custom formula field for the SLA Plan would need to be modified as follows (in bold):

```
{{HTML:SLA_Plan}}
```

For more information about this change, visit [https://help.salesforce.com/HTViewSolution?id=000239661&language=en\\_US](https://help.salesforce.com/HTViewSolution?id=000239661&language=en_US).

## One or more documents failed to convert to PDF

### Error Message

Custom font needs to be uploaded.

### Cause #1

A document merges successfully when selecting Same as template in Composer, but when outputting to a PDF file, the following error appears: One or more documents failed to convert to PDF.

A common reason for this is using a custom font in the selected template. While this doesn't cause issues when merging Same as template, it will error out when converting to PDF.

### What to Do

The resolution is to Contact Support to get the custom fonts uploaded. Customers need to provide an .otf or .ttf font file.

## Cause #2

Two Word documents merge successfully when merged separately. Although when assembling two documents together using either the &AWD=1 or &DefaultPDF=1 parameters, the following error appears: One or more documents failed to convert to PDF.

If the &Defaultpdf=1 parameter is removed from the composer button URL, the following error appears when opening the Word document: Xml parsing error.

## What to Do

Check the compatibility settings on each Word document and make sure it is using the most recent file version. Save each document as a .docx filetype.

## Opplineltems Dataset Missing in Template Builder

Unable to see Opplineltems dataset in Composer Template Builder.

## Issue

When launching Composer from an Opportunity the Opplineltems dataset is not in the Template Builder or View Data Workbook.

## Cause

It is likely that fields with nearly identical API field names are present on the Opportunity Product Object.

For example, if you create a custom field with an API name of Quantity\_\_c on the Opportunity Product object, Composer will try to create two fields with the name Quantity because there is also a standard field on the Opportunity Product called Quantity. When Composer tries to create the same field name in the Template Builder twice, the entire dataset is omitted.

While this data does not display in the Template Builder, it will still merge successfully into an output document.



## What to Do

There are two known solutions:

1. Rename the API Field Name of the duplicate field.

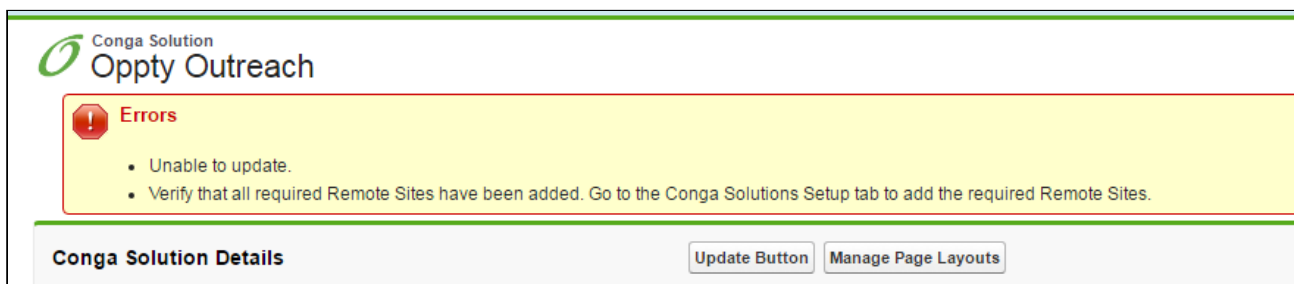
⚠ Exercise caution when renaming API Field Names. This can cause field names to change in the Template Builder, which means any template that references the old field name would need to be updated.

1. Another workaround is to rebuild the standard Opplineltems dataset using a Conga Query.

## Remote Site Error in Conga Solution Manager

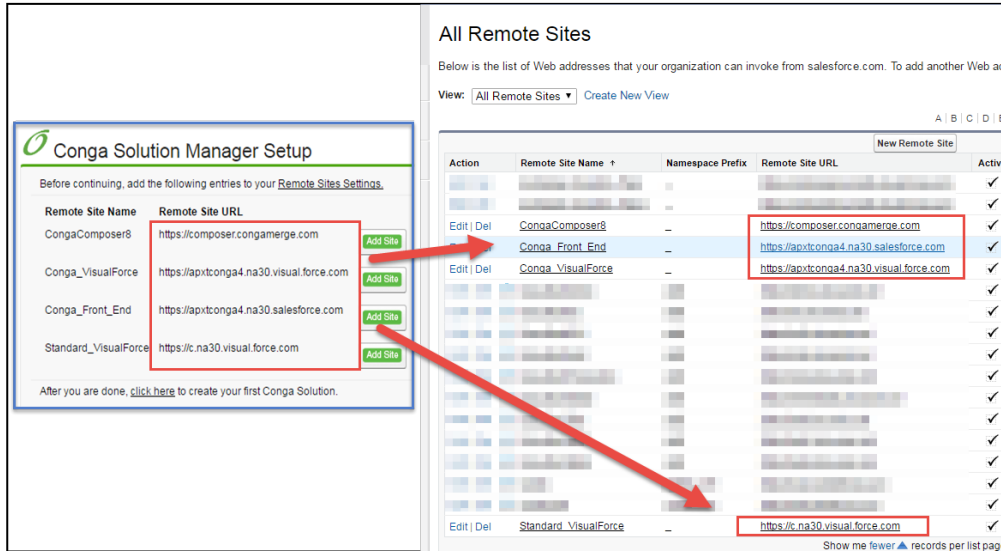
### Cause

If Remote Sites are activated while multiple instances of Salesforce are open it is possible for the wrong Remote Sites to be saved. If that happens, you will receive the following error:

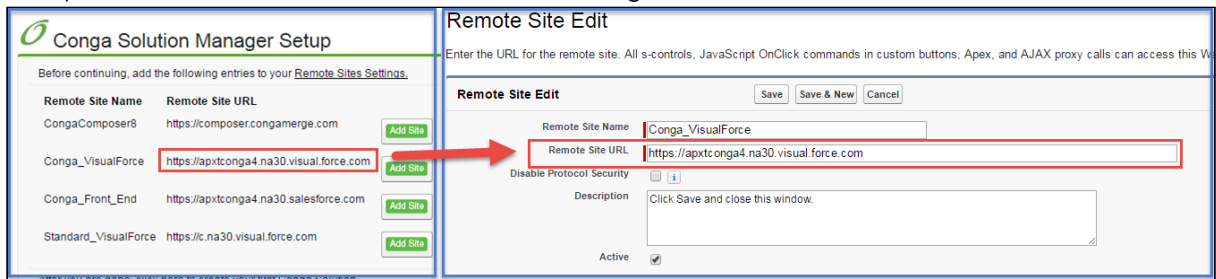


## What to Do

1. Open Conga Solution Setup in one tab, and Salesforce Setup in another.
2. In Salesforce Setup, search for "Remote" and click **Remote Site Settings**.
3. Make sure the Remote Sites in Conga Solution Setup match the Remote Sites in Salesforce Setup (see below).



- 4. If they do not match, click **Edit** next to the Remote Site link in Salesforce Setup (in the image above on the right) and copy the Remote Site URL from Conga Solution Setup into the Remote Site URL field (see image below).



- 5. Click **Save**.
- 6. After updating all four Remote Sites, try updating the Conga Solution record again.

## Removing duplicate tabs

If you upgraded your Composer version, you might see duplicate Composer tabs.

### Question

Why do I see duplicate tabs in on my Composer Set up page?

### Causes

If you upgraded your Composer version, you might see duplicate Composer tabs in the Set up tab.

## Workarounds

Delete duplicate tabs by selecting DEL next to the tab you want to remove.

## Shipping and Billing Merge Fields

The shippingaddress and billingaddress fields in the template builder do not work and often bring in too much or poorly formatted information.

### Issue

The SHIPPINGADDRESS and BILLINGADDRESS fields in the Template Builder do not merge data as expected and often merge in too much or poorly formatted information.

### What To Do

- Option 1: Use the SHIPPINGSTREET and BILLINGSTREET fields instead.
- Option 2: (Recommended) Use the individual merge fields which provide better control and placement in the Composer document. For example, Shipping Street, Shipping City, Shipping State, Shipping Postal.

## Templates not displaying or showing fewer than expected in Composer (Release 8)

Templates do not display or show fewer templates than expected in Composer Release 8.

### Problem

Templates are not displaying in Composer Release 8 or showing fewer templates than you expect.

### Possible Causes and Workarounds

- Apply appropriate object and field level security to the Conga Templates custom object. See Profile Permissions Needed to Use Composer for more information.

- The TemplateGroup parameter filters the list of file templates that appear in the Conga Composer dialog box to those that match certain criteria. Remove this parameter so the list is not filtered and your templates display.
- If the Used Data Space and Used File Space limits are exceeded in your Salesforce org, this can cause templates to not display in the Composer window. This information can be found in Salesforce Setup
  - Company Profile
  - Company Information. If these fields are over 100%, Composer 8 might not be able to display all available templates.
- Ensure there are no colons or other illegal characters in the name of the file or attachment on any Conga template record. If colons or other illegal characters are included, Composer omits that template and any subsequent template that would otherwise display in the template list.

**Example of illegal characters:**

Character Name	Character	Encoded Value
Colon	:	%3A
Hash	#	%23
Open Parenthesis	(	%28
Close Parenthesis	)	%29
Double Quotes	"	%22
Percentage	%	%25
Backslash	\	%5C

- Make sure only one template repository is selected in Composer Setup.

If you use Salesforce Communication Templates, you must select that option.

# Template is not populating in template list on the Conga dialog box or is not visible in Conga Template Manager

## Issue

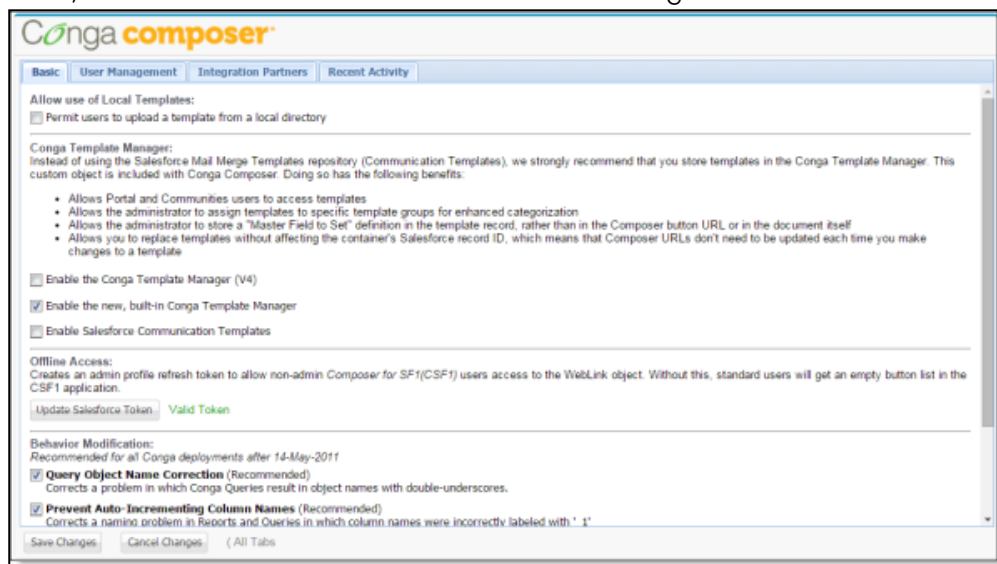
Often times, the system administrator can see and access Conga Templates just fine but some or all of the end-users cannot.

You must have a Conga user license to use Conga Composer. You will need to assign all of your users to a Conga user license so that your users can access and use Conga Composer. For further information, see this article on Conga Composer user management.

## What to Do

Check the following items to troubleshoot this issue:

- Is Conga Template Manager enabled?
  - **Conga Setup → Basic tab → Conga Template Manager**
- Ensure your version (either V4 or V5) of the Conga Template Manager is enabled.  
Note, do not check the box for both versions they do not work at the same time.



- Is the Conga Template object deployed?
  - **Salesforce Setup → App Setup → Create → Object → Conga Template → Deployment Status**

- Deployment Status must be "Deployed" for non-System Administrator users to access the object
- User's Profile permissions
  - **Salesforce Setup → Administration Setup → Manage Users → Profiles**
  - The user's profile must have at least "Read" access to the Conga Template object (under Custom Object Permissions)
- Does the user have access to the fields on the Conga Template object?
  - **Salesforce Setup → App Setup → Create → Objects → Conga Template → Custom Fields & Relationships**
  - Each field on the Conga Template object must be at least "Visible" (optionally, it can also be "Read-Only") to the user's profile
  - Click on a Field Label, then click the Set Field-Level Security button to view the level of access that each profile has for that field (you'll need to do this for every field on the Conga Template object)
- Are Sharing Settings being used?
  - **Salesforce Setup → Administration Setup → Security Controls → Sharing Settings → Conga Template**
  - Conga Template object needs to be set to "Public Read Only" or "Public Read/Write" (NOT "Private")
- Is Template Group being used?
  - Using the &TemplateGroup parameter in your composer button URL will limit the list of templates that are displayed on the Conga Composer dialog box to only those that contain the specified Template Group.

## Troubleshooting Font Styles


When designing Conga Word templates, font choice is important. Select a font from the All Fonts list to ensure your font will translate to your merged output file. If you choose a (Heading) or (Body) font style the font might not appear correctly in the merged output file.

If you choose a (Heading) or (Body) font style the font might not appear correctly in the merged output file.

When merging a Word doc to a PDF output and using custom .OTF font files (that are both installed locally and to our servers), the output generated by Composer will populate fine, but if you try to print the document on a Mac, the text with the .OTF font file applied will print out blank. If a customer is printing a PDF using custom fonts generated by Composer on a Mac, make sure they are using custom .TTF fonts.

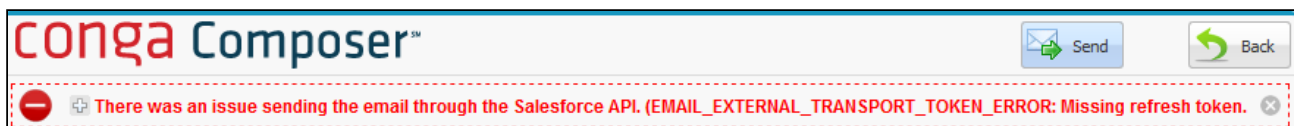
# Using Send through External Email Services with Conga Composer

Use the Send through External Email Services functionality with Conga Composer

 The actions in this article are only required if Send through External Email Services is enabled in your org. If it is not enabled, no action is required.

Salesforce offers optional functionality to send outbound emails from Gmail and Office 365 with the [Send through External Email Services](#) settings in Salesforce Setup.

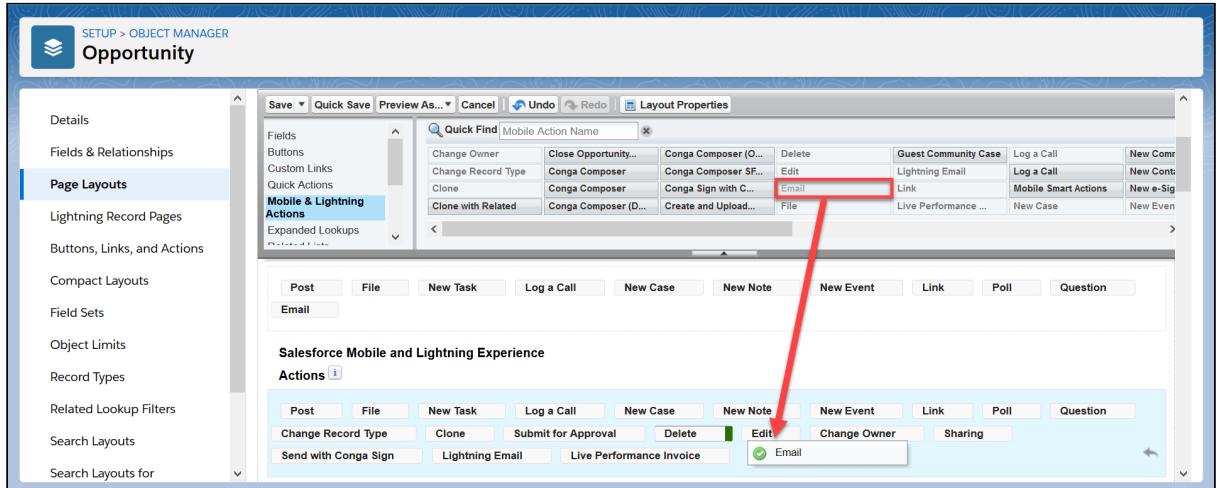
Each Composer user must follow the steps below to authorize external email services and successfully send emails with Conga Composer. If Send through Gmail or Send through Office 365 is enabled and the external email service is not authorized, users see the following error message when sending an email from Composer.



## Prerequisite for Authorizing External Email Service

The native Send Email [Global Quick Action](#) is required to authorize the external email service. Ensure that the Send Email Global Quick Action is added to the Opportunity Page Layout.

1. As an admin, navigate to the Object Manager in Salesforce Setup and click **Opportunity**.
2. Click **Page Layouts** and select the relevant Page Layout.
3. Click **Mobile & Lightning Actions**.
4. Select and drag Email into the Salesforce Mobile and Lightning Experience Actions section of the Page Layout.

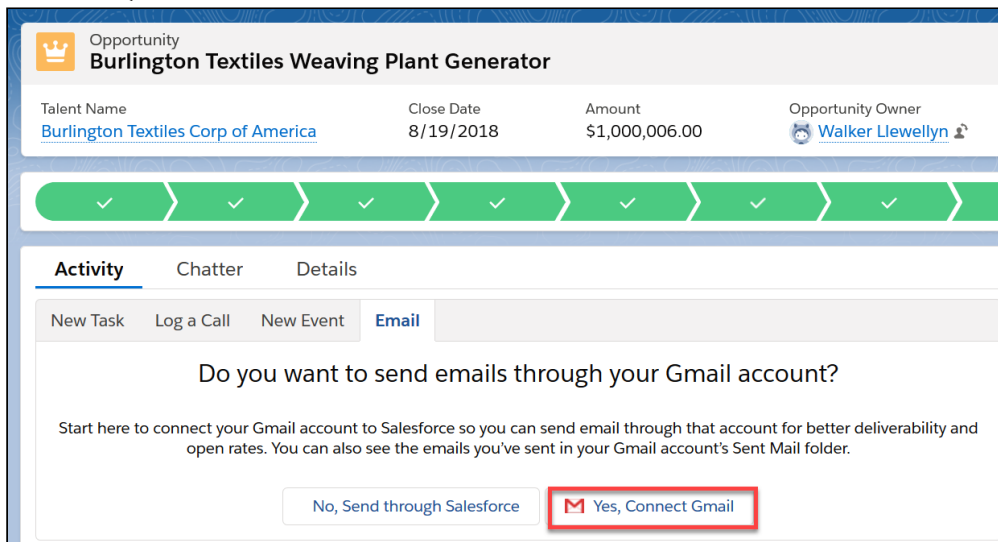


5. Click **Save**.

## Authorize Send through Gmail

**i** Users must use Salesforce Lightning to authorize the external email service. Each individual user must complete the steps below to authorize the external email service and successfully send emails from Conga Composer.

1. As a user, navigate to an Opportunity record in Lightning.
2. Under the Activity tab, click **Email**.
3. Click **Compose** or click in the "Write an email" text box.
4. Click **Yes, Connect Gmail**.



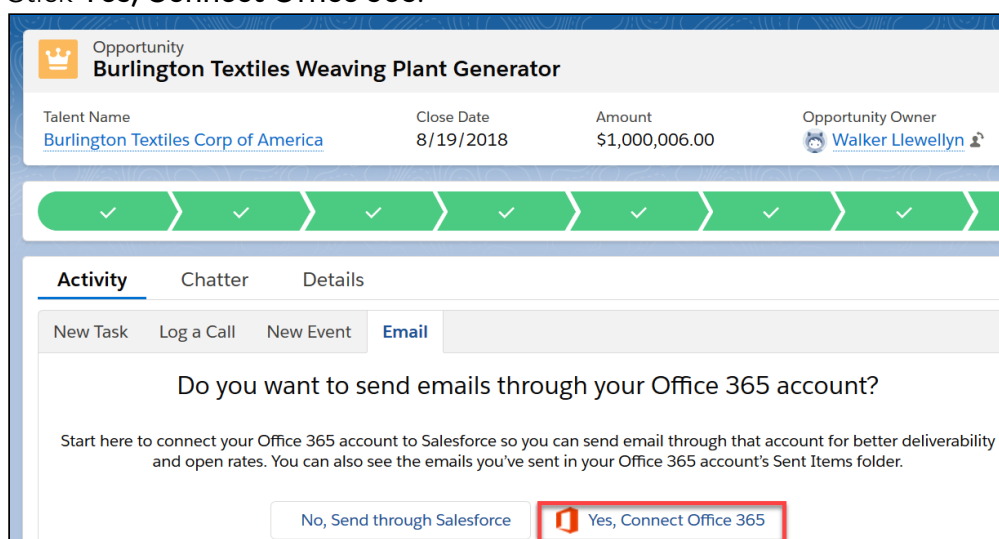
5. Select a Gmail account.
6. Click **Allow**.



## Authorize Send through Office 365

**i** Users must use Salesforce Lightning to authorize the external email service. Each individual user must complete the steps below to authorize the external email service and successfully send emails from Conga Composer.

1. As a user, navigate to an Opportunity record in Lightning.
2. Under the Activity tab, click **Email**.
3. Click **Compose** or click in the "Write an email" text box.
4. Click **Yes, Connect Office 365**.



5. Enter your Office 365 email and click **Next**.
6. Follow the remaining prompted directions to complete the authorization.

**i** These directions vary depending on your organization's Office 365 configuration.

## What is Aspose related to my PDF?

### Question

When I create a PDF, why do I see Aspose as the name of a property and not Conga Composer?

## Causes

Composer uses Aspose cloud-based APIs for document generation, conversion, and automation.

## Where Did My Conga Mail Merge Tab Go? (Release 8)

Missing Conga Mail Merge tab for Composer Release 8.

In Conga Composer Release 8 there is no longer a Conga Mail Merge tab. This means if you are upgrading to Conga Composer Release 8 your Conga Mail Merge tab will disappear. Individual Conga Mail Merge solutions created prior to Conga Composer Release 8 will NOT be changed or removed; you simply need to separately install the new Conga Mail Merge add-on.

Conga Mail Merge will need to be installed as a separate add-on (free of charge) to continue running Conga Mail Merge solutions. To do so, see Installation Links for Conga Products.

## Why didn't my dynamic formula update correctly in my Conga Composer Excel output file?

Why didn't my dynamic formula update correctly in my Conga Composer Excel output file?

## Question or Issue

When I use a Conga dynamic formula in an Excel file and run a merge, the output result displays as #NAME?.

## Related Known Errors and Workarounds

The workaround is to use a regular Excel formula when you encounter the issue, or to manually click in the Excel cell and hit Enter for the results to display as intended.

## Templates appear for administrators but not end-users

The most common reason why templates stored in the Conga Template Manager appear for system administrators, but not for end-users, is that the Conga Template Manager custom object was not deployed. Verify that this setting is enabled on the custom object properties page.

For more information, see [Permissions Needed to Use Composer](#).

## Composer Error Messages

Composer Error Messages:

- Adobe eSign Installation Error
- A problem occurred in the merge process. Please try once more before contacting Support. (Error: 0122) (Release 7)
- Cannot remove because there is no parent (Release 8)
- Composer Authorization Error | 404 Page Cannot Be Found
- Conga Composer Cannot Communicate with Salesforce (Release 7)
- Conga Composer cannot connect with Salesforce because master ID is invalid. (Release 7)
- Conga Composer Expiration Warning
- Conga Composer Settings Error
- Conga Email Staging Record Error
- Error "id parameter is missing" with Composer 8 buttons in Lightning
- Error 0754 The document appears to be corrupted & cannot be loaded (Release 7)
- Error 500 - "Internal Server Error" (Release 7)
- Error appears when clicking security token in Sandbox Orgs
- ERROR at Row:1:Column:40 value of filter criterion for field 'CloseDate' must be of type date and should not be enclosed in quotes (Release 7)
- Error: CreateEnvelope in DocuSignAPI. One or both of Username and Password are invalid. You may need to set &DocuSignEndpoint="AlternateEndpoint, e.g., Demo)
- A problem occurred in the rendering process. Please verify your template is valid (Release 7)
- Found end of mail merge region 'ReportData1 0' without matching start (Release 7)
- Input string was not in the correct format
- Insufficient Access: Current IP Address is not allowed or invalid\_grant: IP restricted
- Insufficient Access to Custom Settings Error

- Invalid Grant: authentication failure
- Invalid master object ID(must be a Salesforce id) (Release 7)
- Invalid PPTX TableGroupStart Field
- Invalid Query Filter Operator
- Invalid Session ID found in SessionHeader (Release 7)
- Merge and Box Errors
- No DocuSign Recipients Defined (Release 7)
- Object reference is not set to an instance of an object (Release 7)
- Object reference is not set to an instance of an object (Release 8)
- One or More Templates Failed to Merge or Operation Was Cancelled
- Ouch... something's definitely gone wrong. Please try again. (Error #1890: ORNF) (Release 7)
- Our Apologies . . . Your Conga URL contained illegal characters (Release 8 and 7)
- Page Cannot Be Found
- Read pdf error: Trailer is not found (Release 7)
- Reduce complexity of SOQL statement to avoid errors
- Report exceeds maximum supported rows from PointMerge (Release 7)
- Report size cannot be determined, either because the report is excessively large, the report is unavailable or the report type is matrix. Please verify the Report Id.
- REQUEST LIMIT EXCEEDED:TotalRequests Limit exceeded
- Server was unable to process request. Value cannot be null. Parameter name: password
- Sign in with Google temporarily disabled for this app. This app has not been verified yet by Google in order to use Google Sign In
- Something's gone wrong! An internal server error has occurred
- Something's not quite right about your data - it can't be displayed in Excel. Column 'id' does not allow nulls. (Release 7)
- Something's wrong with your template (bad mojo). The merge template has serious problems
- Stack Empty (Release 7)
- String not recognized as a valid DateTime
- Template Id(s) do not exist
- Template Id is Invalid
- There hasn't been a file or email template ID defined, background mode cannot continue when trying to use (Release 8)
- There is no row at position 0
- There was an issue getting the Conga Authorization Response
- There was an issue processing the Composer request
- The ',' character, hexadecimal value 0x2C, cannot be included in a name

- The report is invalid. Are you using My Domains or matrix style reports?
- Parameter p3lkid value error
- Unable to parse the query id entries, please check the URL syntax (Release 8)
- Unexpected end of children nodes reached (Release 7)
- URLENCODE function for illegal characters in button URL
- User Management Access Error
- Composer Output Contains Blank Fields
- Your Conga Session has Expired
- Salesforce session expired or the Salesforce API Servers are offline - INVALID SESSION ID or SERVER URL (Release 7 and 8)
- "Conga can't communicate with Salesforce" error
- Collection size exceeds maximum size of 1,000

## Adobe eSign Installation Error

### Cause

Adobe eSign does not appear to be installed or your security settings don't allow access to it. This error usually occurs for an end user after the administrator has completed and tested the solution. The typical cause is that the end user does not have appropriate permissions to work with the object in the solution.

### What to Do

Look at the profile for the end user and ensure that they have appropriate permissions to work with the associated object(s).

**A problem occurred in the merge process. Please try once more before contacting Support. (Error: 0122) (Release 7)**

### Cause

This error usually occurs as a result of a corrupted template file.

## What to Do

1. Review the template you are working with.
2. Ensure that the template you are using has a valid file extension (.doc, .pdf .xls etc.)
3. Verify that the template file can be opened and edited on your local computer.
4. Ensure that the template follows Conga best practices.

## Cannot remove because there is no parent (Release 8)

### Error Message

A carriage return in the TableEnd text-based field will cause the error message.

### Cause

A carriage return in the TableEnd text-based field will cause the error message.

## What to Do

To prevent this error, ensure that there are no carriage returns in the text-based fields in your template.

To check for carriage returns in your Word template turn on the "Show/Hide Paragraph Marks"

## Composer Authorization Error | 404 Page Cannot Be Found

### Cause

The error message is generated when the browser cannot resolve a URL. In Conga, this is generally caused by a malformed URL in your Composer button.

Typical causes can include:

- Spaces within the URL
- Spaces at the end of the URL
- Missing an ampersand (&) between parameters
- Using a URL pointing to an invalid server (such as beta vs. www)

## What to Do

Remove all spaces from the button's URL. Use plus signs instead of spaces in a parameter. Plus signs render as spaces in the Composer merge operation, but do not lead to 404 authorization errors.

Also see [There was an Issue Getting the Conga Authorization Response](#).

# Conga Composer Cannot Communicate with Salesforce (Release 7)

## Error Message

Conga Composer Cannot Communicate with Salesforce (Release 7).

## Cause

Conga Composer is unable to connect to your organization's instance of Salesforce. This may be due to a server issue with Salesforce - click [here](#) to check status of Salesforce instance.

## What to Do

- Ensure "Lock Sessions to IP Address from which they originated" is disabled (Setup > Administration Setup > Security Controls > Session Settings)
- Verify that the Session Timeout value is at least 30 minutes
- Ensure the Conga Composer URL contains an "&Id=" parameter and that it is set to a valid ID
- Don't try to "Preview" a Conga Composer button. You must click it from a Salesforce record

# Conga Composer cannot connect with Salesforce because master ID is invalid. (Release 7)

## Error Message

This error results when the ID designated as master object in the Conga Composer button URL is incorrect.

## Cause

This error results when the ID designated as master object in the Conga Composer button URL is incorrect. This could result from the ID number (e.g. RU-0001) being selected as the master object ID by mistake when Conga needs the record ID (e.g. a00G0000007MAXg) in order to function correctly.

## What to Do

Ensure that the field you have selected in the button URL is the record ID for the record you've put your button on. If you're using a Salesforce standard object as your master object ID, the record ID field will be listed as Account ID, Opportunity ID, Contact ID, etc. If you've put your button on a custom object, the record ID field will be listed as Record ID.

This will be listed as Record ID if you're placing your button on a custom object. The **&id=parameter** designates which object will be your master object.

## Conga Composer Expiration Warning

When launching Conga Composer, you receive an expiration warning.

## Cause

Your Conga Composer licenses are expiring within the number of days listed.




## What to Do

### Trials:

- If you did not purchase Conga Composer yet, but have a trial, engage with your salesperson.

### Paid Customers:

- If this is a Sandbox, Dev, or UAT org, submit a case to extend your licenses to co-term with your subscription end date if the expiration is in 10 days or less.
- If this is your Production Org, and you purchased or renewed Conga Composer already, then submit a case.
- If you recently purchased and installed Conga Composer for the first time, submit a case.

 All new Composer installs in Production or Sandbox Orgs for Composer default to a 14-day trial. Production Orgs default to 10 licenses and Sandboxes default to an Unlimited Site License. ActionGrid, Conga Sign, and Conga Contracts default to a 15 day trial in Production Orgs.

## Conga Composer Settings Error

The following error occurs when navigating to Composer Setup:

- Conga Composer Settings Error.
- There was an issue getting the Composer settings.
- There was an issue contacting the Salesforce servers.
- Check to ensure that 'Lock IPs' is not enabled and that the instance has available API calls.

## What to Do

1. Click **Ok**.
2. Click **Create Salesforce Token**.
3. Refresh the page.

It may take a few minutes to resolve the issue.

# Conga Email Staging Record Error

## Error Message

Conga Email Staging record could not be created, most likely because you don't have access to the object or Create/Read/Edit/Delete for this object on your profile.

## Cause

Conga Email Staging record could not be created, most likely because you don't have access to the object or Create/ Read/Edit/Delete for this object on your profile. (Release 7)

There are two causes for this issue:

- The user does not have full access to the object.
- If it is not a permission issue, it may be a character limit issue. The HTML Body field on the Conga Email Staging object is an SFDC rich text field that has a character limit of 32,000 characters. Take a look at the size of the email template body.

## What to Do

Verify that the user has access to the Conga Email Templates object as well as all fields on this object.

# Error "id parameter is missing" with Composer 8 buttons in Lightning

## Cause

Composer 8 buttons in Salesforce Lightning will produce an id parameter is missing error message when an additional question mark is included in the button URL. This most commonly happens when you use ?pv0= to pass a non-standard field value into the filter of your data source.

For example, the following would cause an error in Lightning:

```
&QueryId=a0441000002PxvVAAS?pv0={!Opportunity.AccountId}
```

 **Tip**

The example above only contains one question mark. Isn't the problem when you have more than one question mark? Yes! Composer buttons also have a question mark in the base URL (which is the first few lines), making the question mark in the above example the second.

## What to Do

Encode the question mark using the URLENCODE function in the Salesforce button, as shown below:

```
&Queryid=a0441000002PxvVAAS{!URLENCODE("?")}pv0{!URLENCODE("=")}{!  
Opportunity.AccountId}
```

## Error 0754 The document appears to be corrupted & cannot be loaded (Release 7)

### Error Message

This error indicates there is something corrupt within your template.

### Cause

This error indicates there is something corrupt within your template.

### What to do

The easiest way to solve this error is to simply save a new version of your template in .doc format instead of .docx. Find more information on different troubleshooting methods see [Error: Something's wrong with your template \(bad mojo\). The merge template has serious problems.](#)

## Error 500 - "Internal Server Error" (Release 7)

### Error Message

This error results when Conga Composer cannot access the report and/or the data within the report.

### Cause

This error results when Conga Composer cannot access the report and/or the data within the report.

### What to Do

Please check the following items to correct the problem:

- If using a Custom Report Type, ensure it is set to "Deployed" and not "In Development".
- Ensure the user's profile has the following permissions: "API Enabled", "Run Reports" and "Export Reports".
- Ensure the user has at least Read access to the folder that contains the Report.
- Ensure that the High Assurance Session Required setting is disabled, allowing Conga Composer to access exported report data. If this setting is enabled, the session security level required to export reports is increased, preventing Conga Composer to access the report data.

## Error appears when clicking security token in Sandbox Orgs

### Problem or Issue

On the Composer Setup page, you see an error when you click **Update Salesforce Token** button in your Sandbox orgs. The error can be ignored because the security token is used for scheduled Batch only.

The screenshot shows the Conga Composer setup interface. At the top, there are tabs for 'Basic', 'User Management', 'Integration Partners', 'Recent Activity', 'API User', and 'External'. The 'Basic' tab is selected. The main content area is divided into sections:

- Allow use of Local Templates:**
  - Permit users to upload a template from a local directory
- Conga Template Manager:**

Instead of using the Salesforce Mail Merge Templates repository (Communication Templates), we have a custom object included with Conga Composer. Doing so has the following benefits:

  - Allows Portal and Communities users to access templates
  - Allows the administrator to assign templates to specific template groups for enhanced categorization
  - Allows the administrator to store a "Master Field to Set" definition in the template record
  - Allows you to replace template content with a record and make changes to a template
  - Enable the Conga Template Manager
  - Enable the new, built-in Conga Template Manager
  - Enable Salesforce Communication
- Offline Access:**

Creates an admin profile refresh token to allow non-admin *Composer for SF1(CSF1)* users access the CSF1 application.

**Update Salesforce Token** Valid Token
- Behavior Modification:**

*Recommended for all Conga deployments after 14-May-2011*

  - Query Object Name Correction** (Recommended)  
Corrects a problem in which Conga Queries result in object names with double-underscores.
  - Prevent Auto-Incrementing Column Names** (Recommended)

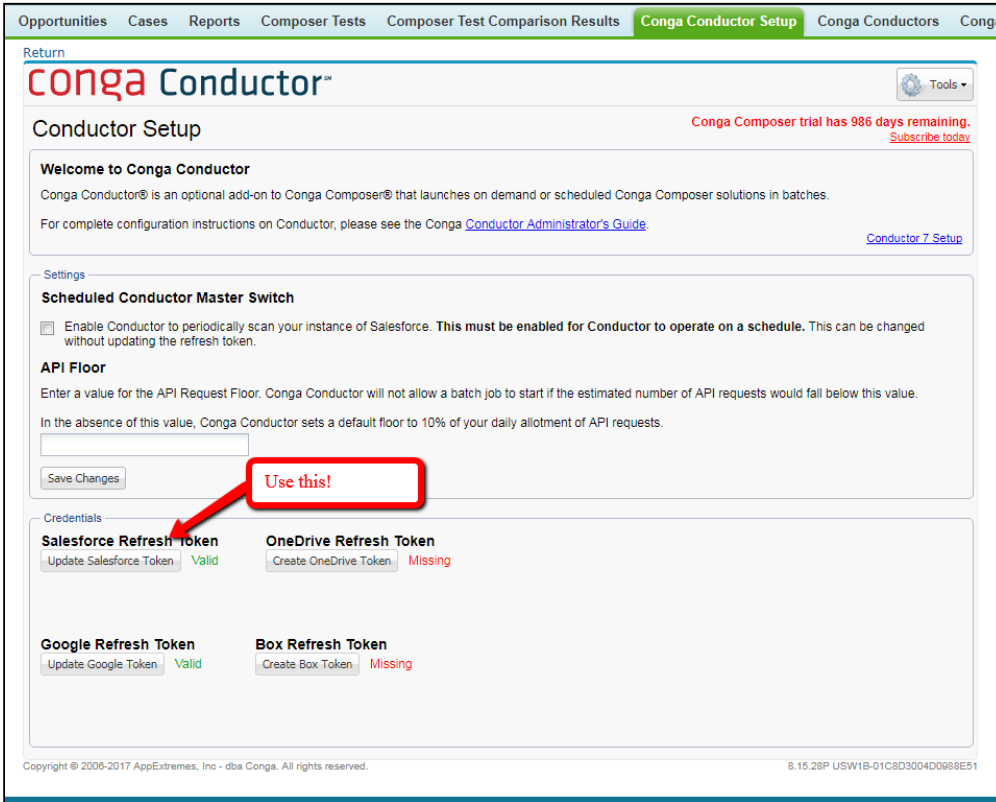
At the bottom, there are buttons for 'Save Changes', 'Cancel Changes', and '( All Tabs )'. A copyright notice at the very bottom reads: 'Copyright © 2008-2017 AppExtremes, Inc - dba Conga. All rights reserved.'

## Possible Causes

There might be code path configuration differences.

## Workarounds

On the Conga Conductor Setup page, click the **Update Salesforce Token** button.



ERROR at Row:1:Column:40 value of filter criterion for field 'CloseDate' must be of type date and should not be enclosed in quotes (Release 7)

## Error Message

This error message occurs because The Conga Query Builder leaves quotation ' marks ' around date fields.

## Cause

The Conga Query Builder is a tool that helps you build SOQL queries. This error message occurs because The Conga Query Builder leaves quotation ' marks ' around date fields.

## What to Do

Remove single or double quotation marks around dates from the generated query as shown below.

Change this:

```
SELECT Amount, Account.Name FROM Opportunity WHERE CloseDate > '2012-12-31'
```

To this:

```
SELECT Amount, Account.Name FROM Opportunity WHERE CloseDate > 2012-12-31
```

**Error: CreateEnvelope in DocuSignAPI. One or both of Username and Password are invalid. You may need to set &DocuSignEndpoint="AlternateEndpoint, e.g., Demo)**

## Cause

The most common reason for this error message is that an incorrect &DocuSignEndpoint parameter has been applied. This issue can also be caused by DocuSign credential setup issues as incorrect credentials can cause problems sending an agreement in background mode.

## What to Do

1. Ensure the &DocuSignEndpoint parameter is configured correctly.
2. Verify that the DSProSFUsername and DSProSFPassword on your DocuSign Account Configuration object and your running user's User record are valid.

## A problem occurred in the rendering process. Please verify your template is valid (Release 7)

### Error Message

This message indicates that a problem prevented Conga Composer from rendering from the template.

### Cause

This message indicates that a problem prevented Conga Composer from rendering from the template. There are a number of issues that can produce this problem.

### What to Do

- Start simple - try clearing the browser cache.
- The template type might be unsupported. For example Conga Composer v7 does not support templates created in Office 2013.
- The user does not have permissions for the object being used in this solution. This could be the cause if a solution works for the Admin but not an end user, or it works in Sandbox but not Production.
- The user does not have field-level security for the field(s) being used in the solution
- Merge fields in a Word Text Box. Text boxes are not supported for merging
- The template may be corrupted template. Try recreating the template.
- This can occur if QVars are not populating fields correctly.

## Found end of mail merge region 'ReportData1 0' without matching start (Release 7)

### Error Message

Missing Table Start merge field.



## Cause

This error occurs when a template table has a TableEnd merge field but is missing the corresponding Table Start.

## What To Do

Review your template to make sure you have all of your Table Start and Table End merge fields.

# Input string was not in the correct format

## Cause

This error occurs when one or more of the merge fields in your template are formatted to display the data as either number/currency or date/time formats while the corresponding data in Salesforce that is being merged to that field is a text value. The template attempts to format a string of text as a number/currency or date/time format, which will not work, so the error occurs.

## What to Do

To correct the problem, locate the offending fields and update them to properly reflect their formatting as dictated in Salesforce which can be done by either a) deleting the merge field and recreating it from scratch, or b) in Word templates, remove the merge field's appearance parameter and right-click to either Toggle Field Codes back to normal or Update Field.

- ✓ In Word templates, toggle field codes open on all merge fields by selecting the entire document (Ctrl + A) and right-clicking on any merge field and selecting Toggle Field Codes; these two steps can also be accomplished via keyboard shortcut Alt + F9 (on a PC) or Alt + Function + F9 (on a Mac). Once all field codes are toggled open, you can more easily look through the merge fields to locate those that have appearance parameters that are attempting to format them in a way that does not match the format of the field in Salesforce.

For example, the merge field { MERGEFIELD Name \# Currency } in a Word template would need to be updated because likely a name field in Salesforce is formatted as text while the

template is attempting to format it for currency using the \# Currency appearance parameter.

# Insufficient Access: Current IP Address is not allowed or invalid\_grant: IP restricted

## Issue

- Insufficient Access or 404 error on Composer Setup tab after clicking the Authorize button.
- Insufficient Access: Current IP Address is not allowed or invalid\_grant: IP restricted

## Workaround

1. If you limit your Salesforce Orgs and Profiles to specific IP Ranges, you must whitelist the Composer IP Addresses to each Profile that needs to use Conga Composer. For more information on how to whitelist Composer IP Addresses, see [Add Conga Composer IP Addresses](#).
2. If you still get an error after whitelisting the Composer IP Addresses, you must check the Login History of the respective user and add the failed IP address as shown below:
  - a. On the Salesforce homepage, click the gear icon (⚙️) in the right corner and select **Setup**.
  - b. Enter **Users** in the **Quick Find** box, then select **Users**.
  - c. Click the respective **User Name** link.
  - d. Go to the **Login History** section and check for **Failure Status** and the **Source IP**.

**Check for failure status and the Source IP**

Login Time	Source IP	Login Type	Status	Application
12/2/2021, 8:46:35 AM MST	34.210.106.249	Remote Access 2.0	Success	Conga Platform
12/2/2021, 8:46:32 AM MST	52.38.231.188	Remote Access 2.0	Success	Conga Platform
12/2/2021, 8:33:06 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:33:02 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:32:45 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:57 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:46 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:26:29 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:23:40 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning
12/2/2021, 8:23:07 AM MST	13.110.74.8	Remote Access 2.0	Success	Conga Composer Lightning

[Download login history for last six months, including logins from outside the website, such as API logins \(Excel .csv file\) »](#)

- e. Add the **Source IP** address into the respective **Profiles** and **Salesforce Org**.

# Insufficient Access to Custom Settings Error

## Issue

Non-admin users that launch Conga Composer in a Salesforce orgs without My Domain and that have accepted the *Require Customize Application permission for direct read access to custom settings* Salesforce Critical Update will incur the error message below. Error message: *Have your system administrator access Composer in this org to enable the custom settings to be read and migrated to a location which can be accessed going forward.*

The error stems from incompatibility with the aforementioned Salesforce Critical Update and the running Composer user's permission to access the *Composer Settings* custom settings. This error only occurs if there has not been a Composer merge since accepting the *Require Customize Application permission for direct read access to custom settings* Salesforce Critical Update.

## Workaround

To resolve this error, contact the Salesforce's org system administrator and have them launch Conga Composer. Launching Conga Composer as a system administrator enables access to the Composer Settings custom setting for all non-admin Composer users.

# Invalid Grant: authentication failure

## Error Message

This error message occurs because the user record attempting to launch Composer is frozen.

## Cause

This Invalid Grant: authentication failure error messages appear in the Composer UI when a user launches their Composer button. The error occurs if a Salesforce user is frozen and attempts to launch Composer.

## What to Do

Unfreeze the Salesforce user to resolve the error message. For more information on freezing or unfreezing User Accounts, see [Freeze or Unfreeze User Accounts](#).

## Invalid master object ID(must be a Salesforce id) (Release 7)

### Error Message

Invalid master object ID.

### Cause

There are two potential causes for this error:

- You have not entered a valid Salesforce ID for the &id parameter in your Composer button URL. This can occur if you hard-code an incorrect ID or an invalidly constructed ID (perhaps using a QVar query)
- You have created a custom field on the Master Object that is also named ID. Even though the Web API name will have a suffix identifying it as a custom field ( c ), this can still cause the error listed above as Conga Composer is "confused" by the presence of two ID fields.

## WHAT TO DO

Rename your custom ID field (e.g. "Custom ID"), updating both the field label and API field name, and re-run your Composer solution. You may need to restart your internet browser in order to clear the cache and have Composer recognize the new name of your custom ID field.

## Invalid PPTX TableGroupStart Field

### Error Message

PPTX templates that use TableGroupStart but which has no data.

### Cause

This error occurs in PPTX templates that use TableGroupStart which has no data.

### What to Do

Use a SlideHide for the same dataset at the bottom of the page.

## Invalid Query Filter Operator

### Cause

This message can occur if you enter a pv1 without a pv0.

### What to Do

- You must specify pv0 as well as pv1.

## Invalid Session ID found in SessionHeader (Release 7)

### Cause

This error typically occurs when there is a problem related to the button URL.

Issues that can cause this error include:

- The object name of the Master Object is too long.

- There are spaces in the button URL. Check the button URL, especially at the end of each line where spaces tend to "hide" to ensure that there are no spaces within the URL.
- The button URL is too long. Too many characters in a button URL can prevent Composer from running properly.
- The "Lock sessions to the IP address from which they originated" setting is enabled in your Salesforce instance. With this setting enabled, Conga Composer cannot operate. To check this whether this setting is enabled, navigate to Salesforce Setup > Administration Setup > Session Settings.
- [Salesforce.com](https://www.salesforce.com) has timed out. Try refreshing your webpage, logging out of Salesforce and logging back in, and restarting your internet browser.

## What To Do

- Clear your cache and try again.
- Upgrade to Conga Composer Release 8
- If you are using Conga Composer Release 7, ensure you are not running it in Lightning or Salesforce1, or using an OAuth button

## Merge and Box Errors

### Known Errors and Workarounds

Error messages might appear when you select Merge and Box in Composer.

#### Error

There was an issue processing the Composer request. The remote server returned an error: (409) Conflict.

#### Cause

A common reason for these errors is that there are more than 1,000 folders included at any given level of the Composer document output file path. For more information on the output file path, see the OFP parameter.

Composer uses Box's Get Folder Items API call to identify folders where the document is saved. The Get Folder Items API call has a limit of 1,000 folders. Composer is unable to query for folders beyond the limit. For more information on Box, see: <https://developer.box.com/reference#...-folders-items>

## Solution

Add folder levels to your OFP parameter that do not exceed 1,000 folders at any given level in your Box account.

## No DocuSign Recipients Defined (Release 7)

### Error Message

No DocuSign recipients defined.

### Cause

This can occur for a few reasons:

- The master object is an Opportunity and the Opportunity Contact roles do not align (aren't synced) with the DocuSign roles. To resolve edit either the Salesforce or DocuSign roles so they match. Then any contact roles from the Opportunity will be assigned as Recipients on the DocuSign Envelope
- The &DocuSignR1Id is not set to a valid ID (for example setting it to a Contact's text value rather than the ID)
- The user doesn't have access to the field used for the DocuSign Recipients parameter.

### What To Do

- Ensure a Recipient is defined
- Ensure the Recipient ID is a valid record
- Check Field Level Security on the Recipient Object (Lead, Contact, or User)
- Check Field Level Security on the Master Object
- Ensure DocuSign for Salesforce is functioning in the org

## Object reference is not set to an instance of an object (Release 7)

### Cause

This error message will appear in the View Data Excel workbook on one or more worksheets where a query has been used to retrieve data. It indicates that something is wrong with that particular query dataset -it's incorrectly referenced, inaccessible, or incorrectly written.

### What to Do

- Confirm that the proper query ID has been referenced in the Conga Composer button or link URL. Copy the query ID from the button URL and paste it into your Salesforce URL to ensure that it's connected to a Conga Query record.
- If you see a Salesforce error message indicating the record is not available, it's either been deleted, was copied from a Sandbox and needs to be re-created in production, or the user doesn't have permissions to access the Query record.
- Confirm that the user profile has access to the Conga Queries custom object.
  - Navigate there by clicking Salesforce Setup > Administer |Manage Users > Profiles > Object Settings > Conga Queries. "Read" is the minimum level of access required while "Edit," "Create," "Delete," "View All," and "Modify All" are optional. Confirm this level of access for all profiles running Conga Composer solutions.

## Object reference is not set to an instance of an object (Release 8)

### Cause

Related to a bad merge field or mismatching TableStart and TableEnd merge fields in a [detail region](#).



## What to Do

A good start is to remove paragraphs (or entire pages for larger templates) and repeat the merge until you find the issue. Then you can correct the merge field or detail region causing the problem.

For example, you may receive the error message if the TableStart and TableEnd merge fields are not in the same row of the detail region. The correct usage of TableStart and TableEnd merge fields is shown below.

SUMMARY			
PRODUCT	SALES PRICE	QTY	TOTAL PRICE
<b>{{TABLESTART:OPPLINEITEMS}}</b> {{PRODUCT_NAME}}	{{OPPORTUNITY_LINEITEM_SALESPRI CE \# CURRENCY}}	{{OPPORTUNITY_LINEITEM_QUANTITY \# \#,##0}}	{{OPPORTUNITY_LINEITEM_TOTALPRICE \# Currency}} <b>{{TABLEEND:OPPLINEITEMS}}</b>
	TOTAL:	{{=SUM(Above)}}	{{=SUM(Above)}}

## Cause

You may also receive the error message if data with double quotes is being merged into an IF statement in a Microsoft Word template.

## What to Do

1. Remove the double quotes from the field.
2. Create a new formula field or DVO parameter and reference the value of the existing field in a Salesforce SUBSTITUTE formula to remove all double quotes from the field value. Then reference the new formula field or [DVO](#) parameter within the IF statement.

## Conga Conductor and Conga Workflow Release 8

### Cause

For Conga Conductor and Conga Workflow, this error message is most commonly caused by field level security issues with the formula field that contains the Conga Conductor or Conga Workflow formula.

### What to Do

1. Navigate to Salesforce Setup > Security Controls > Field Accessibility.
2. Select the object on which the field exists, select View Fields, and then select the field.
3. Lastly, select the Profile of the user running the Conductor solution (or the Profile of the User to Send as user for Workflow) and ensure that Profile has visibility to the formula field.

Field-Level Security:			
Profile	Field	Visible	Read-Only
Contract Manager	Invoice	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## One or More Templates Failed to Merge or Operation Was Cancelled

### Issue

There are several possible causes for the following error message: One or more templates failed to merge or operation was cancelled:

1. Related to a bad merge field or mismatching TableStart and TableEnd merge fields in a detail region.
2. Data containing double quotes are being merged into a Microsoft Word IF Statement.

## Workaround

A good start is to remove paragraphs (or entire pages for larger templates) and repeat the merge until you find the issue. Then you can correct the detail region or IF Statement causing the problem.

### If the issue is with a detail region:

You can receive this error message if the TableStart and TableEnd merge fields are not in the same row of the detail region. The correct usage of TableStart and TableEnd merge fields looks like:

SUMMARY			
PRODUCT	SALES PRICE	QTY	TOTAL PRICE
<code>{{TABLESTART:OPPLINEITEMS}}</code> PRODUCT_NAME}}	<code>{{OPPORTUNITY_LI NEITEM_SALESPRI CE \# CURRENCY}}</code>	<code>{{OPPORTUNITY _LINEITEM_QUA NTITY \# #,##0}}</code>	<code>{{OPPORTUNITY_LINEI TEM_TOTALPRICE \# Currency}}</code> <code>{{TABLEEND :OPPLINEITEMS}}</code>
	TOTAL:	<code>{{=SUM(Above)}}</code>	<code>{{=SUM(Above)}}</code>

### If the issue is with an IF Statement:

1. Remove the double quotes from the field.
2. Create a new formula field or DV0 parameter and reference the value of the existing field in a Salesforce SUBSTITUTE formula to remove all double quotes from the field value. Then reference the new formula field or DV0 parameter within the IF statement.

**Ouch... something's definitely gone wrong. Please try again. (Error #1890: ORNF) (Release 7)**

## Cause

The error is triggered when you have an improperly formatted template: Examples of problems that may trigger this include:

- Carriage Return in a merge field
- An errant character in a merge field's label

- Contact Conga support for assistance in identifying the specific problem in your template.

## What To Do

- Clear your cache
- Upgrade to Conga Composer Release 8

## Our Apologies . . . Your Conga URL contained illegal characters (Release 8 and 7)

### Cause

You receive this error message when illegal characters are used in the Composer button URL.

### What To Do

The following illegal characters must be removed from the URL:

- • Spaces
- • & (anywhere other than at the beginning of a parameter)
- • % (percent)
- • | (pipe)
- • () (parenthesis)
- • " (double quotation marks)
- • ' (Apostrophe)

## Page Cannot Be Found

### Cause

This error occurs when there are spaces in your URL.

## What To Do

Examine your button or link URL and look spaces when you have cut and pasted the button code from another source.

A Conga button, when launched, is essentially a URL being passed between servers. Spaces in the URL will cause the "string" to break. Use Notepad or a similar text editor when copying and pasting buttons in your Salesforce org to avoid formatting errors.

## Read pdf error: Trailer is not found (Release 7)

### Cause

This typically occurs when the associated PDF file is corrupted.

### What to Do

Try regenerating the template and re-uploading it into Salesforce.

## Reduce complexity of SOQL statement to avoid errors

### Question or Problem

You see the following error when viewing the standard Opportunity Line Items (Opplineitem object) data set in both the Template Builder and View Data workbook from a button on the opportunity object: QUERY TOO COMPLICATED - Query is either selecting too many fields or the filter conditions are too complicated.

### Causes or Possible Causes

Long, complex SOQL statements, such as statements that contain many formula fields, can sometimes result in a QUERY\_TOO\_COMPLICATED error. The error occurs because the statement is expanded internally when processed by Salesforce, even though the original

SOQL statement is under the 20,000 character limit. To avoid this, reduce the complexity of your SOQL statement.

When Composer performs queries on Opportunity Line Items, 3 objects are retrieved:

1. OpportunityLineItem
2. OpportunityLineItem.PricebookEntry
3. OpportunityLineItem.PricebookEntry.Product2

The resulting query exceeds the 20,000 bytes and the query fails due to the excessive number of custom fields. For more information, see the Salesforce documentation for SOQL limits: [https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_select.htm](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_select.htm)

## Report exceeds maximum supported rows from PointMerge (Release 7)

### Cause

Conga enforces a data set limit of 1500 rows. This message occurs when you are attempting to use a data set that exceeds that maximum.

### What to Do

- The work around for the data-set row limit is to break that data set up into multiple queries/reports, each limited to 1500 rows.
- A common approach is to sort the query alphabetically and create multiple clones, each for a different portion of the alphabet; i.e. A-G, H-N, O-Z.

**Report size cannot be determined, either because the report is excessively large, the report is unavailable or the report type is matrix. Please verify the Report Id.**

## Cause

This error message appears in the Conga Template Builder and View Data workbook.

- The report is returning too many results (rows of data). Conga Composer Release 7 supports a maximum of 1,500 results per report while Release 8 supports a maximum of 2,500 results per report.
- Matrix reports, joined reports, and historical trending reports are not supported. these report types. You may also receive this error message if you have exceeded the limit for rows or lines of data.
- The report is not returning any data because the report filters are not configured properly.
- The report ID referenced in the Conga Composer button or link URL is incorrect.
- The report is inaccessible.
- The report no longer exists.
- There is a space in the Conga Composer button or link URL preventing the report ID from being read.

## What to Do

- Limit your report to align with the 1,500 or 2,500 result maximum. Please see the Know Your Conga Limits article for more information on limitations.
- Use a summary or tabular type of report.
- Conga Composer's default behavior is to pass the Salesforce record ID of the master object (usually the object from which you launch Conga Composer) to the first filter in each report. Ensure your report is configured to accept the master object ID or update the Composer button or link URL to override this behavior. Please see [How to Use Salesforce Reports with Conga Composer](#) for further details.
- Ensure the report ID referenced in the button or link URL corresponds to the report you want Conga Composer to run. Copy the report ID from the button or link URL and paste it into the browser's Salesforce URL to confirm it is indeed the proper report.

- Do you have access to the report? It may be saved in a report folder to which you don't have access. Copy the report ID from the button or link URL and paste it into the browser's Salesforce URL to confirm
- The report may have been deleted. Copy the report ID from the button or link URL and paste it into the browser's Salesforce URL to confirm it is available. If it's not available, check the Salesforce Recycle Bin and undelete the report.
- Edit the button or link URL and delete any spaces (there should be zero spaces in the URL). Spaces often "hide" at the ends of lines within the URL.

## REQUEST LIMIT EXCEEDED:TotalRequests Limit exceeded

### Cause

This error is received when your organization has reached its 24-hour API Limit.

### What to Do

To check the limit, go to the Company Information page in Salesforce (Setup > Company Profile > Company Information). The API Requests line will indicate how much of your API allocation has been consumed.

## Server was unable to process request. Value cannot be null. Parameter name: password

### Error Message

This error message occurs when a Salesforce User does not have their DocuSign credentials entered on their User record.

### Cause

This error message occurs when a Salesforce User does not have their DocuSign credentials entered on their User record.



## What to Do

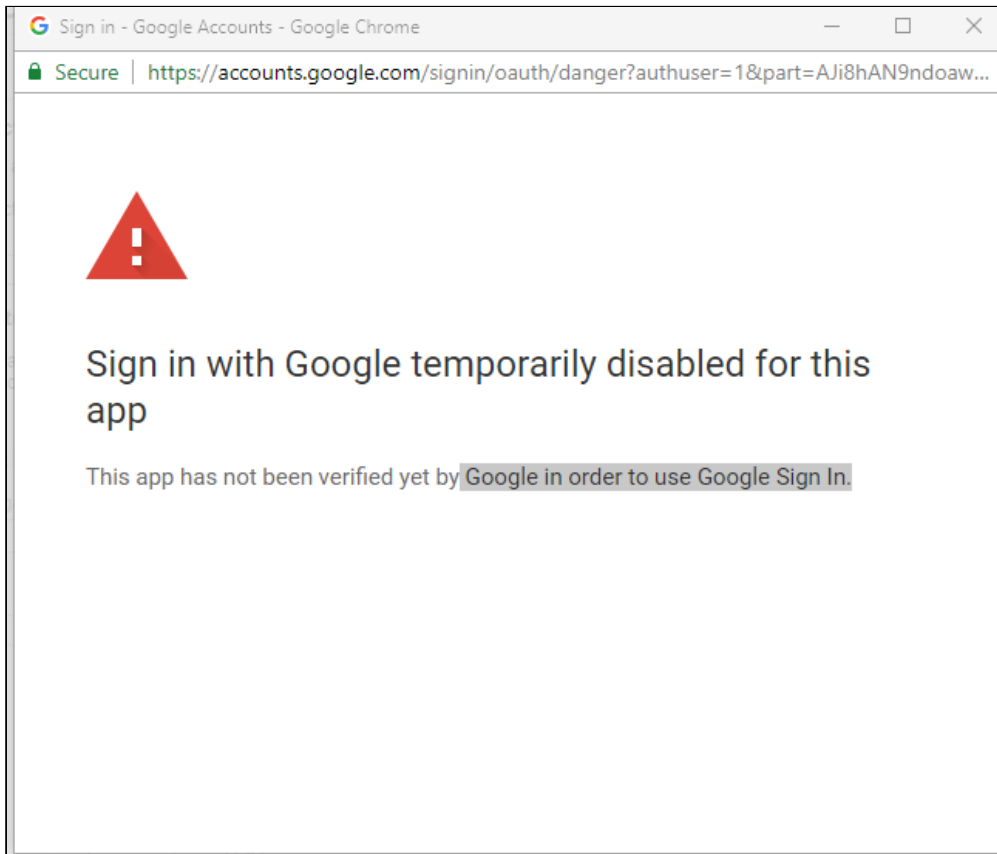
When running a Conga routine involving DocuSign and Composer in background mode (DS7/Workflow/Conductor), a complete DocuSign password must be current on the User Profile in Salesforce.

If a version of DocuSign is upgraded, the Password for the Salesforce User Profile must be re-entered.

**Sign in with Google temporarily disabled for this app. This app has not been verified yet by Google in order to use Google Sign In**

## Question or Issue

When you click to create a Google token button on the Setup pages for Composer, Conductor, or Workflow, you receive the following error during the login process:



Sign in with Google temporarily disabled for this app This app has not been verified yet by Google in order to use Google Sign In.

## Causes or Possible Causes

Recent updates made by Google resulted in this error.

## Related Known Errors and Workarounds

Our development team is investigating a fix. As a workaround, you can launch the process manually.

# Something's gone wrong! An internal server error has occurred

## Error Message

Salesforce API limit exceeded.

## Cause

This error occurs when you reach your Salesforce API limit.

## What to Do

To check the limit, go to the Company Information page. The API Requests line will indicate how much of your API allocation has been consumed. The API request field is not updated in real time. It has a lag time of up to 15 minutes.

# Something's not quite right about your data – it can't be displayed in Excel. Column 'id' does not allow nulls. (Release 7)

## Error Message

Something's not quite right about your data – it can't be displayed in Excel. Column 'id' does not allow nulls.

## Cause

The Conga View Data Workbook generates an Excel workbook based on the dataset supplied to Conga in your solution. This error occurs when the dataset supplied to Conga is empty. There are a variety of reasons that can cause this such as:

- A poorly formed query can filter out all data resulting in an empty dataset
- A Salesforce report that filters out all data

- Permissions that do not allow the user running the solution to see the required data

## What to Do

- Review your data for null values (ensure every record in your report has an ID)
- Test on a different record
- Run your Salesforce report outside of Conga Composer (by entering your record ID in the first filter on the report)
- Unless you are using Mail Merge or Conductor, remove the ID column from your report

## Something's wrong with your template (bad mojo). The merge template has serious problems

### Error Message

This error indicates that something is wrong with the template(s).

### Cause

This error indicates that something is wrong with the template(s). Unfortunately the error doesn't correspond to one specific issue within the template so it can sometimes be challenging to diagnose and find the source of the problem; the below steps will help narrow down the cause.

1. If you are using multiple templates in your merge process, merge to each of them separately to find out specifically which template is causing the "bad mojo" error. If the error message results from more than one of the templates, you'll need to take the following steps with each of the templates individually. Also, if you're merging to PDF output, you may want to try the below steps with merging to both PDF output and "same as template" output to see if the error message results are different one way versus the other.
2. Save a new copy of the problem template (the one resulting in the error message) to your local computer (e.g. My Documents, Desktop) and run it through the merge process as a local template (no need to upload it to Conga Templates since we'll be testing this way over and over again) to ensure you still get the error after saving a new copy of the template (if you do receive the error message again, proceed to step 3).
3. If you do not receive the error again, try simply re-uploading this new version of the

file to your template record in Conga Templates and run your merge process using that version of the template.

3. Open the template in its native program (e.g. Microsoft Word). Click to the end of the template, delete the last section (e.g. table, paragraph, etc.), do a Save As to save your changes, and run it through the merge process as a local template. If you receive the error message again, that indicates that the source of the problem was not located in the last section of the template (the section you just deleted).
4. Proceed once again to the end of the template, delete the next section, save your changes, and run the template through the merge process as a local template. If you receive the error message again, you'll need to repeat this step (#4) until you run the template through the merge process and do not receive the error, at which point you can proceed to step 5.
5. When you can run your template through the merge process and not receive the error, you should be able to isolate the section of the template that is causing the error (e.g. the table using TableGroup on page 2). You can confirm it is indeed that section by adding in the other sections you previously deleted but leaving out the section you believe is causing the error and run the template through the merge process again.
6. You've now isolated which section of your template is causing the "bad mojo" error. Try deleting this section and recreating it from scratch to fix the error. For instance, if it's a table that's causing the problem, delete the entire table and its contents, insert a new table from scratch, and re-insert the merge fields and any static text within the table. DO NOT copy and paste from the previous version(s) of the template as this likely could be what was causing the problem in the first place. Save your changes and run this new version of the template through the merge process as a local template. If it merges and does not result in the error message, upload it to Conga Templates and run the merge again to confirm that it still works without issue. If the merge results in the error message again, proceed to step 8.
7. You've isolated the section of the template that is causing the error and have recreated that section but it still hasn't fixed the template. Therefore, try recreating that section in a brand new, fresh template with no other data in it, save it to your local computer and run it through the merge process as a local template. You can also try adding back in just one component of that section at a time. For instance, you might start with a blank table and if you don't receive the error then you can add the static text and run it again. If that works without error, then add back in a couple of merge fields, and so forth.

## What To Do

After following the above instructions, you should be able to isolate and fix the source of the error. If not, then you can try the below steps as well:

- If using a Word .docx file, save a new copy of the template as .doc and run that version through the merge process instead.
- Are you merging the output file to PDF? If so, try merging to "same as template" (e.g. Word template to Word output) to see if the error still results. The contrary method can also be helpful - if you're merging to "same as template," try PDF output instead to see if the error results are different.
- Are you using the &AWD parameter in your Conga Composer button URL? If so, remove the parameter or change it to &AWD=0 (either accomplishes the same behavior) and run the merge process again.

If you're still experiencing difficulties after following all of the above troubleshooting advice, please contact our support team.

## Stack Empty (Release 7)

### Error Message

Detail region tags (TableStart/End) placed within nested IF statements generate an error message when merging a Word Template.

### Cause

When you receive a 'Stack Empty' error message when merging a Word Template, it usually means that there are detail region tags (TableStart/End) placed within nested IF statements.

IF fields in Word display content conditionally. If some of these detail region tags are shown incorrectly on a conditional basis, it can result in mismatched TableStart/End detail regions due to the conditional formatting. This can result in the "Stack Empty" error.

### What to Do

The best solution to this issue is to adjust your detail region tags so that they're not embedded within IF statements, or if they must be, ensure that each one will have a corresponding Start/End position.

If the above solution does not fix the issue, you may want to check your footers and headers for possible errors, as this can also cause the "Stack empty" error.

To troubleshoot headers and footers, save a new copy of your template and then remove one header or footer at a time, doing a test merge operation using the local template each time. Once you can run the merge operation without an error, you should be able to narrow down which header or footer section is causing the issue.

## String not recognized as a valid DateTime

### Error Message

Conga cannot format a non-date field as a date.

### Cause

This error results when there is a merge field on the Word template that is formatted as a date (e.g. @ "MMMM d, yyyy") while that field in Salesforce is not a date field (e.g. instead it's text). The error results because Conga cannot format a non-date field as a date.

### What to Do

To fix the error, start by toggling all field codes on Word template (Alt + F9). Look for non-date fields that are formatted as dates. Beware, as some fields that sound like date fields (e.g. Deliverable Due Date) are in fact not. When you find the merge field in question, delete the date formatting (e.g. @ "MMMM d, yyyy"), right-click the field and select "Update Field." Alternately, you may delete the entire merge field and simply re-enter it without any date formatting.

The error may appear when merging from some records and not from others. This is because the field in question is not populated on the records where the merge works.

## Template Id(s) do not exist

### Error Message

Having a dash "-" or a forward slash "/" in the template alias produces the error: "Template Id/s do not exist." Dashes and slashes are not supported in the Template Alias.

## Cause

Having a dash "-" or a forward slash "/" or an underscore "\_" in the template alias produces the error.

## What to Do

Dashes and slashes are not supported in the Template Alias. If you need to use those characters you'll need to use the &OFN parameter instead.

This produces an error: `&TemplateId=[Pipeline+Report+_+2015-01-27]a0Cj0000001I1Xj`

This Works: `&TemplateId=a0Cj0000001I1Xj&OFN=Pipeline+Report+_+2015-01-27`

It is important to note that all special characters can be problematic with using the &TemplateId parameter.

## Template Id is Invalid

### Error Message

Formatting template ID parameter.

### Cause

An extra space after a template ID parameter will cause an error when adding multiple template IDs. Example:

### What to do

Remove the spaces and it will fix the error. If you are using formula fields/if statements, make sure to add the comma before the ID in the clause.

**i** There is a limit of 10 template IDs allowed. You will receive an error message you exceed this amount that says: You have exceeded the maximum allowed templates per merge. Allowed: 10



# There hasn't been a file or email template ID defined, background mode cannot continue when trying to use (Release 8)

## Error Message

Background mode parameter (&DS7) is used but a Template ID or Conga Email Template ID has not been specified.

## Cause

This Conga Composer error message occurs when the background mode parameter (&DS7) is used but a Template ID or Conga Email Template ID has not been specified. Because background mode skips the user interface in which you can choose which template or email template to use in the merge process, the template or email template must be pre- defined in the button or link URL via its Salesforce ID.

In other words, anytime you use the &DS7 parameter, you must also have the &TemplateID and/or &CongaEmailTemplateID (a.k.a. &CETID) parameter(s) included.

## What to Do

The below button or link URL example includes a pre-defined template using the &TemplateID parameter as well as a pre-defined Conga Email Template via the &CongaEmailTemplateID parameter. The solution also runs in background mode (&DS7) meaning the typical Conga Composer user interface is skipped; in this case because the background mode value is 2 (&DS7=2), the merged output file is attached to the Conga Email Template for sending via email.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={! Opportunity.Id}
&Queryid=[ProductsTotal]a05i000000LfoYM
&TemplateID=a04i000000GGSj8
&CongaEmailTemplateID=a03i000 +for={!Opportunity.Account}_ {!TODAY() }
```

## There is no row at position 0

### Error Message

Restricted Field-level security settings on user profiles.

### Cause

This Conga error message will occur if the end users have Restricted Field Level Security set for their profile. If the end users are trying to generate a document that requires data from certain fields or reports that they do not have access to they will see this error message. Restricted Field-level security settings let administrators restrict user access to view and edit specific fields. In some cases, there may be a change to security settings that can contribute to why some end users may have access verses others.

### What to Do

Contact your Salesforce Administrator to find out if there has been a change in the security settings. For more information see [Salesforce Field-Level security](#).

## There was an issue getting the Conga Authorization Response

### Cause

This error message is usually caused when there is an issue reaching the Conga servers.

### What to Do

1. Clear your browser's cache and cookies.
2. Ensure that there are no firewall or security settings that may be blocking your connection.
3. Check the status page - <http://status.getconga.com/>

4. You may need to "white-list" these two Conga URLs by adding them to your browser's Allowed Sites:
  - [https://\\*.congamerge.com](https://*.congamerge.com)
  - [https://\\*.appextremes.com](https://*.appextremes.com)
5. Contact Conga Support

## There was an issue processing the Composer request

### Issue

The following error messages may be sporadically generated when clicking Merge and Box in Conga Composer. There was an issue processing the Composer request. The remote server returned an error: (409) Conflict. A common reason for these errors is having more than 1,000 folders included at any given level of the Composer document's output file path (see [OFP](#) for more information). This is because the API call Box makes has a limit of 1,000 folders, so Composer is unable to query for folders beyond the 1,000th folder.

### What to Do

The solution is to ensure there will never be over 1,000 folders at any given level in your Box account.

## The ',' character, hexadecimal value 0x2C, cannot be included in a name

### Cause

This error message results when two parameters that use sequential numbering are included using the same number in a Conga Composer solution that's configured to run in Background Mode (using the &DS7 parameter).

An example of a parameter that uses sequential numbering is the Master Field to Set (&MFTS) parameter because you would begin with &MFTS0 and &MFTSValue0 but can include a second parameter that should be incrementally numbered, &MFTS1 and &MFTSValue1.

The error occurs when the sequentially ordered parameter is mistakenly not sequentially ordered but rather the same number is used more than once (e.g. &MFTS0, &MFTSValue0 and &MFTS0, &MFTSValue0 - both parameter sets using 0 instead of the first using 0 and the second using 1). The error message will only occur when such a Conga Composer solution is configured to run in Background Mode. A sample Composer button URL is included below with the offending Master Field to Set and Background Mode parameters.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&Queryid=[ProductsTotal]a05i000000LfoYM
&TemplateID=a04i000000GGSj8
&CongaEmailTemplateID=a03i000000Gj1Pu
&DefaultPDF=1
&OFN=Invoice+for={!Opportunity.Account}_ {!TODAY()}
&MFTS0=Description
&MFTSValue0=Awesome+description&MFTS0=Amount
&MFTSValue0=10000&DS7=1
```

### What to Do

To correct this error, edit the sequentially numbered parameters so that they are incrementally numbered (e.g. 0, 1, 2). For example, to fix the above button URL we'd change the second occurrence of &MFTS0 and &MFTSValue0 Master Field to Set parameters to &MFTS1 and &MFTSValue1 instead. You may keep the Background Mode (&DS7) parameter.

```
https://composer.congamerge.com
?sessionId={!API.Session_ID}
&serverUrl={!API.Partner_Server_URL_290}
&id={!Opportunity.Id}
&Queryid=[ProductsTotal]a05i000000LfoYM
&TemplateID=a04i000000GGSj8
&CongaEmailTemplateID=a03i000000Gj1Pu
&DefaultPDF=1
&OFN=Invoice+for={!Opportunity.Account}_ {!TODAY()}
&MFTS0=Description
&MFTSValue0=Awesome+description&MFTS1=Amount
```

```
&MFTSValue1=10000
```

```
&DS7=1
```

## The report is invalid. Are you using My Domains or matrix style reports?

### Issue

If permissions to the reports are granted, sharing settings to the folder are set, and it is not a matrix style report and users still see this error, it is likely that they have High assurance session required enabled in their Salesforce org under Access Policies.

### Workaround

Disable the High assurance session required policy in Salesforce Setup > Customize > Reports and Dashboards > Access Policies. Use Conga Queries, which are unaffected by this Access Policy.

## Parameter p3lkid value error

### Cause

The p3\_lkid parameter value contains a character that is not allowed or exceeds the maximum character value length.

The error is a Salesforce error that appears on the Salesforce page rather than the Conga Composer dialog box. The error typically occurs when a problem occurs with the Conga Composer email parameters.

### What to Do

1. Remove any unacceptable characters from the p3\_lkid parameter value or reduce the value length of the value and resubmit.
2. Review the Composer button URL to resolve any of the following issues:

- `&EmailRelatedTold={!Lead.Id}` The Salesforce email message generated from a Lead object does not include the "Related To" field, which results in the above "p3\_lkid" error.
- `&EmailRelatedTold` is set to a name field rather than an ID field.
- You specified the same object ID in both `&EmailTold` and `&EmailRelatedTold`.  
For example:
  - `&EmailTold={!Contact.Id}`
  - `&EmailRelatedTold={!Contact.Id}`
- `&EmailTold` should be used to specify the primary recipient ("To" field on the email message) while `&EmailRelatedTold` is used to populate merge fields in the email template body from a related object or for activity logging purposes. An example of the correct usage of both parameters would be:
  - `&EmailTold={!Contact.Id}`
  - `&EmailRelatedTold={!Account.Id}`
- You used the `&EmailTemplateId` and `&EmailSubject` parameters. Remove `&EmailSubject` since it is redundant. The email template specified using `&EmailTemplateId` already includes an email subject.
- If you use a Custom Object and have not enabled Track Activities, the Conga Button does not allow you to send email. Allow Activities must be enabled for Custom Objects in order to send Emails from Conga.

## Unable to parse the query id entries, please check the URL syntax (Release 8)

### Error Message

This error results from an invalid report or query alias.

### Cause

This error results from an invalid report or query alias. Report and query aliases must be enclosed in square brackets, cannot contain spaces or special characters and are limited to 20 total characters. The dataset names of "Master," "Org," "User," "ReportData" and "QueryData" cannot be used as aliases because they are reserved for the default datasets.

## What to Do

To resolve the error, please check your report and query aliases, change them as needed, save your changes, and test your Conga Composer solution once more. As a reminder, report and query aliases (below, highlighted in yellow) are included before the report or query ID.

For example:

```
https://composer.congamerge.com  
?sessionId={!API.Session_ID}  
&serverUrl={!API.Partner_Server_URL_290}  
&id={!Account.Id}  
&ReportId=[Cases]000i00000063Kae  
&QueryId=[TechBuyerContact]a05i000000L3p0W
```

## Unexpected end of children nodes reached (Release 7)

### Cause

This error is usually caused by a carriage return (line break) located within a merge field.

## What to Do

1. Open your Word template and toggle the field codes by pressing Alt+F9.
2. Click ¶ to view paragraph formatting.
3. Examine the display to locate carriage returns in any merge fields.
4. Delete any carriage return within merge fields.

## URLENCODE function for illegal characters in button URL

The following characters cannot be used as part of the parameter values in a Composer button URL. Using any of these characters will cause your solution to fail

Character Name	Character
Colon	:
Hash	#
Open Parenthesis	(
Close Parenthesis	)
Double Quotes	"
Percentage	%
Backslash	\

## URL Encoding Illegal Characters

The above illegal characters can be encoded using URL Encoding. This will allow you to use them in your parameter values in your Composer URL. Below are their encoded values:

Character Name	Character	Encoded Value
Colon	:	%3A
Hash	#	%23
Open Parenthesis	(	%28
Close Parenthesis	)	%29
Double Quotes	"	%22
Percentage	%	%25
Backslash	\	%5C



## Using the SUBSTITUTE Function in Salesforce

If you are referencing a Salesforce field as a parameter value in your Composer URL, and this field may contain ampersands, you can create a Salesforce formula field and use the SUBSTITUTE Function to replace Ampersands within the field value. You would then use the formula field in the parameter instead. See this article for details: [Replace Ampersands in Reference Field Data](#)

## User Management Access Error

### Cause

- This message occurs when User Management has been enabled but Conga has not been enabled for the user's login (the user is not listed as an authorized user in User Management).
- The same error message can also result if the &DC (disable Conga) parameter is included in the Conga custom button/link URL but the &DCL (disable Conga label) parameter has not been included. &DCL provides a custom message when Conga has been conditionally disabled using &DC. If &DCL is not included, the default message is:

"Access not authorized. Please contact your Salesforce administrator and ask to have your name added to Conga Setup > User Management."

### What to Do

To correct the error, go to Salesforce Setup > Installed Packages and then click Manage Licenses next to the Conga Composer package and ensure the user has been assigned a license.

✓ To learn about the User Management tab, see [User Management Settings](#)

If the &DC (disable Conga) parameter is included in the Conga custom button/link URL but the &DCL (disable Conga label) parameter has not been included, check the Conga button URL to evaluate the &DCL conditions and change as necessary.

For more Composer troubleshooting help, see [Composer Troubleshooting Tips](#).

# Composer Output Contains Blank Fields

## Cause

Conga Composer will retrieve data from [Salesforce.com](https://www.salesforce.com) for all of the fields specified in your template according to the profile permissions of the running user. If a user is merging a document and is not receiving the expected data in the output file, troubleshoot the issue with the list below

## What to Do

Conga Composer will show a blank value in the merged output file if the field in Salesforce is blank,

The merge field name in the document template needs to exactly match the field name as it's listed in the Template Builder or View Data workbook. If you are unsure whether the field name from the Template Builder matches what's in your template, delete the merge field in the template and create a new one from scratch.

If your Word template uses traditional merge fields (e.g. «ACCOUNT\_NAME») instead of text-based merge fields (e.g. {{ACCOUNT\_NAME}}), ensure that you are comparing the field name, not the field label. The field name is what needs to match the field name that's listed in the Template Builder. Please see [What is the difference between a Field Label and a Field Name](#) for further information.

If a user does not have access to a field in Salesforce, then that user will not be able to merge that field using Conga Composer. For Salesforce Enterprise Edition and above, this means ensuring the user's Profile has access to the field via field-level security. For Professional Edition, this means ensuring that the field is included on the page layout.

The user will need access to the report and report folder for any fields that are being retrieved through a Salesforce report for a Conga Composer solution. When fields are retrieved via SOQL queries, the user will need access to the Conga Queries custom object and all of its fields.

For Enterprise Edition and above, this means users must have at least "read" access to the Conga Queries custom object as well as visibility to all fields via field-level security. For Professional Edition, all of the Conga Queries fields need to be included on the Conga Queries page layout. The specific Conga Query record will also need to be accessible by

the user. Please see Profile Permissions needed to use Conga Composer for further information.

If you are still experiencing difficulties after troubleshooting the issue using the above checklist, please Contact Support.

## Your Conga Session has Expired

### Cause

Users receive the following error message when launching Composer: "Invalid Session or your session has expired. (Lock IPs may be enabled.)" This error occurs when the "Lock IP Address" setting is enabled or when launching a non-OAuth button in Lightning.

### What to Do

- Disable the "Lock IP Address" setting for hosted AppExchange applications.
  - a. Navigate to Salesforce Setup > Administration Setup > Security Controls > Session Settings.
  - b. Ensure that the "Lock sessions to the IP address from which they originated" field is NOT checked.
  - c. Click **Save**.
- Convert your non-OAuth buttons to OAuth buttons for Lightning.
- You must RESTART all instances of the browser for the setting to take effect.

## Salesforce session expired or the Salesforce API Servers are offline - INVALID SESSION ID or SERVER URL (Release 7 and 8)

### Cause

This error message usually results because of one (or more) of the following reasons:

- "Lock sessions to the IP address from which they originated" has been enabled in your Salesforce instance. With this setting enabled, Conga Composer cannot operate. To check this whether this setting is enabled, navigate to Salesforce Setup > Administration Setup > Session Settings.

- The users running Conga Composer do not have the "API Enabled" permission on their profile. This setting is on the Profile > Administrative Permissions > API Enabled.
- The maximum API request limit for your Salesforce instance has been exceeded for the day. If your Salesforce instance has exceeded its maximum API requests, then Conga Composer cannot run (as we require the API to operate). This is a rolling 24 hour period so with each hour that passes, you will slowly accrue "fresh" API requests.
- The server for your Salesforce instance (e.g. na3) is experiencing downtime/ degradation issues.
- You will also receive this error if you are launching a non-OAuth button in lightning.

## What to Do

- Check your API usage under Salesforce Setup > Administration Setup > Company Profile > Company Information > API Requests, Last 24 Hours.
- Ensure "Lock sessions to the IP address from which they originated" is NOT enabled
- Ensure the user running the solution has "API Enabled" on their Profile
- Ensure you have not reached your maximum API request limit
- Check if the server for your Salesforce instance is experiencing downtime or degradation issues. See <http://trust.salesforce.com/trust/status>.

## "Conga can't communicate with Salesforce" error

### Cause

Conga is unable to communicate with Salesforce.

### What To Do


- Ensure "Lock Sessions to IP Address from which they originated" is disabled (Setup > Administration Setup > Security Controls > Session Settings).
- Verify that the Session Timeout value is at least 30 minutes.
- Ensure the Conga Composer URL contains an "&Id=" parameter and that it is set to a valid Id.
- Don't try to "Preview" a Conga Composer button. You must click it from a Salesforce record.

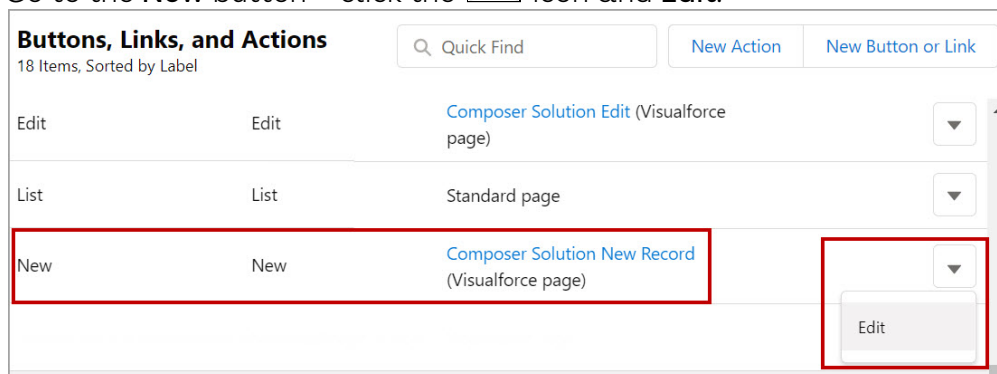
# Collection size exceeds maximum size of 1,000

## Cause

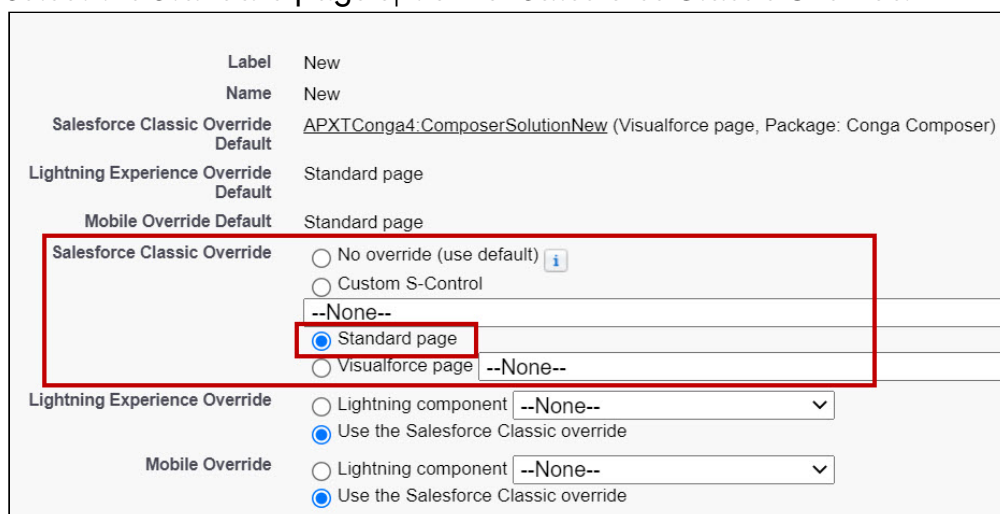
This error usually occurs when you have more than 1000 objects in your Salesforce org. The Visualforce method cannot query more than 1,000 objects for the Master Object dropdown.

## What to Do

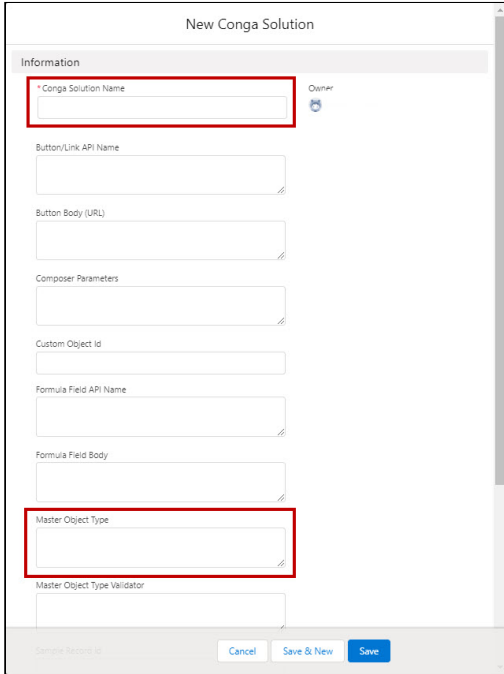
1. Make the following changes to the **Salesforce Classic Override** setting:
  - a. On the Salesforce homepage, click the gear icon (⚙️) in the right corner and select **Setup**.
  - b. Go to the **Object Manager** tab.
  - c. Search and select **Conga Solution** object.
  - d. Click **Button, Links, and Actions** option.
  - e. Go to the **New** button > click the  icon and **Edit**.



- f. Select the **Standard page** option for Salesforce Classic Override.



- g. Click **Save**.
- 2. When you create a new Solution, you must manually enter the name of the solution in the **Conga Solution Name** field and the API name of the object (where you want the button to appear) in the **Master Object Type** field.



✔ If you're creating a lot of solution records, change the *Page Layout* to rearrange the sequence of the **Master Object Type** field for better usability.

## Conga Trigger

Conga Trigger is an optional add-on to <sup>®</sup> that runs individual solutions through Salesforce Workflow Rules. Conga Trigger enables you to create and distribute documents without user interaction.

⚠ Trigger solutions will not work when you have *High Assurance* enabled in your Salesforce Org and uses multiple users with the **User to Send As** feature.

- [Conga Trigger Installation and Setup](#)
- [Conga Trigger Product Guide](#)
- [The About Conga Trigger Tab](#)
- [The Conga Trigger Setup Tab](#)
- [The Conga Trigger Formula Builder Tab](#)
- [The Conga Trigger Dashboard Tab](#)

- [Upgrade Conga Trigger to Release 8](#)
- [Automate the Conga Trigger Solutions](#)
- [Troubleshooting Tips for Conga Trigger Solutions](#)

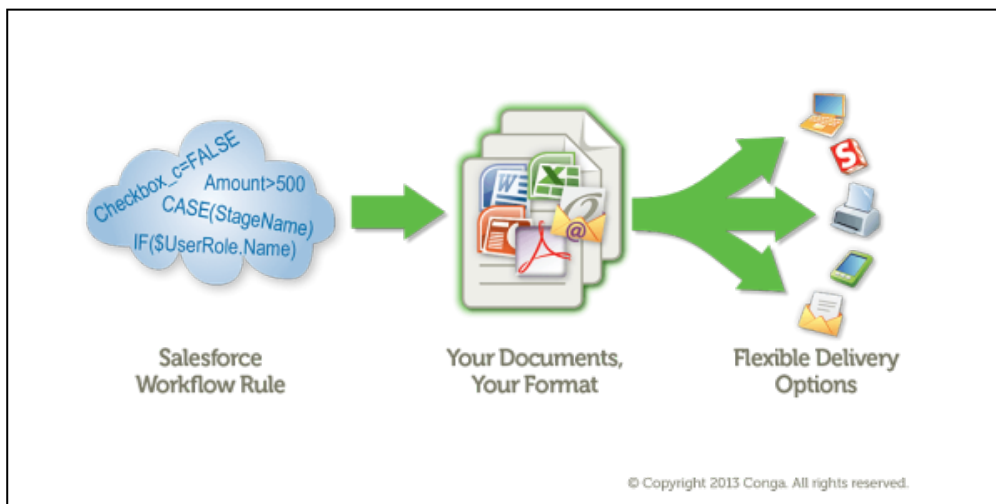
## Conga Trigger Installation and Setup

Easily install and setup Conga Trigger using the following documentation

- [About Conga Trigger](#)
- [Install Conga Trigger](#)
- [What's in the Conga Trigger Package?](#)
- [Before You Begin: Conga Trigger](#)
- [Convert a Button to a Formula Field Using the Conga Formula Builder](#)
- [Create a Workflow Rule for Conga Trigger](#)
- [Test Using the Salesforce Workflow Rule](#)

## About Conga Trigger

Conga Trigger can initiate a Conga Composer solution, thus enabling the creation and delivery of a document without direct user interaction. Using Conga Trigger in this manner is intended as a substitute for manually clicking a Composer button, essentially automating Conga Composer.



Conga Trigger gives you the ability to distribute merged output as:

- Email attachments
- eSignature documents (Conga Sign, DocuSign and Adobe eSign)
- Salesforce record attachments, Content, Documents or Chatter

Conga Trigger is NOT intended for processing many simultaneous requests due to Salesforce API constraints.

You may not use Conga Trigger to trigger a Conga Batch® batch or any other sort of batch operation.


## How Does it Work?


Conga Trigger is initiated by Salesforce workflow rules. Conga Trigger uses an Outbound Message workflow action to launch the specified process.

The steps involved in configuring Conga Trigger are listed below.

1. Convert a working solution to a Salesforce formula field using the Conga Formula Builder.
2. Create a Salesforce workflow rule that will initiate Conga Trigger.
3. Add an outbound message action to the Salesforce workflow rule.
4. Activate the Salesforce workflow rule.
5. Test to ensure everything is working properly.

## Install Conga Trigger

 Installing Conga Trigger is OPTIONAL. You do not need to install Conga Trigger to get started with Conga Composer. Conga Trigger is an add-on and must be installed separately. Conga Trigger must also be activated by the Conga support team.

 Tabs in the Conga Trigger managed package count against your Salesforce organization's tab limit. Be mindful of this when installing.

### To install Conga Trigger:

1. Click the appropriate link to install Conga Trigger in your Salesforce environment:
  - [Install Conga Trigger in a Sandbox org](#)
  - [Install Conga Trigger in Production org](#)
2. Enter the Salesforce credentials for the instance where Conga Trigger will be installed and log in to Salesforce.
3. Select a security level for the Conga Trigger application.
  - a. Conga recommends selecting Install for All Users.



4. Click **Install**.
  - a. An Installation Complete! alert appears, confirming the Conga Batch package has been installed successfully.
5. Click **Done**.

## What's in the Conga Trigger Package?

After installing the Conga Trigger package, you will see the following tabs in your Salesforce Org.

Tab	Description
About Conga Trigger	A custom tab describing Conga Trigger capabilities and the steps to begin the configuration process.
Conga Trigger Setup	A custom tab to integrate Conga Trigger with Google Drive.
Conga Trigger Formula Builder	A custom tab used to access the Formula Builder, which converts a button or link URL to a Salesforce formula field.
Conga Trigger Dashboard	A custom tab to monitor Conga Trigger events and view errors for any records that failed to process.

In addition, there are various documents and folders installed. For details, please see [Salesforce Setup > Build > Installed Packages > Conga Trigger > View Components](#).

## Before You Begin: Conga Trigger

You should be generally familiar with the Salesforce workflow. You should know how modifying a field on a record can trigger Salesforce workflow to perform an action, such as sending an Outbound Message.

Conga Trigger is an add-on to an active subscription and is purchased separately. The service must be activated by Conga before use. Please contact support to subscribe or start a free trial.

### Requirements


1. A working solution is required to use Conga Trigger. Ensure this solution has been fully tested and that all aspects of it (the template, activity logging, integrations, etc.) are working as desired.
2. Salesforce workflow is required in order to use Conga Trigger. Salesforce workflow is not included with all Salesforce editions. Please check the details of your Salesforce edition to ensure you have access to the Salesforce workflow.

## Convert a Button to a Formula Field Using the Conga Formula Builder

Unlike a Conga Composer button or link which contains a URL with various parameters to define your merge operation, Conga Trigger uses a formula field on your Master Object to host those instructions. Use the Conga Formula Builder to convert a working Conga Composer button or link into a Salesforce formula field used by Conga Trigger.

If you prefer to build your formula field manually, please see [How to Manually Convert a Conga Composer Button into a Conga Trigger Formula Field](#).

### To convert a button to a formula field using the Conga Formula Builder:

1. Click the **App Launcher** icon () and search for *Conga Trigger Formula Builder*.
2. Select **Conga Trigger Formula Builder**.
3. Choose the **Master Object** from the drop-down menu. The Master Object is the object where your button or link is located.
4. Select your desired **Output Mode**. This adds a corresponding *QMode parameter* to the formula field.

Additional parameters may be required based on the Output Mode selected. You may need to modify your button/link to include those required parameters. For more information, see [Specify How Conga Trigger Delivers Merged Output Files](#).


#### **Note**


If your button includes a DS7 parameter (to run in background mode), the Output Mode is pre-selected to match the background mode behavior.

#### **Important notes about Google Output Modes**

- The Output Modes “Attach the output to Google Drive” and “Convert the output to a Google Doc” require a Google Refresh Token, which is configured from the Conga Batch Setup tab.
- The merged output files from Google Drive or Google Doc Output Modes are stored in the Google account of the user who creates the Google Refresh Token.
- buttons or links configured to save a copy of the merged output file to Google Drive will include the parameters &SC0=1 and &SC1=Google. However, if these parameters are maintained in the Conga Batch formula field and “Attach the output to Google Drive” is the Output Mode, Conga Batch will store two copies of each merged output file in Google Drive (one from the &SC0=1 and &SC1=Google parameters, and one from the “Attach the output to Google Drive” Output Mode). Therefore, if “Attach the output to Google Drive” is your desired Output Mode, you should remove the &SC0=1 and &SC1=Google parameters from your Conga Batch formula field.

5. In the Button/Link area, select the **Button Name**. The button or link automatically converts to a formula field and display in the left pane for Formula/URL.

 Only traditional URLs for buttons or links display in the Button/Link Name drop-down. URLFOR functions, JavaScript buttons, etc. are not supported.

6. (Optional) Clear the **Show Converted Formula** box to toggle back to the URL view of the button/link.
7. In the Formula Field area, rename your button by typing a name in the Name box. We recommend “Conga Trigger [Output Mode] [Name of solution],” for example, “Conga Trigger Email Proposal.”
8. Click **Create Field**. The formula field is created on the master object.
9. Click the Salesforce **Setup** icon () in the upper right corner of the screen.
10. Click **Setup**.
11. Navigate to the **Object Manager** tab and locate the master object where the formula field is created.
12. In the left pane, click **Fields & Relationships** and select the formula field hyperlink.
13. Click the **Set Field-Level Security** button.
14. Check the **Visible** checkbox for the necessary profile and click **Save**.

⚠ If you do not change your field-level security, your solution will not work (you will receive the error message "Object Reference not set to an instance of an object."). You must edit field-level security even if the person creating the formula field has a Standard Salesforce Administrator profile.

## Create a Workflow Rule for Conga Trigger

### To create a workflow rule for Conga Trigger

1. Navigate to Setup in Salesforce.
2. Click Create.
3. Click Workflow & Approvals.
4. Click Workflow Rules (in Salesforce Lightning: Setup
  - a. (in Salesforce Lightning: Setup → Process Automation → Workflow Rules)
5. Click New Rule.
6. Complete the steps using the information below:
  - a. The object selected must be the same as the Master Object from your Composer solution (the same object on which you created the Conga Trigger formula field).
  - b. We recommend setting the Rule Name to "Conga Trigger [Name of Composer solution] [Output Mode]" (ex: "Conga Trigger Invoice Email").
  - c. We recommend setting the Evaluation Criteria to created, and any time it's edited to subsequently meet the criteria. However, you may select another option depending on your requirements. See the warning below for more details.
  - d. You may create a new field (for example, a checkbox named Conga Trigger) or leverage an existing field (such as Opportunity Stage) in the criteria of your

## Conga Trigger rule.

**Step 2: Configure Workflow Rule**

Enter the name, description, and criteria to trigger your workflow rule. In the next step, associate workflow actions with this workflow rule.

**Edit Rule**

a Object: Opportunity

b Rule Name: Conga Workflow - Invoice

Description: Sends invoice via email using Conga Workflow when Stage = Closed Won.

**Evaluation Criteria**

Evaluate the rule when a record is:

- created
- created, and every time it's edited
- created, and any time it's edited to subsequently meet criteria

How do I choose?

**Rule Criteria**

Run this rule if the following criteria are met:

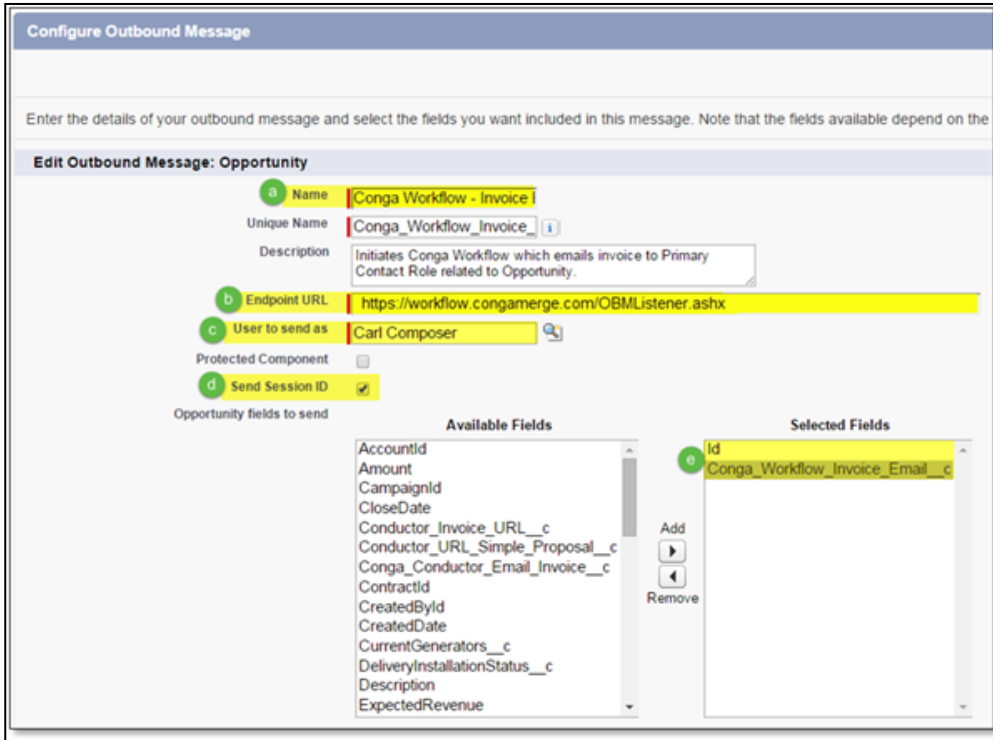
Field	Operator	Value	
Opportunity Stage	equals	Closed Won	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND

In the example above, the Salesforce workflow rule runs when the Opportunity Stage field is changed to “Closed Won,” which then initiates Conga Trigger. Conga Trigger creates an invoice document, attaches it to an email, and sends that email to the Opportunity’s Primary Contact.

- ⚠ The ability to merge and send a document without user oversight or intervention potentially poses a great risk. Remember that every time the Salesforce workflow rule fires, the Conga Trigger process runs as well, meaning the merged output file will be generated, which can have far-reaching implications, such as an infinite loop.
- For example, assume your Salesforce workflow rule is set to evaluate every time a record is edited and you’re using a checkbox field as the rule criteria (e.g. “Conga Trigger equals true”). If you don’t include an additional field update workflow action to disable the checkbox field after Conga Trigger fires, then a document will be generated every time the record is edited.
- Thoroughly test your solution and confirm it’s working correctly before deploying it fully.

## To Add an Outbound Message workflow action to the Salesforce workflow rule

1. Add an Outbound Message workflow action to the Salesforce Workflow Rule.



- a. We recommend setting the Name of the outbound message as “Conga Trigger [Name of Composer solution] [Output Mode]” (ex: “Conga Trigger Invoice Email). Optionally, include a description as well.
- b. Add the following Endpoint URL: <https://workflow.congamerge.com/OBMListener.ashx>

**i** The Endpoint URL is different for integrating Conga Trigger with the Conga Invoice Generation solution. The Conga Trigger Endpoint URL for Conga Invoice Generation is <https://workflow.congamerge.com/sfb/obmlistener.ashx>

- c. User to Send As is the user under whose authority Conga Trigger will run. The User to Send As should have a custom or standard System Administrator profile and a Composer license. Ensure that this user is active and has access to all of the components used in the Conga Trigger solution (ex: reports, queries, objects, fields, email privileges, eSignature privileges) as well as access to the Conga Trigger formula field.

**Important**

The *User to send as* overwrites the *EmailFromId* parameter when you add the *EmailFromId* parameter to your solution.

- d. Send Session ID must be enabled.
  - e. Selected Fields must include the Master Object ID (id) and the Conga Trigger formula field.
2. Activate the Salesforce workflow rule by clicking Activate. The Active checkbox enables and the Activate button changes to Deactivate.

[Test the Conga Trigger solution.](#)

## Test Using the Salesforce Workflow Rule

To test the solution by triggering the Salesforce workflow rule.

1. Confirm that the Conga Trigger formula field is accessible (field-level security is set to Visible) by System Administrators only. This ensures that the activated Salesforce workflow rule that initiates the Conga Trigger process won't work for anyone except System Administrators while you are still in the testing phase.
2. Select a sample Master Object record to use for the test. Sample records are used so that the testing phase won't send documents or communications to actual recipients (customers, partners, employees, etc). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request.
3. Trigger the Conga Trigger routine in the manner you defined on the Salesforce Workflow Rule.
4. Analyze the results.

This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot.

You should confirm that the following aspects of the solution are working properly:

- Reports and queries are retrieving data to be used in the template.
- Data is merging to the template, creating the merged output file.
- Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
- Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.

## Conga Trigger Product Guide

### Learn Conga Trigger Basics

- [About Conga Trigger](#)
- [Before You Begin: Conga Trigger](#)
- [Conga Trigger Caveats and Warnings](#)
- [What's In the Conga Trigger Package?](#)

### Navigate the Conga Trigger App

- [The About Conga Trigger Tab](#)
- [The Conga Trigger Setup Tab](#)
- [The Conga Trigger Formula Builder Tab](#)
- [The Conga Trigger Dashboard Tab](#)

### Install and Configure Conga Trigger

- [Install Conga Trigger](#)
- [Upgrade Conga Trigger to Release 8](#)
- [Integrate Conga Trigger with Google Drive](#)

### Use Conga Trigger

- [Automate Conga Trigger Solution](#)
- [Convert a Conga Composer Button to a Conga Trigger Formula Field Using the Conga Formula Builder](#)
- [Specify How Conga Trigger Delivers Merged Output Files](#)
- [Create a Workflow Rule for Conga Trigger](#)
- [Update an Existing Conga Trigger Formula Field Using the Formula Builder](#)



- [Replace Ampersands \(&\) in Reference Field Data](#)

## Troubleshoot Conga Trigger Solutions

- [Troubleshooting Tips for Conga Trigger Solutions](#)
- [Troubleshooting using Hyperlink Formula Fields](#)
- [Test Using a Hyperlink Formula Field](#)
- [Test Using the Salesforce Workflow Rule](#)
- [Test Conga Trigger Solutions Upgraded from Release 7](#)
- [Test an Upgraded Conga Trigger Solution](#)
- [Viewing Errors](#)

## The About Conga Trigger Tab

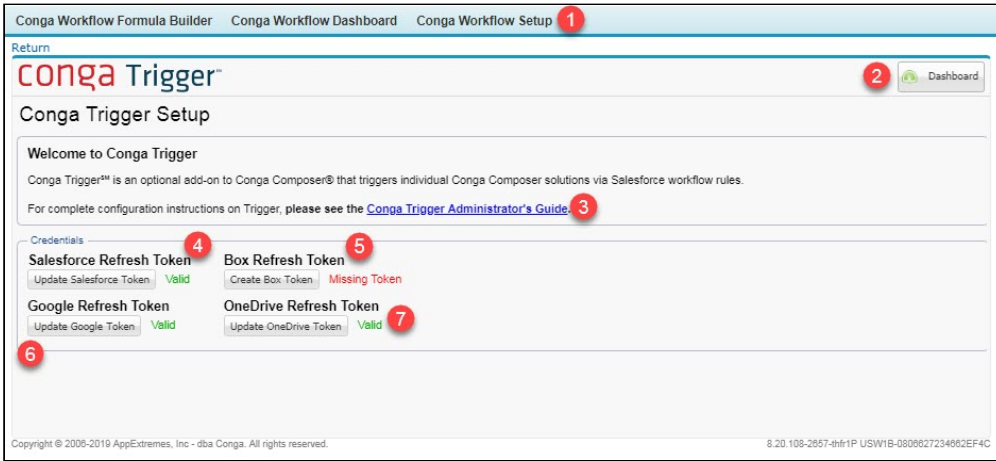
The About Conga Trigger tab describes Conga Trigger capabilities and provides information on how to begin the configuration process.



1	Click the About Conga Trigger tab.
2	Create a Workflow Formula is the starting point of Conga Trigger, explaining how to convert an existing Conga solution to a Conga Trigger formula field.
3	Create a Workflow Rule describes how to create the Salesforce workflow rule that will initiate the Conga Trigger solution.
4	Monitor describes the Conga Trigger Dashboard which monitors Conga Trigger events and lists errors for any records that failed to process.

# The Conga Trigger Setup Tab

The Conga Trigger Setup tab provides tools to administer Conga Trigger.



1	Click the Conga Trigger Setup tab.
2	The Dashboard button provides another way to access the Conga Trigger Dashboard.
3	Conga Trigger Configuration Guide links to the PDF of the Conga Trigger Configuration Guide which explains how to build Conga Trigger solutions.
4	Create a Salesforce Refresh Token to use Salesforce Reports as a data-gathering source for Conga Trigger.
5	Create a Box Refresh Token to integrate Conga Trigger with Box.
6	Create a Google Refresh Token to integrate Conga Trigger with Google Drive.
7	Create a OneDrive Refresh Token to integrate Conga Trigger with OneDrive.

## The Conga Trigger Formula Builder Tab

The Conga Trigger Formula Builder provides an entirely new way to create Conga Trigger formula fields by converting an existing button or link into a Conga Trigger formula field. In previous versions of Conga Trigger, this conversion had to be done manually.

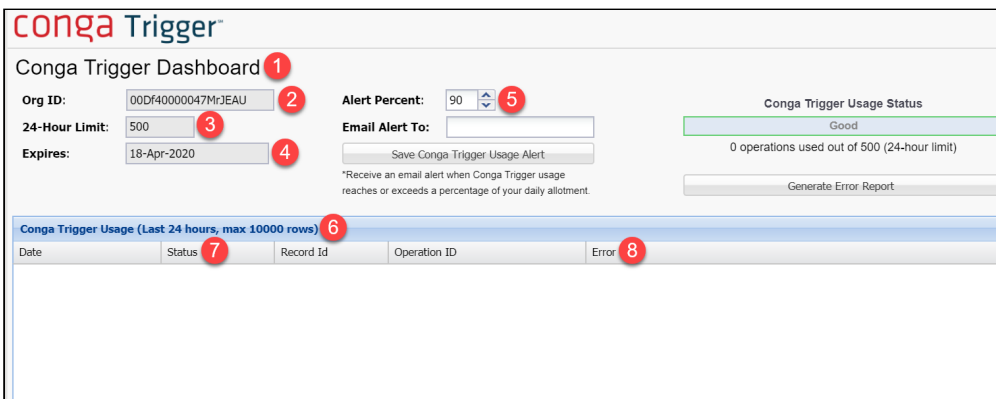
1	The Conga Trigger Formula Builder tab.
2	The status bar walks through the steps of the Formula Builder. Click any step of the status bar to learn what is required.
3	In the Master Object box, select the object on which the button or link is displayed.
4	In the Output Mode box, select how the merged output file will be delivered.
5	In the Button/Link area, in the Name box, select the button or link to convert to a formula field.
6	After you select a button or link, the URL displays briefly, followed by its formula field equivalent in the Formula / URL area. Clear the Show Converted Formula box to see the button or link URL instead of the formula.

7	In the Formula Field area, in the Name box, type a name for the formula field. We recommend using a name that includes Conga Trigger and the button name, such as Conga_Workflow_Email_Proposal.
8	Click Create Field to create the formula field on the master object.
9	(Optional) Click the Edit button to open the button or link in Salesforce Setup and make changes to the button.
10	After the field is created, click the Field-Level Security button to modify the field-level security for the Workflow formula field (not pictured).

The Conga Trigger Formula Builder may also be used to edit and update existing formula fields. For more information, please see Update an Existing Conga Trigger Formula Field Using the Formula Builder.

## The Conga Trigger Dashboard Tab


The Conga Trigger Dashboard tab monitors Conga Trigger events and lists errors for any records that failed to process.



1	The Conga Trigger Dashboard tab.
2	Org ID displays the Salesforce Organization ID.
3	24-Hour Limit is the number of Conga Trigger service events allowed per 24 hour period according to your Conga Trigger subscription and configuration.

4	Expires displays the Conga Trigger subscription expiration date.
5	Alert Percent is the percentage of usage.  <div style="border: 1px solid red; padding: 5px; background-color: #f9f9f9;"> <p><b>Known Issue</b></p> <p>The system does not send an email to the participant listed in the Email Alert To field when the percentage is reached.</p> </div>
6	The Usage Detail (Last 24 Hours) area displays details of the Conga Trigger events that ran successfully in the last 24 hours.
7	Conga Trigger Usage Status displays the number of Conga Trigger service events that have been consumed as well as the total number allowed per 24 hour period.
8	The Error List area displays details of records that have failed to run in Conga Trigger in the last 24 hours. <ul style="list-style-type: none"> <li>• The Record Id column displays the Master Object ID of the record within a Conga Batch that failed.</li> <li>• The Error column displays the error message, providing details of why that particular record failed.</li> </ul>

## Upgrade Conga Trigger to Release 8

 This section is only applicable to those customers who are currently using an older version of Conga Trigger (Release 7) and wish to upgrade to Conga Trigger Release 8.

Conga Trigger is a composite application, which means that some parts of the system are installed into the Salesforce instance in a managed package; however, most of the system runs on Conga's servers.

The portions of the application that run on Conga's servers are updated automatically. However, the portions of the system that are installed into Salesforce are upgraded by installing Conga Trigger Release 8 and following the prompts to update the installed components.

The following instructions provide specific instructions and considerations for those who are upgrading.

## Upgrade the Conga Trigger App to Release 8

Previous versions of Conga Trigger did not require installation because they were not packaged as an application; instead, Conga Trigger Release 7 solutions were built by manually creating the Conga Trigger formula field and configuring the Salesforce workflow rule and action. Consequently, upgrading from Conga Trigger Release 7 to Release 8 simply involves installing Conga Trigger Release 8 into the same Salesforce instance in which Conga Trigger Release 7 is configured.

Please see [installing Conga Trigger](#) for details on how to install the Conga Trigger Release 8 app.

The Conga Trigger solutions built in the previous version of Conga Trigger (Release 7) are maintained in the Salesforce instance when Conga Trigger is upgraded to Release 8. These solutions will continue to run on the Conga Trigger Release 7 servers unless they are updated to run instead on the Release 8 servers. Please see the following topics to learn how to upgrade Conga Trigger solutions.

## Upgrade Conga Trigger Solutions to Run on Release 8

Upgrading Conga Trigger solutions involves updating the endpoint URL specified on the Salesforce workflow action outbound message. Redirect the endpoint URL from the Conga Trigger Release 7 servers to the Conga Trigger Release 8 servers.

Repeat this procedure for all Conga Trigger solutions needing to be upgraded to Release 8.

## To upgrade Conga Trigger solutions to run on Release 8 servers

1. Locate the Salesforce workflow rule that initiates the Conga Trigger solution and click Rule Name.

- Navigate to Salesforce Setup > Build > Create > Workflow & Approvals > Workflow Rules.

**All Workflow Rules**

Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:

- Criteria that cause the workflow rule to run.
- Immediate actions that execute when a record matches the criteria. For example, salesforce.com can automatically send an email that notifies a contact when a sales lead has been assigned.
- Time-dependent actions that queue when a record matches the criteria, and execute according to time triggers. For example, salesforce.com can automatically schedule a task for a user when an opportunity is still open ten days before the close date.

View: **All Workflow Rules** [Create New View](#)

Action	Rule Name +	Description
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	<b>Email Price Quote - Conga</b>	Merges quote template and emails it to primary contact role related to Oppty

- In the Immediate Workflow Actions area, click the Description field entry for the Outbound Message workflow action associated to the workflow rule.

**Workflow Rule**  
**Email Price Quote - Conga**

[Back to List: Workflow Rules](#)

**Workflow Rule Detail** [Edit](#) [Clone](#) [Deactivate](#)

Rule Name	Email Price Quote - Conga
Active	<input checked="" type="checkbox"/>
Description	Merges quote template and emails it to primary contact role related to Oppty
Rule Criteria	Opportunity: Stage EQUALS Proposal/Price Quote
Created By	Shelley Hoolihan, 1/12/2014 10:12 PM

**Workflow Actions** [Edit](#)

**Immediate Workflow Actions**

Type	Description
Outbound Message	<b>Email Price Quote - Conga</b>

- Click Edit on the outbound message.
- Change the Endpoint URL to point to the Release 8 servers.
  - Replace <https://workflow.appextremes.com/apps/Conga/PMWorkflow.aspx> with <https://workflow.congamerge.com/OBMListener.ashx>

**Edit Outbound Message**  
**Email Price Quote - Conga**

Enter the details of your outbound message and select the fields you want included in this message. Note that the fields available depend on the type of record.

[Save](#) [Save & New](#) [Cancel](#)

**Edit Outbound Message: Opportunity**

Name	Email Price Quote - Conga
Unique Name	Email_Price_Quote_Conga
Description	
Endpoint URL	https://workflow.appextremes.com/apps/Conga/PMWorkflow.aspx
User to send as	Shelley Hoolihan
Protected Component	<input type="checkbox"/>
Send Session ID	<input checked="" type="checkbox"/>

**Edit Outbound Message: Opportunity**

Name	Email Price Quote - Conga
Unique Name	Email_Price_Quote_Conga
Description	
Endpoint URL	https://workflow.congamerge.com/OBMListener.ashx

- Click Save.

7. Test the Conga Trigger solution on the Release 8 servers before deploying.

## Using Data Gathering Reports in a Conga Trigger Solution

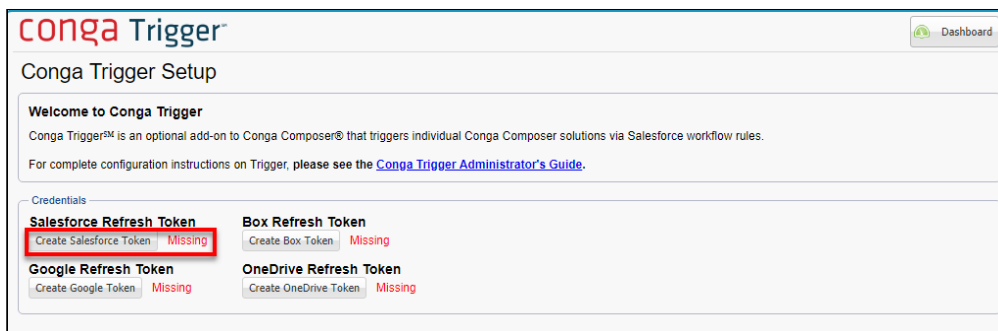
To use Salesforce Reports as a data gathering source in a Conga Trigger Solution, users must create and save a Salesforce Token in [Conga Trigger Setup](#).

⚠ When using reports in a Conga Trigger formula field, the user that created the Conga Trigger token overrides the "User to Send As" in the outbound message.

While the Salesforce Token enables Conga Trigger to utilize Salesforce Reports as a data gathering source, it is highly recommended to replace Salesforce Reports with Conga Queries. Conga Queries function as a more stable data gathering source for all products, including Conga Trigger. For more information on converting reports to queries in a Conga Trigger solution see, [Convert Data Gathering Reports to Queries for Conga Trigger Solutions](#).

### To Create the Salesforce Token

1. Click the Conga Workflow Setup tab to access Conga Trigger Setup.
2. Click Create Salesforce Token.

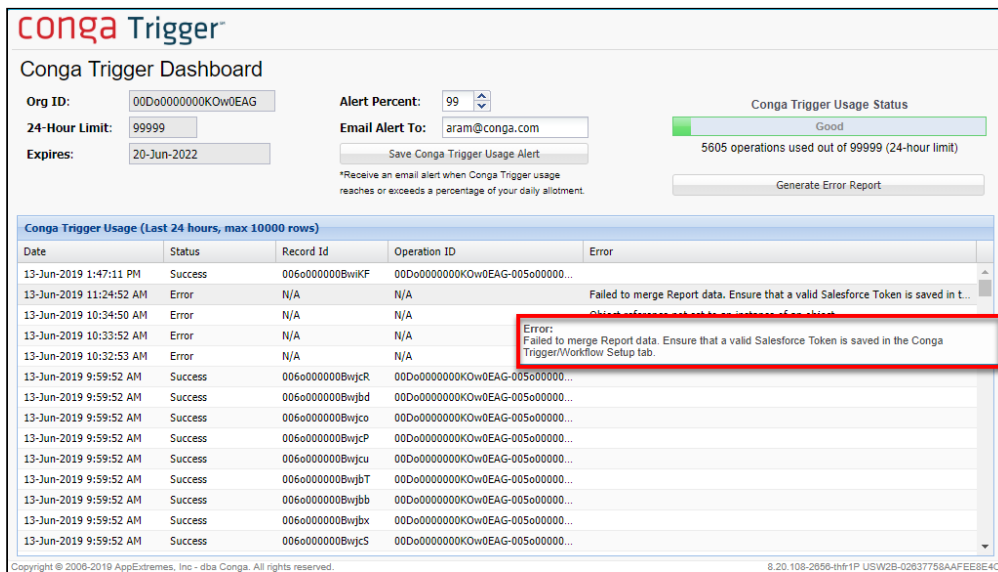


3. Click **Allow**.

If a Conga Trigger solution uses the ReportID parameter and the Salesforce Refresh Token is not present in Conga Trigger Setup, the following error appears in the Conga Trigger Dashboard:

Error: Failed to merge Report data. Ensure that a valid Salesforce Token is saved in the Conga Trigger/Workflow Setup tab.





**Conga Trigger Dashboard**

Org ID: 00Do000000Kow0EAG | Alert Percent: 99 | Conga Trigger Usage Status: Good  
 24-Hour Limit: 99999 | Email Alert To: aram@conga.com | 5605 operations used out of 99999 (24-hour limit)  
 Expires: 20-Jun-2022 | Save Conga Trigger Usage Alert | Generate Error Report

\*Receive an email alert when Conga Trigger usage reaches or exceeds a percentage of your daily allotment.

Date	Status	Record Id	Operation ID	Error
13-Jun-2019 1:47:11 PM	Success	006e0000008BwIKF	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 11:24:52 AM	Error	N/A	N/A	Failed to merge Report data. Ensure that a valid Salesforce Token is saved in t...
13-Jun-2019 10:34:50 AM	Error	N/A	N/A	Object reference not set to an instance of type System...
13-Jun-2019 10:33:52 AM	Error	N/A	N/A	Error: Failed to merge Report data. Ensure that a valid Salesforce Token is saved in the Conga Trigger/Workflow Setup tab.
13-Jun-2019 10:32:53 AM	Error	N/A	N/A	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjcR	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjbD	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjcO	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjcP	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjcU	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjbT	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjbB	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjbX	00Do000000Kow0EAG-005e000000...	
13-Jun-2019 9:59:52 AM	Success	006e0000008BwjcS	00Do000000Kow0EAG-005e000000...	

Copyright © 2008-2019 AppExtremes, Inc. - dba Conga. All rights reserved. | 8.20.108-2056-shfr1P-USWZB-02837758AAFE0E4C

## Test Conga Trigger Solutions Upgraded from Release 7

If you are currently running Conga Trigger solutions on Release 7 and want to upgrade to Conga Trigger Release 8, these instructions enable you to test a solution to ensure that it works as expected.

Before You Begin: [Upgrade to Conga Trigger Release 8](#). The upgrade process will retain all existing Conga Trigger solutions from Release 7 and these solutions can continue to run on Release 7 or can be upgraded to run on Release 8.

See [Test an Upgraded Solution using the Salesforce Workflow Rule](#).

For more information and specific instructions, see [Troubleshooting Conga Trigger Solutions](#).

## Automate the Conga Trigger Solutions

Conga Trigger is initiated by Salesforce workflow rules. Conga Trigger uses an Outbound Message action to launch the specified process.

After you install Conga Trigger, you should configure the app for your organization.

The steps involved in configuring Conga Trigger are listed below.

1. [Convert a working solution to a Salesforce formula field using the Conga Formula Builder](#).
2. [Create a Salesforce workflow rule that will initiate Conga Trigger](#).
3. [Test to ensure everything is working properly](#).

# Edit a Button and Formula Field using the Formula Builder

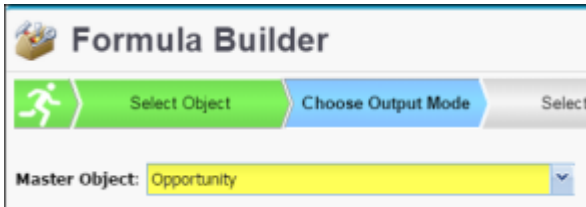
How to convert a button URL to a Salesforce formula using Conga Formula Builder:

## To Edit a Formula Field

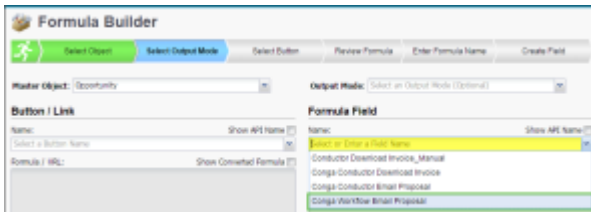
1. Click the Conga Formula Builder tab.



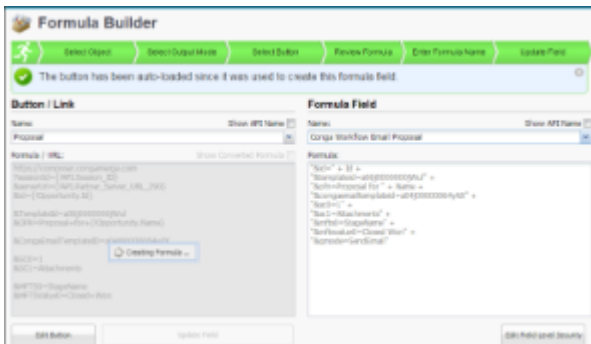
2. In the Master Object box, select the master object specified in your Conga Composer button or link. The Master Object is the object where your Conga formula field is placed.



3. In the Formula Field area, in the Name box, select the formula field to update.



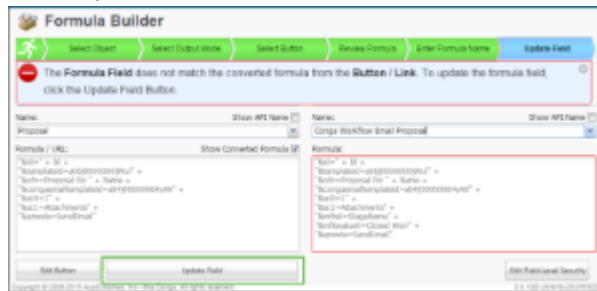
The converted Conga Composer button or link appears in the left pane while the formula field it was converted to is in the right pane.



Disable the Show Converted Formula checkbox to view the button in URL format.

4. Click **Edit Button**. The Conga Composer button opens in edit mode in a new browser tab.

5. Edit the button and click Save.
6. Return to the Formula Builder and in the Button / Link Name drop-down menu, select any other button, and then select the original button you edited above. This is similar to a browser refresh (alerting Formula Builder that changes were made in the button) but allows the Formula Builder to remember your place in the update process.
7. Formula Builder notifies you that the button and formula field don't currently match – click Update Field.



8. Click OK on the dialog box confirming the formula field has been updated.

The button and the formula field now match exactly and the updates are complete.

## Specify How Conga Trigger Delivers Merged Output Files

With Conga Composer, you can specify how the merged output file will be delivered or where it will be stored after launching the button. However, Conga Trigger runs in the background in response to Salesforce workflow rules, so you must specify in advance how Conga Trigger will deliver the output files.

You do this by using a QMode parameter in the Conga Trigger formula field.

The parameter name remains constant (&QMode=) but its value changes depending on the desired delivery method (e.g. "Attachments" for storing a file in Notes & Attachments, "SendEmail" for emailing).

 Only one QMode parameter may be used per Conga Trigger solution.

**Category:** Automation-Background Mode Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QMode

**Parameter Value:** [value]

**Parameter Requirements:** None

**Parameter Description:** Parameters for Conga Batch and Conga Trigger. The QMode parameter is required when working with Conga Batch or Conga Trigger solutions.

If you use the QMode parameter to distribute documents (such as email or eSignature) and you also want to save copies of the document, ensure the SC1 parameter is used in conjunction with QMode.

**Example:**

```
&QMode=Attachments
```

**QMode Parameter Values for Release 8**

QMode Value	Description
&QMode=Attachments	Attaches the merged output files to the Master Object records
&QMode=Download	<p>Downloads the merged output files to the internet browser.</p> <p>This parameter value is available only to Conga Batch. There is no download option available in Conga Trigger.</p> <p>Recommended: &amp;ContentWorkspaceId=[Workspace ID] to specify the Content Workspace (Library) where Conga Batch will store the merged output file if the Conga Batch solution is configured to run on a schedule. Without this parameter, the merged output file from a scheduled Conga Batch solution configured to download will be stored in the running user's personal Salesforce Content Workspace (Library).</p> <p>If either the &amp;DefaultPDF=1 or &amp;FP0=1 parameter is not included in the Conga Batch formula field, the merged output files will be consolidated into a .zip file. However, if one of these PDF output parameters is included in the formula field, the merged output file will be a consolidated PDF.</p>

QMode Value	Description
&QMode=SendEmail	<p>Emails the output file.</p> <p>Requires:</p> <p>&amp;TemplateID unless using only a Conga Email Template (&amp;CongaEmailTemplateID) and no document template.</p> <p>&amp;EmailTemplateId=[Salesforce Email Template ID] or &amp;CongaEmailTemplateID=[Conga Email Template ID]</p> <p>&amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</p> <p>If you cannot use the &amp;EmailToId or &amp;ContactId parameters to specify an email recipient by their Contact Id, you can instead specify &amp;EmailAdditionalTo=[email address] and &amp;EmailSubject=[subject text].</p>
&QMode=Chatter	<p>Posts the merged output file in the Salesforce Chatter feed.</p> <p>By default, Conga Trigger posts files to the Master Object record feed. To override this behavior and post a file to another record's feed, a Chatter Group or a specific User's feed, use &amp;AttachmentParentID.</p>
&QMode=Content	<p>Attaches the merged output files to the running user's personal Workspace (Library) in Salesforce Content.</p> <p>Recommended: &amp;ContentWorkspaceId=[Salesforce Content Workspace ID] to specify an alternate Content Workspace (Library).</p> <p>Does not permit saving additional attachments outside of the initial document created in your Content Workspace (for example, cannot save a copy to Attachments or Chatter in addition to your Content Workspace).</p> <p>Use this method if you are scheduling a content deliver.</p>

QMode Value	Description
&QMode=Documents	<p>Attaches the merged output files to a folder on the Documents tab.</p> <p>Requires: &amp;DocFolderId=[Salesforce Documents Folder ID]</p> <p>Does not permit saving additional attachments outside of the initial document created in your Document folder (i.e. cannot save a copy to Attachments or Chatter in addition to your Document folder).</p>
&QMode=DocuSign	<p>Sends the merged output files via DocuSign (one DocuSign Envelope per Master Object record). Requires:</p> <ul style="list-style-type: none"> <li>• &amp;DocuSignEndPoint= [See <a href="#">DocuSignEndpoint</a> for relevant values]</li> <li>• &amp;DocuSignR1ID=[Salesforce Contact, Lead, or User ID]</li> <li>• &amp;DocuSignR1Role= [See <a href="#">DocuSignR1Role</a> for relevant values]</li> <li>• &amp;DocuSignR1Type= [See <a href="#">DocuSignR1Type</a> for relevant values]</li> <li>• DocuSign Credentials on the USER record</li> <li>• The user's email address on their User record in Salesforce must match the user's email address in DocuSign.</li> </ul> <p>There are many combinations of possible DocuSign parameters. Please see <a href="#">DocuSign Integration Parameters</a> for more information.</p>
&QMode=EchoSign	<p>Sends the merged output files via Adobe eSign (one Adobe eSign Agreement per Master Object record). For more information, see <a href="#">Adobe eSign Integration Parameters</a>.</p>
&QMode=EchoSignCreate	<p>Merges the output files and attaches each to an Adobe eSign Agreement but does not automatically send those Agreements.</p>

QMode Value	Description
&QMode=GoogleDrive	<p>Saves the merged output files in Google Drive.</p> <p>Requires the Google Refresh Token on Conga Batch Setup tab.</p> <p>Use &amp;OFP to specify in which folder the merged output files are to be saved.</p> <p>Remove &amp;SC0=1 and &amp;SC1=Google from the Conga Batch formula field to prevent the storage of duplicate copies of the merged output file in Google Drive.</p>
&QMode=GoogleDocs	<p>Converts the merged output files to Google Documents, Spreadsheets, or Slides, which are saved in Google Drive.</p> <p>Requires Google Refresh Token on Conga Trigger Setup tab.</p> <p>Recommended: &amp;OFP to specify in which folder the merged output files are to be saved.</p>
&QMode=SalesforceFile	<p>Sends the merged output files to Salesforce Files for the Master Object.</p> <p>The &amp;SC0=1 and &amp;SC1=SalesforceFile parameters are required in the URL for this QMode.</p>
&QMode=Sertifi	<p>Sends the merged output files via Sertifi. The browser page refreshes and lands on the Sertifi EContract. There are many combinations of possible Sertifi parameters. For more information, see Sertifi Integration Parameters.</p>
&QMode=SpringCM	<p>Saves the merged output files in a SpringCM folder, as defined by the SpringCM configuration rules.</p>
&QMode=Box	<p>Sends the output documents to Box.</p>
&QMode=OneDrive	<p>Sends the output documents to the user's OneDrive repository.</p>
&QMode=CongaSign	<p>Sends the output documents to Conga Sign.</p>
&QMode=Contracts	<p>Sends the output documents for negotiation using Conga Contracts for Salesforce's Send for Negotiation email service.</p>
&QMode=AmazonS3	<p>Sends the output documents to AmazonS3</p>

QMode Value	Description
&QMode=LinkService	<p>Sends the output documents with Composer Link Service.</p> <p><b>Requires:</b></p> <ul style="list-style-type: none"> <li>• Composer File Storage enabled in Composer Setup.</li> <li>• &amp;CongaEmailTemplateID to designate a specific Conga Email Template</li> <li>• The {Conga_Doc_Link} merge tag to successfully generate the document link in the email.</li> <li>• &amp;TemplateID unless using only a Conga Email Template and no document template.</li> <li>• &amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</li> </ul> <div style="border: 1px solid green; border-radius: 10px; padding: 5px; margin-top: 10px;"> <p>✔ Use the NETId parameter to send Notification Emails with Composer Link Service.</p> </div>
&QMode=EmailLink	Allows you to send the composer-generated link via a unique URL in a Conga Email Template with Trigger or Batch.
&QMode=GenerateLink	Allows you to generate a unique URL for the Composer generated document and adds that link to the document history record.

### QMode parameter values for Release 7

Value	Definition
1	<p>Download the output file to your computer.</p> <p>(One of the following parameters should be used with QMode=1: LG3=1 OR LG4=1.) (default)</p>
2	<p>Attaches the merged output to the master object.</p> <p>LG4=1 is the only Activity Logging parameter needed when using QMode=2.</p> <p>Analogous to Qmode=Attachments</p>



Value	Definition
3	<p>Emails the output file.</p> <p>Requires &amp;Templateld</p> <p>For an Opportunity with a Primary Contact (Contact Roles), there is no need to set the &amp;EmailTold. Also, if you use &amp;EmailTemplateld, you must specify the &amp;EmailTold.</p>
4	Delivers the output via Adobe eSign. Requires Adobe eSign parameters.
5	<p>Attaches the output file to a folder on the Documents tab.</p> <p>QMode=5 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
6	<p>Attaches the output file to a workspace in Salesforce Content.</p> <p>QMode=6 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
7	Delivers the output via DocuSign. Requires DocuSign parameters.
94	Creates an Adobe eSign Agreement but does not automatically deliver it. Requires Adobe eSign parameters.

## Test an Upgraded Conga Trigger Solution

### To test the solution:

1. Confirm that the Conga Trigger formula field is accessible (field-level security is set to Visible) by System Administrators only. This ensures that the activated Salesforce workflow rule that initiates the Conga Trigger process won't work for anyone except System Administrators while you are still in the testing phase.
2. Locate the Salesforce workflow rule and click Description for the Outbound Message workflow action.
3. Edit the outbound message and change the Endpoint URL to `https://workflow.congamerge.com/OBMListener.ashx`.
4. Save your change.

5. Select a sample Master Object record to use for the test. Sample records are used so that the testing phase won't send documents or communications to actual recipients (customers, partners, employees, etc.). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request.
6. Trigger the Conga Trigger routine in the manner defined on the Salesforce Workflow Rule.
7. Analyze the results.

This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot.

You should confirm that the following aspects of the solution are working properly:

- Reports and queries are retrieving data to be used in the template.
- Data is merging to the template, creating the merged output file.
- Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
- Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.

## Convert a Button into a Formula Field Manually for Conga Trigger

### To manually convert a button into a formula field

1. Create a custom formula field on your Master Object with Return Type of Text and label it.  
Note: It is helpful to include the type of delivery method (ex: download, email, etc.), the template, and the master object in the Label such as, "Conga Trigger Download

Simple Proposal Oppty.”

2. Copy the button URL from your working button and paste it in your formula field's formula window.

Example:

```
/apex/csfb__Conga_Composer?SolMgr=1
&serverUrl={!API.Partner_Server_URL_370}
&userName={!$User.Username}
&Id={!bInq__Invoice__c.Id}
&TemplatId=a01800000000laT3
&FP0=1
&AC0=1
&AC1=Invoice+for+{!Opportunity.Name}
```

3. Remove the following unnecessary lines:

```
/apex/csfb__Conga_Composer?SolMgr=1&serverUrl={!API.Partner_Server_URL_370}
&userName={!$User.Username}
```

You should be left with just the &Id parameter and what follows it. For example:

```
&Id={!bInq__Invoice__c.Id}
&TemplatId=a01800000000laT3
&FP0=1
&AC0=1
&AC1=Invoice+for+{!Opportunity.Name}
```

4. Convert the remaining lines into a valid Salesforce formula.

The formula must adhere to the following rules:

- a. Literal text strings (a.k.a. static text) must be enclosed in quotes (" ... ")
- b. Literal spaces within literal text strings must be replaced with plus signs (+)
- c. Merge fields from the button must be replaced with the corresponding field available in the formula. For example, {!Opportunity.Id} would be replaced with Id

- d. Each element (literal text strings and fields) must be joined with the concatenation operator (+) or (&)

**Example:** "&id="+ Id +"&TemplateId=a0l80000000lAT3" +"&FP0=1" +"&AC0=1" +"&AC1=Invoice+for+" + Name +

- 5. Append the QMode parameter to indicate how Conga Trigger should deliver the merged output files. Example: + "&QMode=Download"

Button URL	Formula
/apex/csfb__Conga_Composer?SolMgr=1 &serverUrl={!API.Partner_Server_URL_370}&userName={!\$User.Username} &Id={!bIng__Invoice__c.Id} &TemplateId=a0l80000000lAT3 &FP0=1 &AC0=1 &AC1=Invoice+for+{!Opportunity.Name}	"&id=" + Id + "&TemplateId=a0l80000000lAT3" + "&FP0=1" + "&AC0=1" + "&AC1=Invoice+for+" + Name + "&QMode=Download"

## Convert Data Gathering Reports to Queries for Conga Trigger Solutions

Conga Trigger solutions require Conga Queries to gather additional data for merged documents. Salesforce Reports do not successfully retrieve template data for Conga Trigger solutions.

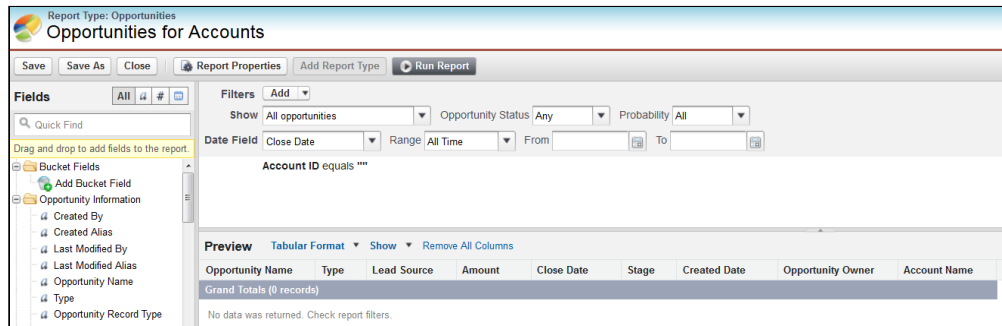
Users must convert existing reports used in Conga Trigger solutions to Conga Queries to ensure that template data merge successfully. Converting a data gathering report to a Conga Query also requires updates to merge field syntax in a Conga Trigger solution's Conga Templates.

Follow the steps below to convert reports to Conga Queries and make the necessary merge field syntax updates in.

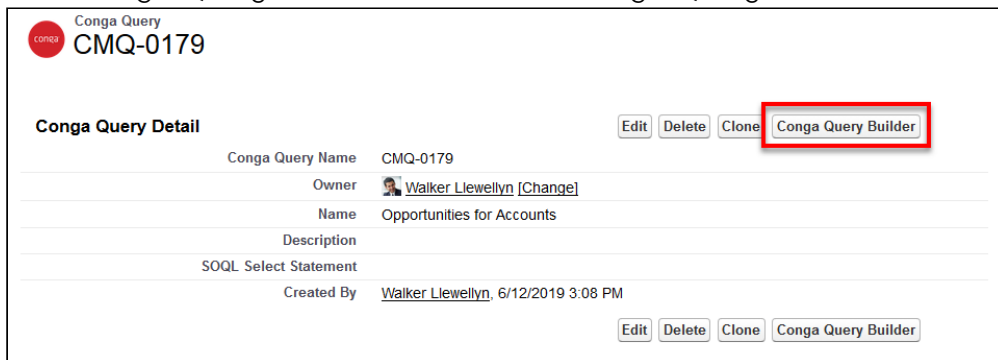
### Convert a Report to a Conga Query

1. Navigate to the Salesforce Report used in the Conga Trigger solution to reference the fields needed for the SOQL query.
  - It is recommended to keep the report open in a separate browser tab for reference.

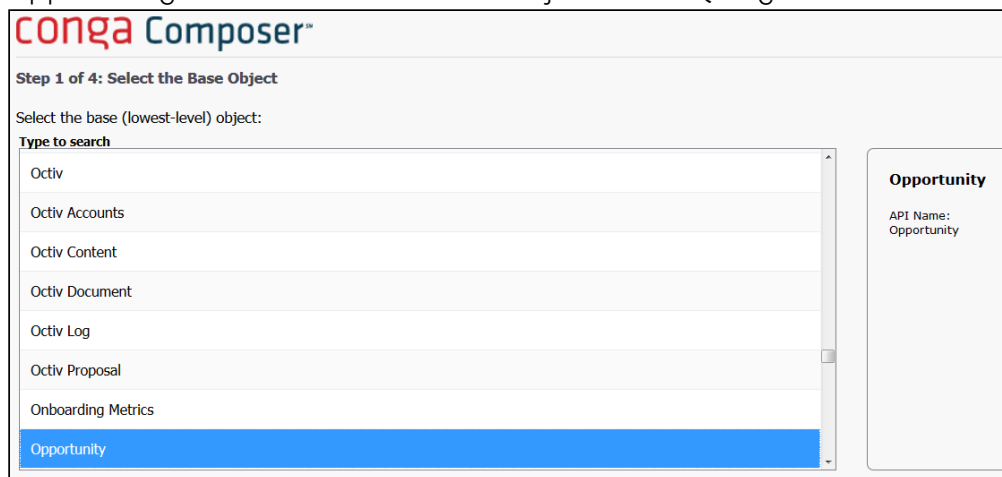
- In the example below, the report is pulling Opportunity information related to an Account record.



2. Click the Conga Queries tab.
3. Click New to create a new Conga Query record.
4. Enter a value for the Name field and click Save.
5. Click Conga Query Builder to access the Conga Query Builder.



6. Select the lowest level object as the base object and then click Next.
  - The lowest level object in example report above is Opportunity, therefore Opportunity is selected as the base object in the Query Builder.



7. Select all of the same fields referenced in the report and then click Next.

- Use the directional arrows to move fields into the Selected Fields section.

### Base: Opportunity

**Available Fields:**

- Account ID
- Amended Contract
- Auto Close
- Auto Close - Steps Not Created
- Auto-Renewal
- Background Mode
- Batch Send Proposals URL

>>>

>

<

<<<

**Selected Fields:**

- Name
- Opportunity Type
- Lead Source
- Amount
- Close Date
- Stage
- Created Date

8. Enter selection criteria that match the same filters on the report and click Next.
  - In the example above, Account ID equals "" is the only filter in the report. This filter functions as pv0 and dynamically filters the report by the specific Account record that a user initiates the Trigger solution on. To match the report's filter criteria, the Conga Query's selection criteria is to Account ID equals [null] as depicted below.

## Conga Composer™

**Step 3 of 4: Choose the Selection Criteria**

Enter the selection criteria:

Field	Operator	Value	
Account ID	equals		AND
--None--	equals		AND
--None--	equals		AND
--None--	equals		AND
--None--	equals		

Choose the sort order:

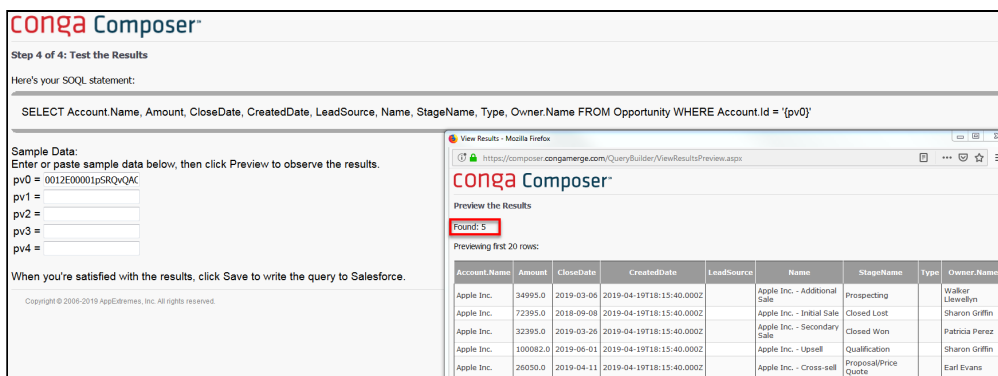
Field	Direction
--None--	ASC
--None--	ASC
--None--	ASC

Number of rows to display:

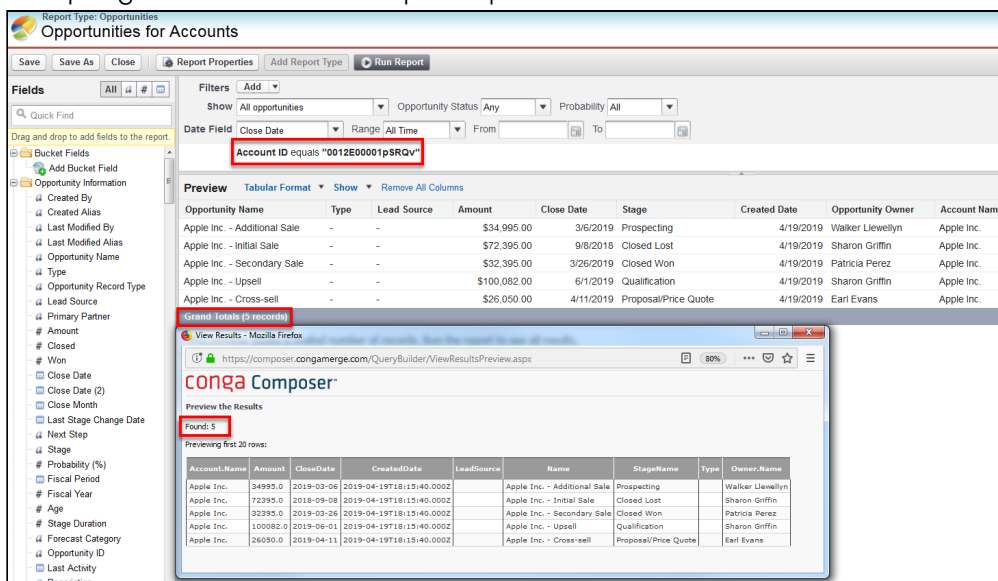
- For more information about pv0 and using Conga Query filters, see Dynamically Filter Reports and Queries with pv values.

9. Enter sample data values that fulfill the query criteria in the pv fields and click Preview.

- This ensures that the results returned from the query match the results of the report. Clicking the Preview button runs the query and shows the retrieved data.
- In this example, an Account ID is used as a sample for pv0. The specific Account record successfully returns 5 related opportunities upon clicking the Preview button.



- It is highly recommended to use the same criteria, such as a record Id, to test that the report and query both return the same results.
  - To test the report in our example, the same sample Account Id used to test the query is inserted in the report's pv0 filter.



- The report and Conga Query must return the same results as demonstrated above.

10. Click Save in the Conga Query Builder to save the query as a new Conga Query record.

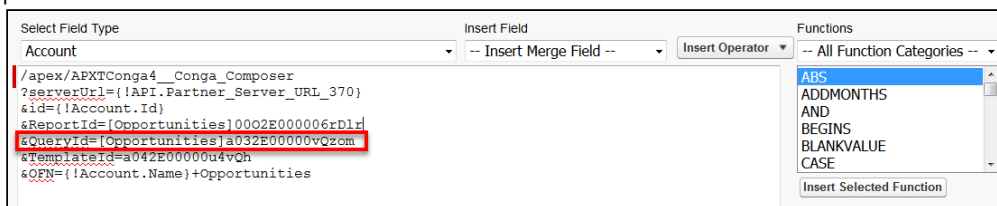
11. Repeat Steps 1-10 for each report used to gather data for a Trigger solution.

## Update Merge Field Syntax in the Template

After creating a Conga Query that returns the same results as the existing report, users must update the merge fields in Conga Template. It is recommended to first update the Conga Trigger solution's corresponding button to obtain the new merge field syntax and test the updated template.

For more information on a Conga Trigger solution's corresponding button, see [Convert a Conga Composer Button to a Formula Field Using the Conga Formula Builder](#) and [Edit a Conga Composer Button and Formula Field Using the Formula Builder](#).

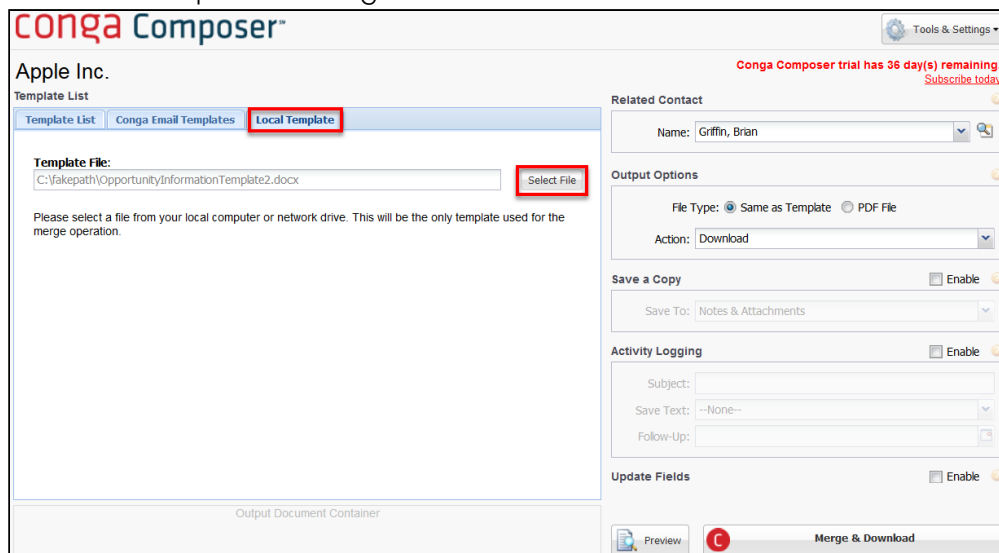
1. Copy the Conga Query's 15 digit record Id from the browser URL.
2. Click the Conga Workflow Formula Builder tab.
3. Select the Master Object that Trigger formula field lives on and then select the Composer button that corresponds to the Trigger formula field.
4. Click Edit Button.
5. Add the QueryID parameter and use the copied Conga Query record ID as the parameter value.



- Use the same alias as the previous ReportID parameter. Keeping the same alias for the QueryID parameter prevents the query dataset from being renamed and results in less work when editing merge field syntax in the template. In the example above, the alias for the Query ID is Opportunities which were also the alias for the ReportID parameter.
  - If an existing alias for the ReportID parameter does not exist, choose and enter a new alias for the QueryID parameter value.
6. Remove the ReportID parameter from the Composer button URL and click Save.
  7. Navigate to a sample object record and click the recently edited Composer button to access the Composer user interface.
    - If present, remove the DS7 parameter.
  8. Click Tools & Settings in the Composer window and then click Template Builder.
  9. Click the + button next to the new query dataset to display all available merge fields.
    - Use the directional arrows to select merge fields from the new dataset.
  10. Copy and paste the merge field syntax into the template to replace the previous dataset's merge field syntax.



- If the same alias is used for the QueryID parameter, the TableStart:dataset syntax remains the same and does not require updating. In the example above, TableStart:Opportunities and TableEnd:Opportunities is the correct syntax for both the ReportID and QueryID parameter values.
  - Although the report and query retrieve the same fields, merge field syntax produced from queries may differ than merge syntax produced from reports. Ensure that all merge fields produced from the query use the correct syntax taken directly from the Template Builder.
11. Save the new template file locally.
  12. Click the Back arrow button in the Composer window.
  13. Navigate to the Local Template tab and click Select File to select the new template version with updated merge fields.



14. Click Preview or Merge & Download to merge and download the local template
  - Ensure that all merge fields successfully merge data.
15. (Optional) Repeat Steps 1-14 for all new Conga Queries replacing reports.
16. Click the Conga Templates tab.
17. Click the Conga Template record(s) referenced in the Composer button and Conga Trigger formula field.
18. Click Upload Files and upload the new template file to the existing Conga Template record.

If the Conga Trigger formula field was built manually (without the Formula Builder) and a corresponding button does not exist, it is highly recommended to build a temporary button to access the Template Builder. A temporary button allows users to easily test and troubleshoot the modified template with the new merge field syntax.

## Update the Conga Trigger Formula Field

After confirming that data successfully populates in the corresponding button, it is time to update the Conga Trigger formula field.

1. Click the Conga Workflow Formula Builder tab.
2. Select the Master Object that the Trigger formula field lives on and then select the button that corresponds to the Trigger formula field.
3. Click Update Field.

The screenshot shows the Conga Formula Builder interface. The 'Update Field' button is highlighted with a red box. The interface displays the following information:

- Master Object:** Account
- Output Mode:** Attach to Master Record as Salesforce File
- Button / Link:** Opportunity Information
- Formula Field:** Opportunity Information
- Formula / URL:**

```

/apex/APXTConga4__Conga_Composer
?serverUri={API.Partner_Server_URL_370}
&id={!Account.Id}
&QueryId={Opportunities}a032E00000vQzom
&TemplateId=a042E00000u4vQh
&OFN={!Account.Name}+Opportunities
&defaultpdf=1

```
- Formula:**

```

"&id=" + Id +
"&queryid={Opportunities}a032E00000vQzom" +
"&templateid=a042E00000u4vQh" +
"&ofn=" + Name + " Opportunities" +
"&qmode=SalesforceFile"

```

- The formula field then updates with the new QueryID parameter(s)
4. Test the Conga Trigger solution on a sample record to ensure that all data populates successfully in the output document.

## Replace Ampersands in Reference Field Data

Conga formula fields use ampersands (&) to identify parameters. If you reference fields that might contain ampersands or illegal characters as data values, you must replace the ampersands and illegal characters with a character string that Conga apps recognize as a substitute such as: "{~~~}".

This function is useful for users who are replacing illegal characters with spaces or other legal characters but {~~~} does not turn into a "&" upon merge.

Use the following string to replace ampersands within merge field values at merge time:

```
SUBSTITUTE (fieldname, "&", "{~~~}" )
```

This formula field references the Account Name field in the OFN (output file name) parameter:

```
"&OFN=" + Account.Name + "Case+Review"
```

Since some Account Names might contain an ampersand, you should replace the ampersands like this:

```
"&OFN=" + {!SUBSTITUTE (Account.Name, "&", "{~~~}" )} + "Case+Review"
```

In the example below, the `Account.Name` field for the merged record contains the company name of Merging & Batching Inc. The merge fails without substituting the ampersands within field values. With the string to substitute ampersands at merge time, however, the merge is successful, as shown in the table below:


	Incorrect	Correct
<b>Input</b>	"&OFN=" + Account.Name	"&OFN=" + {!SUBSTITUTE (Account.Name, "&", "{~~~}" )}
<b>Result</b>	ERROR	Merging & Batching Inc.

## Integrate Conga Trigger with Google Drive

Conga Trigger offers many options for saving the merged output files. Files can be stored in Salesforce (for example, as attachments, Chatter files, or Content files) or in Google Drive.

### To integrate Conga Trigger with Google Drive

1. Click the Conga Trigger Setup tab, and then click Create Google Token.
2. Sign into the appropriate Google account. A Google dialog box appears listing the permissions that is requesting.

 The Google account holder must have access to the folders where merged output files will be saved.

3. Click Accept.
4. Click OK on the dialog box confirming that the Google refresh token has been saved successfully.

The Google Refresh Token area in Conga Trigger Setup now displays as a Valid Token.

## Conga Trigger Caveats and Warnings

Helpful tips and warnings for using Conga Trigger.

- Conga Trigger is limited to 500 transactions per 24-hour period. Conga Trigger is a transaction-based service. You will be charged per transaction (a.k.a service event).
- While not generally necessary, set a 12-hour session timeout value when using Conga Trigger or Scheduled Conga Batch to increase the likelihood that a session is still active when the document processing takes place.
- If you're updating records with a tool like the Salesforce Data Loader, we strongly recommend you use the credentials of a user whose Profile has the "Send Outbound Messages" permission disabled to avoid inadvertently triggering mass execution of Conga Trigger.
- Keep in mind that the number of outbound emails permitted through Salesforce is dependent on your Salesforce edition. Consider using 's integration with [SendGrid](#) if you need to exceed the Salesforce email limits.
- Each Conga Trigger merge operation consumes API requests; the number varies depending on the nature and complexity of the solution. Therefore, you'll need to gauge the number of Conga Trigger operations that will take place in your instance with the number of API requests you're allowed in a 24-hour period.
- Since an Outbound Message operates under the authority of a named user, be careful to ensure that you do not trigger a merge operation that exposes more information than a user would normally have the privilege to access. Consider using a collection of workflow rules and outbound messages that execute depending on the identity of the end-user.
- When using Conga products and DocuSign in a fully automated fashion (such as using the background parameter/value pair &DS7=17, Conga Trigger, or Conga Batch) the DocuSign Username and Salesforce User Email must match. If not, you will receive this error: Error: CreateEnvelope in DocuSignAPI. The UserID does not have a valid membership in this Account.
- User fields normally made available when executing a standard solution will, in the case of a Conga Trigger-triggered operation, reflect the values represented by the user under whose authority the workflow solution is running (the User to Send As on the outbound message). Thus, an invoice template with a merge field intended to retrieve the name of the person clicking a button/link will instead populate with the name of the user-defined as the User to Send As on the Outbound Message.
- Ensure that ampersands (&) in text fields passed via Conga Trigger have space before and after the ampersand character. For more information, see [Replace Ampersands \(&\) in Reference Field Data](#).

- Conga Trigger is intended to be executed against individual records. While it is possible to update many records in Salesforce, resulting in a batch of Conga Trigger requests, the use of Conga Trigger in this manner risks tripping an automatic Conga system governor designed to prevent a “runaway” process from “looping” uncontrollably and potentially harming certain systems.

## Troubleshooting Tips for Conga Trigger Solutions

When troubleshooting, one of the first things to review is the Conga Trigger formula field. The primary factors that affect this field’s behavior are:

- Field accessibility (the field is restricted to certain profiles)
- Whether it contains a value
- The field contains or references fields containing ampersands that don't have spaces around them when intended as data
  - A field value of Ben&Jerry's will cause an error. However, the field value of Ben & Jerry's does not. Conga Trigger can't distinguish between the data and the parameters if the ampersands within the data aren't surrounded by spaces.
- The user who initiates the workflow process must have the "Send Outbound Messages" permission enabled, otherwise the email will not be sent. The User to Send As is the user under whose authority Conga Trigger will run. It is highly recommended that the User to Send As is a full System Administrator and has a license. Ensure that this user is active and has access to all components used in the Conga Trigger solution (reports, queries, objects, fields, email privileges, eSignature privileges), as well as access to the Conga Trigger formula field.

## Conga Trigger daily limit email alerts

You can set up an email alert to subscribers when an Org's daily Conga Trigger is nearing its limit set by the administrator. The administrator can set an email alert based upon a percentage of their Conga Trigger Usage status. The Conga Trigger Dashboard now contains two new fields for alerts: Alert Percent and Email Alert To

- Alert Percent - Can be set to 1 to 100 percent once the user's Conga Trigger usage per 24 hours reaches its threshold an email alerting the user triggers.
- Email Alert To - Email address for the user to be alerted once the Alert Percent threshold is reached.

## To set up Conga Trigger usage limit email alerts

1. Navigate to the Conga Trigger Dashboard tab in Salesforce.

2. In the Alert Percent field enter or select the percentage of Conga Triggers out of the daily maximum allotment for your org when an Alert email is triggered.
3. In the Email Alert To field enter the email who the alert is sent to.
4. Click Save Conga Trigger Usage Alert to save your settings.

The email alert is only sent once per 24 hours when the threshold is reached.

**Conga Workflow™**

**Workflow Dashboard**

Org ID:  Alert Percent:  Workflow Usage Status: Good  
 24-Hour Limit:  Email Alert To:  12 operations used out of 50000 (24-hour limit)  
 Expires:    
\*Receive an email alert when Workflow usage reaches or exceeds a percentage of your daily allotment.

Usage Detail (Last 24 hours)					Error List		
Date	Requested	Queued	Status	Comment	Action Id	Record Id	Error
05-Sep-2017 10:39:04 AM	1	1	Success				
05-Sep-2017 3:40:21 AM	1	1	Success				
05-Sep-2017 3:38:33 AM	1	1	Success				
05-Sep-2017 3:36:54 AM	1	1	Success				
05-Sep-2017 3:35:12 AM	1	1	Success				
05-Sep-2017 3:33:08 AM	1	1	Success				
05-Sep-2017 3:31:02 AM	1	1	Success				
05-Sep-2017 3:29:17 AM	1	1	Success				
05-Sep-2017 3:25:58 AM	1	1	Success				
05-Sep-2017 3:24:11 AM	1	1	Success				
05-Sep-2017 3:20:19 AM	1	1	Success				
05-Sep-2017 3:18:32 AM	1	1	Success				

Page 1 of 1 | Displaying records 1 - 12 of 12 | Page 0 of 0 | No records to display

To remove the Conga Trigger usage alert, simply remove the email address from the Email Alert To field and click Save.

## Troubleshooting Conga Trigger Using Hyperlink Formula Fields

Conga recommends the use of the Hyperlink Formula Field technique as a first step whenever challenges arise in the use of Conga Trigger. This hyperlink field will be placed on the Master Object page layout and allows the system administrator to run the Conga Trigger process without having to invoke the Salesforce workflow rule.

The hyperlink field launches the user interface, but runs the Conga Trigger solution, thus providing all of the user interface capabilities of (such as the View Data workbook, templates, user license status, and error messages) while running Conga Trigger.

This approach also helps identify problems associated with access to templates, reports, and queries that can cause Conga Trigger process failures resulting from Salesforce session timeouts.

To create a hyperlink formula field to troubleshoot a Conga Trigger solution:

1. Create a custom formula field on your Master Object.

**New Custom Field**

**Step 1. Choose the field type**

Specify the type of information that the custom field will contain.

**Data Type**

None Selected  
 Auto Number  
 **Formula**  
 Roll-Up Summary

2. In the Field Label box, enter the Test and the solution name.

**Step 2: Choose output type**

Field Label: **Test Conga Workflow Invoic** Field Name: **Test\_...**

**Formula Return Type**

None Selected  
 Checkbox  
 Currency  
 Date  
 Date/Time  
 Number  
 Percent  
 **Text**

Select one of the data types below.

Calculate a boolean value. Example: `TODAY() < CloseDate`  
 Calculate a dollar or other currency amount and automatically format the field as a currency. Example: `Gross Margin + Amount__c`  
 Calculate a date, for example, by adding or subtracting days to other dates. Example: `Reminder Date + CloseDate__c`  
 Calculate a datetime, for example, by adding a number of hours or days to another datetime. Example: `Test + NOW() + 1`  
 Calculate a numeric value. Example: `Estimated__c * 30`  
 Calculate a percent and automatically add the percent sign to the number. Example: `(Amount - Encountered_Amount__c) / Amount`  
 Create a text string, for example, by concatenating other text fields. Example: `Full Name = LastName & ", " & FirstName`

3. In the Formula Return Type area, click Text.
4. Select the Advanced Formula tab and, in the Functions area, select HYPERLINK, and click Insert Selected Function.

**Step 3: Enter formula** Step 3 of 5

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `Full Name = LastName & ", " & FirstName` More Examples...

Simple Formula  
 **Advanced Formula**

Test Conga Workflow Invoice Email (Text) =

`HYPERLINK(url, friendly_name [, target])`

**Functions**  
 -- All Function Categories --  
 EXP  
 FLOOR  
 GEOLOCATION  
 GETSESSIONID  
**HYPERLINK**  
 Insert Selected Function  
 HYPERLINK(url, friendly\_name [, target])  
 Creates a hyperlink

5. In the HYPERLINK function string, replace URL with the following:  
Lightning:

```
"/apex/APXConga4__Conga_Composer?ServerURL="+ $Api.Partner_Server_URL_370 +
```

Classic:

```
"https://composer.congamerge.com/?SessionId="+ $Api.Session_ID + "&ServerURL="
+
$Api.Partner_Server_URL_290 +
```

6. Enter a new line after the URL and enter the API name of the formula field.
7. Click Insert Field to insert the field.
8. In the HYPERLINK function string, replace friendly\_name with the name of your hyperlink as it appears on the page layout.
9. In the HYPERLINK function string, delete [, target]. Your full hyperlink formula field would then be:


Lightning:


```
HYPERLINK("/apex/APXTConga4__Conga_Composer?ServerURL="+
$Api.Partner_Server_URL_370 +
Conga_Workflow_Invoice_Email__c,"Test Conga Trigger via User Interface")
```

Classic:

```
HYPERLINK("https://composer.congamerge.com/?SessionId=" + $Api.Session_ID + "&Se
rverURL=" +
$Api.Partner_Server_URL_290 + Conga_Workflow_Invoice_Email__c,"Test Conga
Trigger via User Interface")
```

10. Click Next to establish field-level security.

 We recommend making this field visible to only the System Administrator profile.

 If you use Salesforce Professional Edition, the hyperlink troubleshooting field will be visible to all users due to the lack of field-level security. Therefore, we recommend including the &DC (Disable Conga Composer) parameter with an IF function to limit access to this field to System Administrators only. See the




```
example parameter below."&DC=" + IF( $Profile.Name = "System
Administrator", "0", "1")
```

11. Add the field to the appropriate page layouts.
12. Click Save.

You now have a hyperlink field on your Master Object page layout to run your solution without having to initiate the Salesforce Conga Trigger rule.

You can use this troubleshooting link to launch Conga Trigger through the user interface to view error messages, download the View Data workbook, and so on.

 QMode parameters do not function when you run Conga Trigger using this technique, so you must run your solution using the appropriate action from the user interface (such as Merge & Download or Email).

## How do I see all errors from a request?

### Question

How do I see all of my errors in from a schedule merge request in Conga Trigger?

### Causes


The dashboard shows the latest results in a 24-hour period. Only 500 lines of results are shown on the dashboard so you cannot see errors that came after the top 500 results were returned.

### Workaround

You can export your results/data set. To view your results, you can view the recent activity report. To export your data set and see your activity in the report, go to Activity Report Overview.

## Conga Trigger Dashboard - No Status Generated for Failure

When the Conga Trigger dashboard shows actions as "requested," but does not include a success or failure in the Status column, the issue is likely caused by field-level security on the source formula field.

 The Conga Trigger formula field needs to be visible to the "User to Send as," which is defined in the outbound message of the Conga Trigger rule.

Example of this issue in the Conga Trigger dashboard:

The screenshot shows the Conga Workflow Dashboard. At the top, it displays 'Conga Workflow' and 'Workflow Usage Status' as 'Good' with '0 operations used out of 500 (24-hour limit)'. Below this, there are fields for 'Org ID', '24-Hour Limit', and 'Expires'. A red arrow points to the 'Usage Detail (Last 24 hours)' table. The table has columns for Date, Requested, Queued, Status, and Comment. The data shows two records from 06-Oct-2016, both with 1 Requested and 0 Queued operations. To the right, the 'Error List' table is empty, showing 'There are no errors for this record.'

## Conga Batch

Conga Batch® is an optional, paid add-on that launches on-demand or scheduled solutions in batches.

Select one of the following topics for more information:

- [Conga Batch Installation and Setup](#)
- [Conga Batch Product Guide](#)
- [The Conga Batches Tab](#)
- [The Conga Batch Dashboard Tab](#)
- [The Conga Batch Setup Tab](#)
- [The About Conga Batch Tab](#)
- [Building Conga Batch Solutions](#)
- [Troubleshooting Conga Batch](#)

## Conga Batch Installation and Setup

Easily install and setup Conga Batch using the following documentation.


- [About Conga Batch](#)
- [Install Conga Batch](#)
- [What's in the Conga Batch Package?](#)
- [Working Conga Composer Solutions](#)
- [Create a Conga Batch Record](#)
- [Convert a Conga Composer Button to a Conga Batch Formula Field Using the Conga Formula Builder](#)
- [Choose How to Launch Conga Batch](#)
- [Create a Conga Batch List View Button](#)
- [Determine Which Records to Include in Your Batch](#)

- [Launch Conga Batch](#)
- [Schedule Conga Batch to Run Automatically](#)

## About Conga Batch

Conga Batch® is an optional, paid add-on that launches on-demand or scheduled solutions in batches. Conga Batch saves you time, effort, and money when you need to run the same solution for many records at once or on a recurring interval.

## How Conga Batch Works

<p>Choose what records to include in a batch</p>	<p>Predefined - Use Salesforce® standard reports or SOQL queries to select records. Ad hoc - Make selections from a list view or related list view.</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin-top: 10px;"> <p> While Salesforce Reports are useful to gather data for Composer solutions, it is highly recommended to use Conga Queries due to better data gathering performance, versatility, and stability.</p> </div>
<p>Choose how to deliver your files</p>	<p>Consolidated - Files from up to 200 records can be batched and consolidated for immediate download as a single PDF or .ZIP file.</p> <p>Distributed - Individual files from up to 1000 records can be batched for distribution as:</p> <ul style="list-style-type: none"> <li>• Email attachments</li> <li>• eSignature documents (Conga Sign, DocuSign®, Adobe eSign®, and Sertifi)</li> <li>• Salesforce record attachments, Documents, Chatter® posts, Content records, Google Drive, and SpringCM</li> </ul>
<p>Choose when it happens</p>	<p>On Demand - Launch from a button on a list view, page layout, or Conga Batch record. Scheduled - Runs in the background hourly, daily, weekly, or monthly.</p>

## Install Conga Batch

**i** Installing Conga Batch is OPTIONAL. You do not need to install Conga Batch to get started with Conga Composer. Conga Batch is an add-on to and must be installed separately from Conga Composer. Additionally, Conga Batch must be activated by the Conga support team.

**⚠** Tabs in the Conga Batch managed package count against your Salesforce organization's tab limit. Be mindful of this when installing.

### To install Conga Batch:

1. Click the appropriate link to install Conga Batch in your Salesforce environment:
  - [Install Conga Batch in a Sandbox org](#)
  - [Install Conga Batch in Production org](#)

**⚠** Conga Batch requires your trial to be enabled by Conga.

2. Enter the Salesforce credentials for the instance in which you want to install Conga Batch and log in to Salesforce.
3. Select a security level for the Conga Batch application.

**⚠** Conga recommends selecting Install for All users, as anyone who launches batched merges via through Conga Batch requires access to the Conga Batch app components.

4. Click Install.
  - Installation Complete! appears, confirming the Conga Batch package has been installed successfully.
5. Click Done.

## What's in the Conga Batch Package?

The Custom Tabs and Objects in the Conga Batch Package.

After installing the Conga Batch package, you will see the following components in your Salesforce Org:

- Conga Batches: A custom object from which batch processes are launched or scheduled. The Conga Formula Builder is accessed from a Conga Batch custom object record. Conga Batches is also a custom tab.
- Batch Dashboard: A custom tab to monitor batches and view errors for any records that failed to process.
- Formula Builder: A custom button on the Conga Batch record used to convert a button or link to a Conga Batch formula field.
- Conga Batch Setup: A custom tab to administer Conga Batch, including enable automatic scheduling of Conga Batch, restrict API requests by setting an API floor, and integrate with Google Drive.
- Scheduled Batch History: A custom object (child of the Conga Batch custom object) that displays schedule details for any Conga Batch records that run on a scheduled basis.
- About Conga Batch: A custom tab describing Conga Batch capabilities and how to begin the configuration process.
- Various fields, buttons, and Visualforce pages

## Working Conga Composer Solutions

You need a working Conga Composer solution to use Conga Batch.

- Test your Composer solution to ensure all aspects of it (the template, activity logging, integrations, etc.) are working as desired. Also, your solution must be compatible with Background Mode (but does not have to be in Background Mode). This means you must pre-define the template(s) to use in the merge process as well as the delivery method (download, email, etc).
- You will convert this working Composer solution into a Conga Batch solution using the Conga Formula Builder. For more information on Conga Composer solutions, please see Conga Composer.
- Conga Batch is an add-on to an active Conga Composer subscription and is purchased separately. The Conga Batch service must be activated by Conga before use.
- Please contact support to subscribe or start a free trial.

## Create a Conga Batch Record


The Conga Batch record serves multiple purposes:

- Provides access to the Conga Formula Builder, which automatically converts your button or link into a Salesforce formula field.

- Serves as the launch point for Conga Batches, hence much of this guide referring to launching Conga Batch from a Conga Batch record.

## To create a Conga Batch record:

1. Click the Conga Batches tab.
2. Click New to create a new Conga Batch record.
3. Give your Conga Batch record a Title and optionally a Description. We find it helpful to include the type of delivery method (e.g. download, email, etc.), the template, and the Master Object in the Title or Description such as, "Download Proposal Oppty." If you're launching Conga Batch from a List View button, we recommend including "To be deleted" in the Title or Description fields, as the Conga Batch record is only used temporarily.
4. Add the Conga Batch formula field's API name (for example: Batch\_formula\_field\_\_c) to the URL Field Name field.

 If the Conga Batch formula field is not yet created, add placeholder text in the URL Field Name field in the interim to create the Conga Batch record. Replace the placeholder text later when the Conga Batch formula field is created.

5. For Conga Batch solutions that will be launched from the Conga Batch record, enter the Salesforce ID from a report or query into the corresponding field (Report Id or Query Id). Most of the time, you will use Report Id or Query Id as the Master Object Source. However, instead of running a full batch using a report or query, you may run a single record using Record Id to test the Conga Batch solution. Otherwise, Record Id is only used in rare situations with specific requirements. Do not confuse the report or query you select in the Master Object Source with those that you may have referenced in your solution; the Conga Batch report or query is simply used to determine which records in Salesforce you wish to include in the batch operation while reports and queries are used to gather detail data that are referenced on the template.
6. Click Save.

## Convert a Conga Composer Button to a Conga Batch Formula Field Using the Conga Formula Builder

If you are on an older version of Batch you might need to upgrade, otherwise, these instructions apply to you: If you are trying to convert an OAuth button, Formula Builder


does not recognize it. The current recommendation is to convert to a non-OAuth button, create the formula with Formula Builder, then change the button back.

Unlike a Conga Composer button or link which contains a URL with various parameters to define your merge operation, Conga Batch uses a formula field on your Master Object to host those instructions. Use the Conga Formula Builder to convert a working Conga Composer button or link into a Salesforce formula field used by Conga Batch.


If you prefer to build your formula field manually, see [How to Manually Convert a Conga Button into a Conga Batch Formula Field](#).

## To convert a Conga Composer button to a formula field using the Conga Formula Builder:

1. Click the Formula Builder button to launch the Conga Formula Builder.
2. Choose the Master Object from the drop-down menu. The Master Object is the object where your button or link is located.

 The Master Object is pre-selected if you have entered a Report or Query in Master Object Source on the Conga Batch record.


3. Select your desired Output Mode. This adds a corresponding QMode parameter to your formula field. Additional parameters may be required based on the Output Mode selected. You may need to modify your button/link to include those required parameters. For more information, see [Determine How Conga Batch Will Deliver the Merged Output Files](#).

 If your button includes a DS7 parameter (to run in background mode), the Output Mode is pre-selected to match the background mode behavior.


- Important notes about Google Output Modes
  - The Output Modes “Attach the output to Google Drive” and “Convert the output to a Google Doc” require a Google Refresh Token, which is configured from the Conga Batch Setup tab.
  - The merged output files from Google Drive or Google Doc Output Modes are stored in the Google account of the user who creates the Google Refresh Token.
  - buttons or links configured to save a copy of the merged output file to Google Drive will include the parameters &SC0=1 and &SC1=Google. However, if these parameters are maintained in the Conga Batch formula field and “Attach the output to Google Drive” is the Output Mode, Conga Batch will store two copies of each merged output file in Google Drive (one from the &SC0=1 and &SC1=Google parameters, and one from the “Attach

the output to Google Drive” Output Mode). Therefore, if “Attach the output to Google Drive” is your desired Output Mode, you should remove the &SC0=1 and &SC1=Google parameters from your Conga Batch formula field.

4. In the Button / Link area, in the Name box, select the solution. The button or link automatically converts to a formula field and display in the left pane for Formula / URL. Only traditional URLs for buttons or links display in the Button / Link Name drop-down. URLFOR functions, JavaScript buttons, etc. are not supported.
5. (Optional) Clear the Show Converted Formula box to toggle back to the URL view of the button/link.
6. In the Formula Field area, rename your button by typing a name in the Name box. We recommend “Conga Batch [Output Mode] [Name of solution],” for example, “Conga Batch Email Proposal.”
7. Click **Create Field**. A dialog box confirms the creation of the field and asks if you’d like to update the Conga Batch record.

 Salesforce Professional Edition customers will not be able to use the Create Field button to have Conga Batch create the formula field automatically (due to API limitations). Instead, copy the contents of the Formula / URL pane and manually create the formula field in Salesforce Setup on your Master Object. Lastly, enter the API name of this field into the URL Field Name field on the Conga Batch record (seen below in step 9).

8. Click No to edit field-level security of the newly-created formula field OR Click Yes to update the field without editing field-level security.
9. Click Edit Field-Level Security. Field-level security for your formula field opens in a new browser tab.
10. Edit as necessary and save changes.

 If you do not change your field-level security, your solution will not work (you will receive the error message "Object Reference not set to an instance of an object."). You must edit field-level security even if the person creating the formula field has a Standard Salesforce Administrator profile.

11. Return to the Formula Builder browser tab.
12. Click Update Conga Batch.

Click OK to close the confirmation dialog box.

- In Salesforce instances on Enterprise Edition and higher, this formula field is not added to the Master Object’s page layout; rest assured, Conga Batch functions



correctly without the field on the page layout. However, for testing purposes, you may want the field displayed on the page layout, at least temporarily.

- Salesforce Professional Edition customers must place the formula field on the Master Object's page layout.
- If you put the formula field on a page layout and would like to keep it there, we find it helpful to position it within a new section on the page layout so the section may be collapsed as desired.

## Choose How to Launch Conga Batch

There are two different ways to launch Conga Batch - either through a [Conga Batch record](#) or a [List View Button](#). You might configure some Conga Batch operations to be launched by a Conga Batch record and others by a List View button, and each method has its advantages and limitations.

### Launching from a Conga Batch Record

This method allows you to select the records to include in your batch using a Salesforce report or SOQL query. It also allows you to schedule your Conga Batch to run on a schedule with no user interaction. This method is preferred if you want to schedule your batches to run with no user interaction.

### Launching from a List View Button

This method allows you to manually select which records to include in your batch (up to 75 records). You can choose to add the List View button to a Search Layout or to an Object Page Layout, letting users choose the records from a List View or a Related List.

### Other Considerations

If your output method is an immediate download as a single PDF or .zip file (consolidated output), your batch is limited to 200 records regardless of which launch option (above) you choose.

If you are emailing the output (known as distributed output), sending for eSignature, or storing in another location (like Salesforce or Google Drive, you can include up to 1000 records in a batch (if launching the solution from a Conga Batch record).

These considerations can help you decide how to launch your Conga Batch solution.

## Create a Conga Batch List View Button

This is a Javascript button and Javascript is not supported in Salesforce Lightning. This feature is not supported in the Salesforce Lightning user experience. For a Lightning-ready Visualforce button, see [Create a Conga Batch List View Button \(Lightning Experience\)](#)

Setting up a List View button to launch Conga Batch is easy. You create a button on the Master Object, paste the code from the Conga Batch Sample List View button, adjust the code to your needs, and place the button on the object's Search Layout.

**⚠** Due to their complicated nature, the integration of Conga with JavaScript, Visualforce, and Apex (or other programming languages) is outside the scope of support available from the Technical Customer Success team.

### To create a Conga Batch List View button:


1. Create the button. Use the JavaScript code below

```
// STEP 1: CHANGE THE NEXT LINE TO THE TYPE OF RECORDS YOU'RE ACCESSING ON THE
VIEW var idArray = {!GETRECORDIDS( $ObjectType.Opportunity )};
// STEP 2: CHANGE THE NEXT LINE TO THE API NAME OF THE FIELD THAT CONTAINS THE
CONGA COMPOSER URL var urlFieldName="ComposerFormula_URL__c";
// DO NOT MODIFY ANYTHING BELOW THIS POINT *****
// YOU MAY PASS UP TO 50 IDS var CongaURL = "https://conductor.congamerge.com"
+ "?MOID=" + idArray + "&SessionId={!$Api.Session_ID}" + "&ServerUrl={!
$Api.Partner_Server_URL_160}" + "&urlFieldName="+urlFieldName;
window.open( CongaURL, "Conga", "width=700,height=450,menubar=0" );
```

2. Create a button on your Master Object. This Master Object needs to be the same as the Master Object you specified in the [solution that you converted into a Conga Batch formula field](#).
3. Copy and paste the OnClick JavaScript code from Step 1.
4. Label the button.
5. Set the Display Type to List Button and enable Display Checkboxes.
6. Ensure the button's Behavior is set to Execute JavaScript.
7. Set the Content Source as OnClick JavaScript.
8. Set the \$ObjectType to your Master Object by choosing \$ObjectType from the Select Field Type picklist and then choosing your Master Object from the Insert Field picklist.

Note: In this example, the Master Objects from our solution is Opportunity, so the line is set to \$ObjectType.Opportunity.

9. Set the urlFieldName value to the API name of the formula field you created with the Conga Formula Builder. In this example, our formula field API name is Conductor\_URL\_Simple\_Proposal\_\_c.
10. Edit the Master Object List View under Search Layouts. For example, since we are using the Opportunity as our Master Object, we navigate to Build → Customize → Opportunities → Search Layouts Opportunities List View.

 We recommend using Enable Enhanced Lists under Salesforce Setup → Customize → User Interface to best utilize List View buttons.

11. Click the Add button to move your List View button to Selected Buttons menu. Click Save.
12. The configuration is complete and your Conga Batch List View button is ready to use.

## Example button syntax:

```
var idArray = {!GETRECORDIDS( $ObjectType.Opportunity )};
var urlFieldName="ComposerFormula_URL__c";
var CongaURL = "https://conductor.congamerge.com" + "?MOID=" + idArray +
"&SessionId={!$Api.Session_ID}" + "&ServerUrl={!$Api.Partner_Server_URL_160}" +
"&UrlFieldName="+urlFieldName;
window.open( CongaURL, "Conga", "width=700,height=450,menubar=0" );
```

## Determine Which Records to Include in Your Batch

When launching Conga Batch from the Conga Batch record, a Salesforce report or SOQL query is used to determine which Salesforce records are included in the batch operation. The report or query is also referred to as the “Master Object source.” You might only use one Master Object source – either a report or a query, not both. However, queries tend to run faster than reports and work better with larger data tables.

 Queries are faster and more reliable than Salesforce Reports.

## Build a SOQL Query for the Master Object Source


Instead of a Salesforce report, you may use a SOQL query as the Master Object Source. Queries are typically used when reports are not available due to Salesforce security

settings or if the desired records cannot be retrieved using a report. Just like using a report as the Master Object Source, Conga Batch uses this query to determine which records are included in the batch operation. The Salesforce ID of the Conga Query record will be referenced in the Query Id field under Master Object Source on the Conga Batch record.

Referencing a SOQL query as the Master Object Source requires the Conga Queries object. If you do not have this object installed in Salesforce, please contact Conga Support for assistance.


## To build a SOQL query for the master object source:

1. Create the Conga Query record.
  - a. Give the record a Name and, optionally, a Description.
  - b. We recommend naming the query as "Conga Batch - [Name of Conga Batch Record]." Continuing with the same example, our name is "Conga Batch - Batch Simple Proposal - Oppty." Save the record.
  - c. Ensure that the user who runs the Conga Batch solution has access to: the Conga Queries object, the fields on the object and the specific Conga Query record.
2. Launch Conga Query Builder to write the SOQL Select Statement.
  - The base object you select when building your Master Object Source query must match the Master Object specified in the solution you converted into a Conga Batch formula field (see [Convert a Button to a Formula Field Using the Conga Formula Builder](#) for further details).
  - In other words, if you built your solution with the Opportunity as the Master Object, the base object for your Master Object Source query for Conga Batch will also be Opportunity.

 The Conga Batch formula field is located on the Master Object. This Master Object must be selected as the base object (the "FROM" object) in the query. If you reference ANOTHER object in the Where clause, Conga Batch cannot handle this level of complexity.

3. Select the Master Object ID as the first field in the query.
  - This is the only field that is required for Conga Batch to read the query and determine which records to include in the batch.
  - You can include other fields in the query to assist in your reading of the query (like a name field) as desired, keeping in mind that the fewer fields included on the query, the faster it runs.

4. Define the selection criteria of the query so that the proper records are included in the batch.
  - You can also choose a sort order and limit the number of rows as needed.

 Optimize the processing time of your query and the overall speed of the Conga Batch solution by using simple filters whenever possible. The operators “equals,” “contains,” and “starts with” minimize the possibility of a timeout.

5. Confirm the accuracy of your SOQL statement and Save. Ensure that the Master Object ID (typically, “Id”) is included as a field in the SELECT clause, that your filter criteria in the WHERE clause are correct, and so forth.

## Build a Report for the Master Object Source

We need to create a Salesforce report that contains a list of Master Object IDs. Conga Batch will use this report to determine which records are included in the batch operation. The Salesforce ID of this report will be referenced in the Report Id field under Master Object Source on the Conga Batch record.

### To build a report for the master object source:

1. Create a new Report folder called Conga Batch Reports. This folder is used in a later step to house the Master Object Source Report
2. Clear the High Assurance Session Required setting to allow and Conga Batch to access exported report data. Do this from Setup → Customize → Reports and Dashboard Settings → Access Policies
 

Note: If this setting is enabled, the session security level required to export reports is increased, preventing and Conga Batch access to the report data.
3. Ensure that the Conga Batch Reports folder is accessible by the users who will run the Conga Batch solution.
4. Choose the Salesforce Report Type that matches your master object. The Report Type you select when building your Master Object Source Report must include the same object type as the Master Object specified in the solution you converted to Conga Batch (see [Convert a Button to a Formula Field Using the Conga Formula Builder](#) for further details). For example, if you built your solution with the Opportunity as the master object, your Master Object Source Report Type for Conga Batch must also include Opportunities (ex: Opportunities report type, Opportunities with Contact Roles Report Type, Opportunities with Products Report Type, etc).
5. Include the Master Object ID as the first (far left) column on the report.
6. This is the only field that is required for Conga Batch to read the report and determine the records included in the batch.

7. You can include other fields as columns on the report to assist in your reading of the report (ex: a name field) but the first ID column encountered (as read left-to-right) must be the column of Master Object IDs (see image below). Keep in mind that the fewer fields included on the report, the faster it runs.
8. Define the filter criteria of the report so that the proper records are included in the batch (for example, Time Frame or Opportunity Status).
9. Optimize the processing time of your report and the overall speed of the Conga Batch solution by using simple report filters whenever possible. The operators “equals,” “contains,” and “starts with” minimize the possibility of a timeout.
10. Save the report in the Conga Batch Reports folder.
11. We recommend naming the report as “Conga Batch - [Name of Conga Batch Record].” Using our previous example, our report name is “Conga Batch - Batch Simple Proposal - Oppty.”

## Launch Conga Batch

Launching Conga Batch runs your solution (designated by the Conga Batch formula field referenced in your Conga Batch solution) for the records included in your batch.

You can launch Conga Batch using either of these methods:

### Launch Conga Batch from a Conga Batch Record.

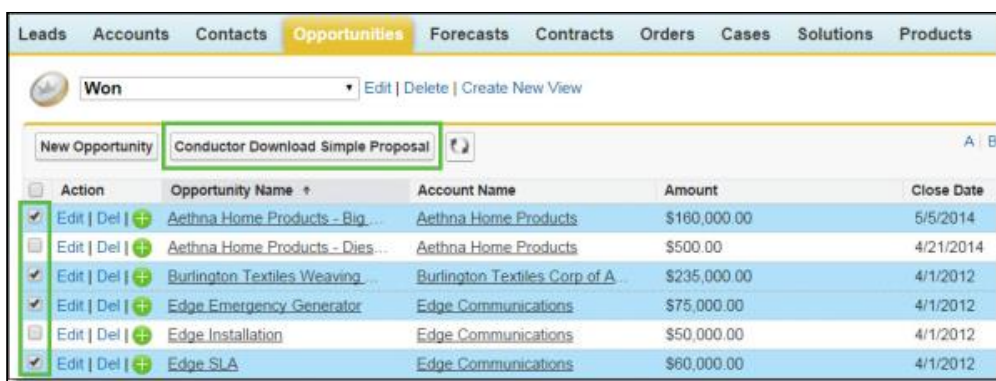
If you use the Conga Batch record, click the Launch button on the Conga Batch record to run the batch of records included in the specified Report or query.



This Launch button uses Salesforce OAuth (open authorization session identification) instead of a Salesforce Session ID. If you cannot use OAuth, edit the Conga Batch page layout to remove this Launch button and replace it with the other Launch button (also called “Launch”) which uses a Session ID instead.

## Launch Conga Batch from a List View Button

If you use a Conga Batch List View button, use the checkboxes in the List View to select the records for your batch on an ad hoc basis and then click your Conga Batch List View button.



Action	Opportunity Name	Account Name	Amount	Close Date
<input checked="" type="checkbox"/> Edit   Del   +	Aethna Home Products - Big...	Aethna Home Products	\$160,000.00	5/5/2014
<input checked="" type="checkbox"/> Edit   Del   +	Aethna Home Products - Dies...	Aethna Home Products	\$500.00	4/21/2014
<input checked="" type="checkbox"/> Edit   Del   +	Burlington Textiles Weaving...	Burlington Textiles Corp of A...	\$235,000.00	4/1/2012
<input checked="" type="checkbox"/> Edit   Del   +	Edge Emergency Generator	Edge Communications	\$75,000.00	4/1/2012
<input checked="" type="checkbox"/> Edit   Del   +	Edge Installation	Edge Communications	\$50,000.00	4/1/2012
<input checked="" type="checkbox"/> Edit   Del   +	Edge SLA	Edge Communications	\$60,000.00	4/1/2012

The Conga Batch dialog box appears, displaying the details of the Conga Batch solution.



**Conga conductor**

Download Invoice Oppty

Use Conga Conductor to generate a batch of Conga Composer files. Click **Start** to begin.

**Selected:** 18 Opportunities

**API Usage:** 318 in the last 24 hours (15000 max)

**Output Mode:** Download

**Consolidate:**  Consolidate PDFs into a single file

**Conductor Status:**

Press Start

Start Close

Copyright © 2009-2015 AppExtremes, Inc - dba Conga. All rights reserved. 8.8.98 USW1B-20CF65EB

**Selected:** Lists how many master object records are included in the batch you're about to run.

**API Usage:** Informs you of how many API requests have been used in your Salesforce instance. Think of this as a for your information or heads up – Conga Batch informs you of this because the nature of running a batch of records with Conga Batch will consume more API requests than running one record at a time (as is the case with).

**Output Mode:** Describes how the files generated from the batch process will be delivered. The output mode is determined by the QMode parameter.

**Consolidate:** If the merged output files are configured to be output as PDF and the Output Mode is set to download the output files, you can enable this checkbox to consolidate the merged output PDF files into one single PDF file (otherwise, the merged output files are in a .zip file when the Output Mode is download).

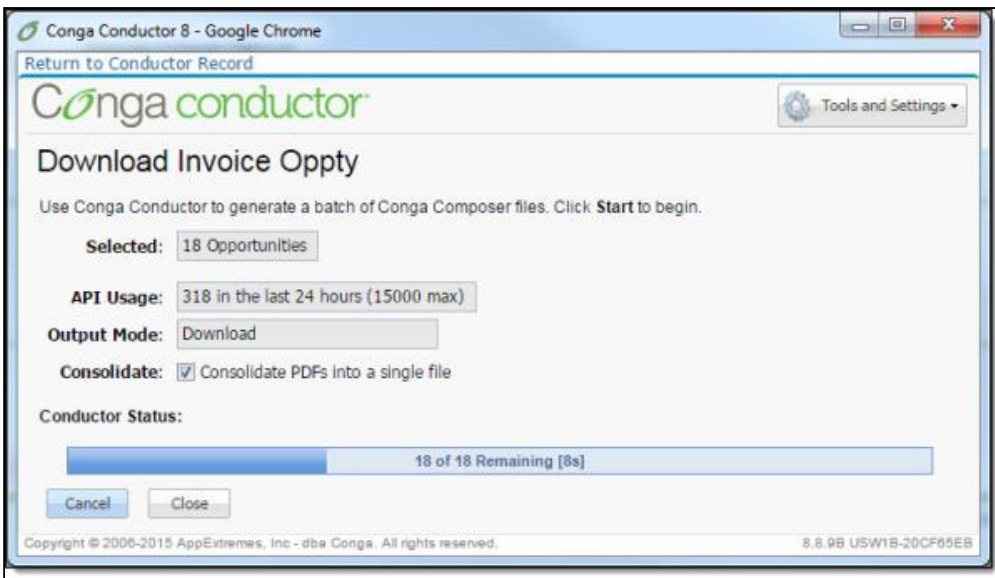
**Conga Batch Status:** Displays the progress of the batch (after the Start button is clicked).

**Start:** Begins the Conga Batch batch process.

**Close:** Closes the Conga Batch dialog box.

**Tools & Settings:** This drop-down allows you to navigate to Conga Batch Setup or will provide a Debug URL (only necessary when troubleshooting).

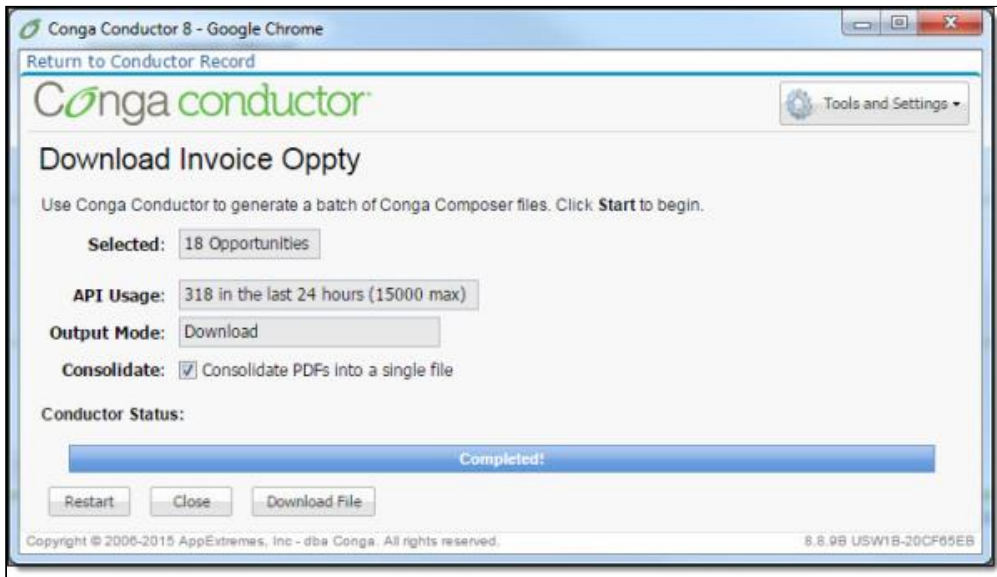
Click Start to run the batch. The progress bar displays the current status of your batch.



If you have configured Conga Batch to download your files, click Download File to download your .zip or PDF file. The PDF is generated if you have enabled the Consolidate option before starting the batch; otherwise a .zip file results.

Other output mode configurations send email or eSignature or attach the merged output files.





A few more details about launching Conga Batch:

- Depending on the number of requests already queued for processing, it may take several minutes after clicking the Start button before processing begins on your requests.
- If errors occur, you are redirected to the Conga Batch Dashboard to view details of the errors.
- If you are planning to download the merged output files and they're all PDFs, you may consolidate each individual PDF into one single PDF file by enabling the checkbox value for the field, Consolidated PDF Output File, on the Conga Batch record (requires version 8.13 or above).

## Schedule Conga Batch to Run Automatically

The final touch in your Conga Batch solution may be to schedule the Conga Batch events to execute automatically, on a schedule. Scheduling Conga Batch means that a user does not have to manually launch the batch but rather Conga Batch looks for batches that are queued to run on a scheduled basis and runs them for you.

The ability to run a batch of records on a scheduled basis without user oversight or intervention potentially poses a great risk. For example, once scheduled, you could inadvertently send hundreds of documents, communications, and/or eSignature requests to external recipients. [Thoroughly test your solution](#) and confirm it's working correctly before scheduling it to run automatically.

- ⚠ You must utilize the Conga Batch record (not a List View button) in order to run the Conga Batch solution on a scheduled basis. As a best practice, it is recommended to schedule your Conga Batch solution to run at least 15 minutes apart. Scheduling multiple Conga Batch solutions at the same time can have unexpected results.

## To schedule Conga Batch to run automatically

1. Click the Conga Batch Setup tab.
2. Enable the Scheduled Conga Batch Master Switch and click Save Changes.
3. Click Create Salesforce Token.

- ⚠ The user who clicks the Create Salesforce Token button is under whose authority Scheduled Conga Batch will operate. The Salesforce Token for Conga Batch will be listed in Salesforce under that user's My Settings → Personal → Advanced User Details → OAuth Connected Apps. If the user revokes the Salesforce Refresh Token (in Salesforce Setup, under OAuth Connected Apps), all scheduled Conga Batch records will cease to operate on a scheduled basis.

4. Open the Conga Batch record that you wish to schedule and click Schedule.
5. Select a Frequency of Daily, Weekly, or Monthly. The Repeat section changes depending on the Frequency you choose. The Status message at the bottom of the screen provides helpful hints and updates as you navigate through the steps of the Scheduler.
6. Choose the Start Date, End Date and time of day (Preferred Start Time) to run the scheduled batch merge. The date displays in U.S. formatting (month/day/year). Note that the time zone displayed is based on the running user's time zone.
7. Click Save Changes.
8. Click the Enable button.
  - The Disable button will deactivate the schedule while the Unschedule button will delete the schedule information.
  - The Schedule section on the Conga Batch record displays the details of the schedule and provides a visual reminder to users that this Conga Batch record runs on a schedule. Your Conga Batch record will now run on an automatic, scheduled basis.
  - The Content Workspace ID field is only used when a Scheduled Conga Batch operation is set to deliver the output files by download (&QMode=Download). In this case, you must populate the ID of a Salesforce Content Library (formerly

- called Content Workspace) so that the files have somewhere to be stored since the merge operation runs in the background.
- Errors that occur during a scheduled event are logged as child records to the Conga Batch Record in the Scheduled Conga Batch History related list; the errors also appear on the Conga Batch Dashboard.

## Conga Batch Product Guide

### Learn Conga Batch Basics

- [About Conga Batch](#)
- [Before You Begin](#)
- [What's in the Conga Batch Package?](#)

### Install Conga Batch

- [Install Conga Batch](#)

### Configure Conga Batch

- [Choose How to Launch Conga Batch](#)
- [Launching from the Conga Batch Record](#)
- [Launching from a List View Button](#)
- [Determine Which Records to Include in Your Batch](#)
- [Create a Conga Batch Record](#)
- [Convert a Button to a Formula Field Using the Conga Formula Builder](#)
- [Create a List View Button](#)
- [Delete the Conga Batch Record](#)
- [Set an API Floor](#)
- [Determine How Conga Batch Will Deliver the Merged Output Files](#)
- [How to Manually Convert a Button into a Conga Batch Formula Field](#)

### Use Conga Batch

- [Launch Conga Batch](#)
- [Launch Conga Batch from a Button on an Object Page Layout](#)

## Automate Conga Batch

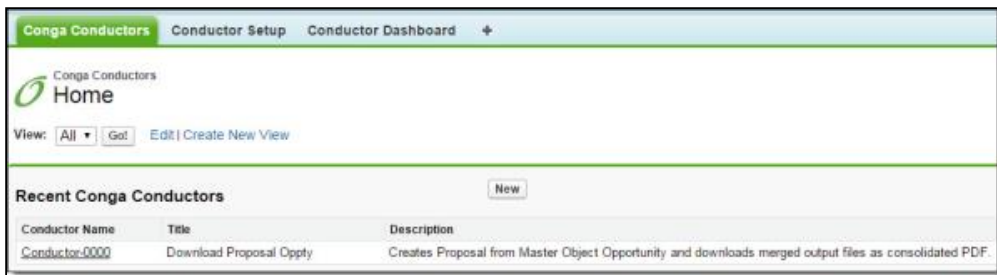
- [Automate a Solution with Conga Batch](#)
- [Schedule Conga Batch to Run Automatically](#)

## Troubleshoot Conga Batch Solutions

- [Conga Batch Troubleshooting Tips](#)
- [Troubleshooting Conga Batch using Hyperlink Formula Fields](#)
- [Test Using the Conga Batch Record](#)
- [Test Using a Conga Batch List View Button](#)
- [Scheduled Conga Batch Failed to Run](#)

## The Conga Batches Tab

The Conga Batches tab houses the records for the Conga Batch custom object, from which a Conga Batch batch process is configured and launched. Please see [The Conga Batch Setup Tab](#) for configuration details.



## The Conga Batch Dashboard Tab

The Conga Batch Dashboard tab monitors Conga Batch batches and lists errors for any records that failed to process.

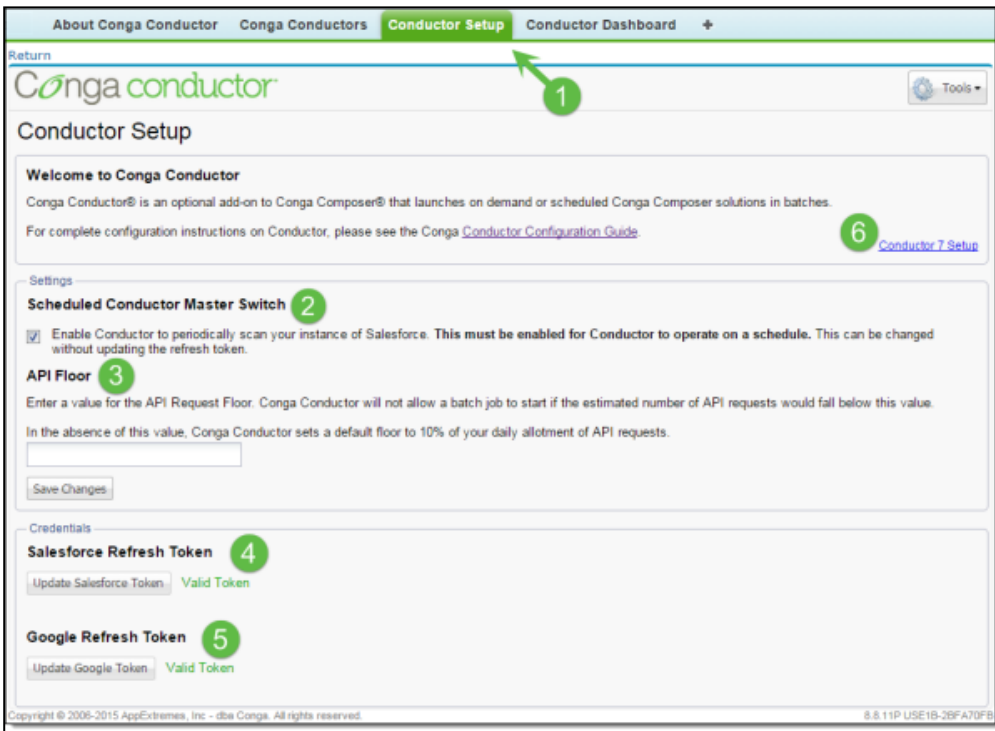
The screenshot shows the Conga Conductor Dashboard interface. At the top, there are navigation tabs: 'About Conga Conductor', 'Conga Conductors', 'Conductor Setup', and 'Conductor Dashboard' (highlighted with a green arrow and a circled '1'). Below the tabs, the dashboard displays the Conga Conductor logo and the title 'Conductor Dashboard'. On the left, there are three input fields: 'Org ID' (with value '000j00000001nwrEAC'), 'API Floor' (with value '0'), and 'Expires' (with value '01-Dec-2016'). On the right, there is a green progress bar labeled 'API Usage Status' with the text 'Good' and '1126 API requests used out of 15000 available'. Below these are two tables: 'Conductor Log' and 'Error List'. The 'Conductor Log' table has columns for Conductor Id, User Id, Success, Failed, Cancelled, Status, Duration, and Log. The 'Error List' table has columns for Record Id and Error Text. Both tables show multiple rows of data. At the bottom, there are pagination controls for both tables, showing 'Page 1 of 1' and 'Displaying records 1 - 9 of 9'.

1	Click the Conga Batch Dashboard tab.
2	Org ID displays the Salesforce Organization ID.
3	API Floor is the number of daily API requests at which, if reached, Conga Batch will cease to operate. Please see the topic on the Conga Batch Setup tab for details on setting an API floor.
4	Expires displays the Conga Batch subscription expiration date.
5	Conga Batch Log displays details of the Conga Batch batches run in the last 24 hours. The Status column displays whether the batch was successfully completed or failed. Click a failed batch to view the error details in the Conga Batch Errors pane.
6	API Usage Status displays the number of daily API requests that have been consumed as well as the total number of API requests allowed per 24 hour period.

7	Error List displays details of records that have failed within Conga Batch batches run in the last 24 hours. The Record ID column displays the Master Object ID of the record within a Conga Batch batch that failed. The Error Text column displays the error message, providing details of why that particular record failed.
---	---

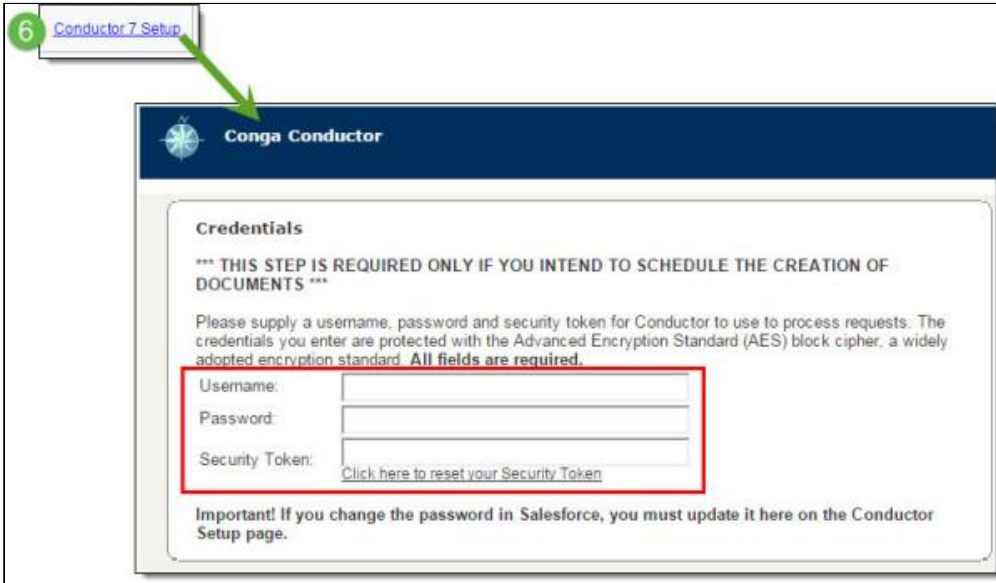
## The Conga Batch Setup Tab

The Conga Batch Setup tab provides tools to administer Conga Batch.



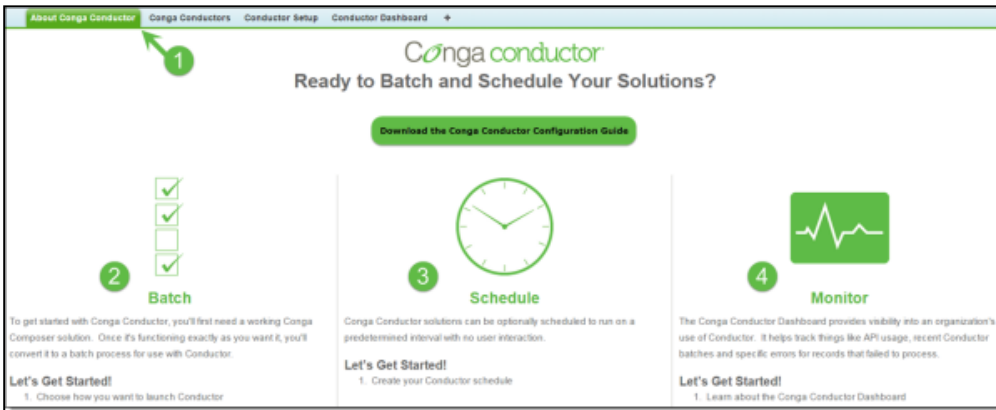
1	Click the Conga Batch Setup tab.
2	Enable the Scheduled Conga Batch Master Switch to configure Conga Batch records to run on a scheduled, recurring basis.

3	<p>API Floor is optional and acts as a governor for Conga Batch. Enter a number in the API Floor field and Conga Batch will not run if the number of available API requests for the Salesforce instance reaches the number specified.</p> <p>This is particularly helpful when API requests need to be reserved for Salesforce Data Loader operations, data warehousing services, etc.</p> <p>If an API Floor value is not entered here, Conga Batch defaults to an API floor of 10% of the daily API request allotment.</p>
4	<p>Create a Salesforce Refresh Token (or update an existing one) in order to run Conga Batch on a scheduled basis using a Salesforce OAuth token.</p>
5	<p>Create a Google Refresh Token in order to integrate Conga Batch with Google Drive.</p>
6	<p>The Conga Batch 7 Setup link provides access to Conga Batch Setup in Release 7 (the older version of Conga Batch).</p> <p>This link is only used for customers who have upgraded from Conga Batch Release 7 to Release 8 and are maintaining some scheduled Conga Batch records on Release 7, in which case they will need to access Conga Batch Setup for Release 7 to properly maintain the Username, Password, and Salesforce Security Token information.</p> <p>The credentials entered are under whose authority scheduled Conga Batch Release 7 records will run. Remember to save the changes.</p> <p>The Username, Password, and Security Token in Conga Batch Release 7 have been replaced with the Salesforce Refresh Token (see #4 above) for scheduled Conga Batch solutions in Release 8.</p>



## The About Conga Batch Tab

The About Conga Batch tab describes Conga Batch capabilities and provides information on how to begin the configuration process.



<p>1</p>	<p>Click the About Conga Batch tab</p>
<p>2</p>	<p>The Batch section is the starting point of Conga Batch and explains how to deploy an existing solution as a Conga Batch batched solution.</p>
<p>3</p>	<p>The Schedule area provides an overview of how to run Conga Batch solutions on a recurring scheduled basis.</p>



4

The Monitor section describes the Conga Batch Dashboard which lists API request usage and errors for any records that failed to process in Conga Batch.

## Building Conga Batch Solutions

Information covering the core functionality of Conga Batch, including using, launching, and scheduling Conga Batch.

Select one of the following topics for more information:

- [Using Conga Batch](#)
- [Launching Conga Batch](#)
- [Scheduling Conga Batch](#)
- [Conga Batch Tips and Tricks](#)

## Using Conga Batch

Information about configuring Batch, converting button URLs to formula fields, and determining how to launch Batch.

- [Automate a Solution with Conga Batch](#)
- [Convert a Composer Button into a Formula Field Manually for Conga Batch](#)
- [Determine How Conga Batch Will Deliver the Merged Output Files](#)
- [Update an Existing Conga Batch Formula Field Using the Formula Builder](#)
- [Conga Batch Configuration Steps](#)
- [Delete a Conga Batch Record](#)
- [Send Output to Content for Conga Batch](#)

## Automate a Solution with Conga Batch

How to configure Conga Batch to automate a solution via Conga Batch Record or a Salesforce List View Button.

The steps involved in configuring Conga Batch are listed below.

The first step in setting up a Conga Batch solution is to determine how you want to launch Conga Batch.


Choose how to launch Conga Batch (via the Conga Batch record or a Salesforce list view).

## To Launch via a Conga Batch Record


1. Determine which records to include in your batch by building a Salesforce report or SOQL query.
2. Create a Conga Batch record.
3. Convert a button or link to a Salesforce formula field using the Conga Formula Builder.
4. Test your Conga Batch solution to ensure everything is working properly.
5. Launch the Conga Batch solution from the Conga Batch record.
6. Optionally, schedule the Conga Batch solution to run automatically at a designated and recurring date and time.

## To Launch via a Salesforce List View Button

1. Convert a button or link to a Salesforce formula field using the Conga Formula Builder.
2. Create a List View button.
3. Delete the Conga Batch record.

 Because you are launching Conga Batch from a List View button, you only needed the Conga Batch record to access the Conga Formula Builder.

4. Test your Conga Batch solution to ensure everything is working properly.
5. Launch the Conga Batch solution from the List View button.

 Conga Batch List View buttons utilize Javascript, which is not supported in Salesforce Lightning.

## Convert a Composer Button into a Formula Field Manually for Conga Batch

### To manually convert a button into a formula field

1. Create a custom formula field on your Master Object with Return Type of Text and label it. It is helpful to include the type of delivery method (ex: download, email, etc.), the template, and the master object in the Label such as, "Conga Batch Download Simple Proposal Oppty."

- Copy the button URL from your working button and paste it in your formula field's formula window.

Example:

```
/apex/csfb__Conga_Composer?SolMgr=1
&serverUrl={!API.Partner_Server_URL_370}
&userName={!$User.Username}
&Id={!blng__Invoice__c.Id}
&TemplateId=a0I800000001aT3
&FP0=1
&AC0=1
&AC1=Invoice+for+{!Opportunity.Name}
```

- Remove the following unnecessary lines:

```
/apex/csfb__Conga_Composer?SolMgr=1&serverUrl={!API.Partner_Server_URL_370}
&userName={!$User.Username}
```

You should be left with just the &Id parameter and what follows it. For example:

```
&Id={!blng__Invoice__c.Id}
&TemplateId=a0I800000001aT3
&FP0=1
&AC0=1
&AC1=Invoice+for+{!Opportunity.Name}
```

- Convert the remaining lines into a valid Salesforce formula.

The formula must adhere to the following rules:

- Literal text strings (a.k.a. static text) must be enclosed in quotes (" ... ")
- Literal spaces within literal text strings must be replaced with plus signs (+)
- Merge fields from the button must be replaced with the corresponding field available in the formula. For example, {!Opportunity.Id} would be replaced with Id
- Each element (literal text strings and fields) must be joined with the concatenation operator (+) or (&)

Example:

```
"&id="+ Id +"&TemplateId=a0I80000000laT3" +"&FP0=1" +"&AC0=1" +"&AC1=Invoice+for+" + Name +
```

- Append the QMode parameter to indicate how Conga Batch should deliver the merged output files.

Example:

```
+ "&QMode=Download"
```

Button URL	Formula
<pre>/apex/csfb__Conga_Composer?SolMgr=1 &amp;serverUrl={!API.Partner_Server_URL_370} &amp;userName={!\$User.Username} &amp;Id={!bng__Invoice__c.Id} &amp;TemplateId=a0I80000000laT3 &amp;FP0=1 &amp;AC0=1 &amp;AC1=Invoice+for+{!Opportunity.Name}</pre>	<pre>"&amp;id=" + Id + "&amp;TemplateId=a0I80000000laT3" + "&amp;FP0=1" + "&amp;AC0=1" + "&amp;AC1=Invoice+for+" + Name + "&amp;QMode=Download"</pre>

## Determine How Conga Batch Will Deliver the Merged Output Files

With Conga Composer, a user can determine how the merged output file will be delivered or where it will be stored after launching the button; however, Conga Batch runs in the background - without user input after the initial launch process - so we must specify in advance how Conga Batch will deliver the output files.

We do this using a QMode parameter in the Conga Batch formula field. The parameter name remains constant (&QMode=) but the value changes depending on the desired delivery method (e.g. "Download" for downloading, "SendEmail" for emailing). Only one QMode parameter may be used per Conga Batch solution, meaning all of the records within that Conga Batch batch will be delivered in the same way.

The QMode table is also helpful for ensuring that your Conga Batch solution includes the required parameters to correctly deliver the merged output files. If you're unfamiliar with these parameters, please see [How Parameters are Formatted](#).

The below table lists the available QMode parameter values and describes how each will deliver the merged output files for a Conga Batch routine using that QMode value.

**Category:** Automation-Background Mode Parameter

**Compatible with:** Composer 8 and Composer 7

**Parameter Name:** QMode

**Parameter Value:** [value]

**Parameter Requirements:** None

**Parameter Description:** Parameters for Conga Batch and Conga Trigger. The QMode parameter is required when working with Conga Batch or Conga Trigger solutions.

If you use the QMode parameter to distribute documents (such as email or eSignature) and you also want to save copies of the document, ensure the SC1 parameter is used in conjunction with QMode.

**Example:**

```
&QMode=Attachments
```

#### QMode Parameter Values for Release 8

QMode Value	Description
&QMode=Attachments	Attaches the merged output files to the Master Object records
&QMode=Download	<p>Downloads the merged output files to the internet browser.</p> <p>This parameter value is available only to Conga Batch. There is no download option available in Conga Trigger.</p> <p>Recommended: &amp;ContentWorkspaceId=[Workspace ID] to specify the Content Workspace (Library) where Conga Batch will store the merged output file if the Conga Batch solution is configured to run on a schedule. Without this parameter, the merged output file from a scheduled Conga Batch solution configured to download will be stored in the running user's personal Salesforce Content Workspace (Library).</p> <p>If either the &amp;DefaultPDF=1 or &amp;FP0=1 parameter is not included in the Conga Batch formula field, the merged output files will be consolidated into a .zip file. However, if one of these PDF output parameters is included in the formula field, the merged output file will be a consolidated PDF.</p>

QMode Value	Description
&QMode=SendEmail	<p>Emails the output file.</p> <p>Requires:</p> <p>&amp;TemplateID unless using only a Conga Email Template (&amp;CongaEmailTemplateID) and no document template.</p> <p>&amp;EmailTemplateId=[Salesforce Email Template ID] or &amp;CongaEmailTemplateID=[Conga Email Template ID]</p> <p>&amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</p> <p>If you cannot use the &amp;EmailToId or &amp;ContactId parameters to specify an email recipient by their Contact Id, you can instead specify &amp;EmailAdditionalTo=[email address] and &amp;EmailSubject=[subject text].</p>
&QMode=Chatter	<p>Posts the merged output file in the Salesforce Chatter feed.</p> <p>By default, Conga Trigger posts files to the Master Object record feed. To override this behavior and post a file to another record's feed, a Chatter Group or a specific User's feed, use &amp;AttachmentParentID.</p>
&QMode=Content	<p>Attaches the merged output files to the running user's personal Workspace (Library) in Salesforce Content.</p> <p>Recommended: &amp;ContentWorkspaceId=[Salesforce Content Workspace ID] to specify an alternate Content Workspace (Library).</p> <p>Does not permit saving additional attachments outside of the initial document created in your Content Workspace (for example, cannot save a copy to Attachments or Chatter in addition to your Content Workspace).</p> <p>Use this method if you are scheduling a content deliver.</p>

QMode Value	Description
&QMode=Documents	<p>Attaches the merged output files to a folder on the Documents tab.</p> <p>Requires: &amp;DocFolderId=[Salesforce Documents Folder ID]</p> <p>Does not permit saving additional attachments outside of the initial document created in your Document folder (i.e. cannot save a copy to Attachments or Chatter in addition to your Document folder).</p>
&QMode=DocuSign	<p>Sends the merged output files via DocuSign (one DocuSign Envelope per Master Object record). Requires:</p> <ul style="list-style-type: none"> <li>• &amp;DocuSignEndPoint= [See <a href="#">DocuSignEndpoint</a> for relevant values]</li> <li>• &amp;DocuSignR1ID=[Salesforce Contact, Lead, or User ID]</li> <li>• &amp;DocuSignR1Role= [See <a href="#">DocuSignR1Role</a> for relevant values]</li> <li>• &amp;DocuSignR1Type= [See <a href="#">DocuSignR1Type</a> for relevant values]</li> <li>• DocuSign Credentials on the USER record</li> <li>• The user's email address on their User record in Salesforce must match the user's email address in DocuSign.</li> </ul> <p>There are many combinations of possible DocuSign parameters. Please see <a href="#">DocuSign Integration Parameters</a> for more information.</p>
&QMode=EchoSign	<p>Sends the merged output files via Adobe eSign (one Adobe eSign Agreement per Master Object record). For more information, see <a href="#">Adobe eSign Integration Parameters</a>.</p>
&QMode=EchoSignCreate	<p>Merges the output files and attaches each to an Adobe eSign Agreement but does not automatically send those Agreements.</p>

QMode Value	Description
&QMode=GoogleDrive	<p>Saves the merged output files in Google Drive.</p> <p>Requires the Google Refresh Token on Conga Batch Setup tab.</p> <p>Use &amp;OFP to specify in which folder the merged output files are to be saved.</p> <p>Remove &amp;SC0=1 and &amp;SC1=Google from the Conga Batch formula field to prevent the storage of duplicate copies of the merged output file in Google Drive.</p>
&QMode=GoogleDocs	<p>Converts the merged output files to Google Documents, Spreadsheets, or Slides, which are saved in Google Drive.</p> <p>Requires Google Refresh Token on Conga Trigger Setup tab.</p> <p>Recommended: &amp;OFP to specify in which folder the merged output files are to be saved.</p>
&QMode=SalesforceFile	<p>Sends the merged output files to Salesforce Files for the Master Object.</p> <p>The &amp;SC0=1 and &amp;SC1=SalesforceFile parameters are required in the URL for this QMode.</p>
&QMode=Sertifi	<p>Sends the merged output files via Sertifi. The browser page refreshes and lands on the Sertifi EContract. There are many combinations of possible Sertifi parameters. For more information, see Sertifi Integration Parameters.</p>
&QMode=SpringCM	<p>Saves the merged output files in a SpringCM folder, as defined by the SpringCM configuration rules.</p>
&QMode=Box	<p>Sends the output documents to Box.</p>
&QMode=OneDrive	<p>Sends the output documents to the user's OneDrive repository.</p>
&QMode=CongaSign	<p>Sends the output documents to Conga Sign.</p>
&QMode=Contracts	<p>Sends the output documents for negotiation using Conga Contracts for Salesforce's Send for Negotiation email service.</p>



QMode Value	Description
&QMode=AmazonS3	Sends the output documents to AmazonS3
&QMode=LinkService	<p>Sends the output documents with Composer Link Service.</p> <p><b>Requires:</b></p> <ul style="list-style-type: none"> <li>• Composer File Storage enabled in Composer Setup.</li> <li>• &amp;CongaEmailTemplateID to designate a specific Conga Email Template</li> <li>• The {Conga_Doc_Link} merge tag to successfully generate the document link in the email.</li> <li>• &amp;TemplateID unless using only a Conga Email Template and no document template.</li> <li>• &amp;EmailToId=[Salesforce Contact ID] or &amp;ContactId=[Salesforce Contact Id]</li> </ul> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>✔ Use the NETId parameter to send Notification Emails with Composer Link Service.</p> </div>
&QMode=EmailLink	Allows you to send the composer-generated link via a unique URL in a Conga Email Template with Trigger or Batch.
&QMode=GenerateLink	Allows you to generate a unique URL for the Composer generated document and adds that link to the document history record.

### QMode parameter values for Release 7

Value	Definition
1	<p>Download the output file to your computer.</p> <p>(One of the following parameters should be used with QMode=1: LG3=1 OR LG4=1.) (default)</p>
2	<p>Attaches the merged output to the master object.</p> <p>LG4=1 is the only Activity Logging parameter needed when using QMode=2.</p> <p>Analogous to Qmode=Attachments</p>

Value	Definition
3	<p>Emails the output file.</p> <p>Requires &amp;Templateld</p> <p>For an Opportunity with a Primary Contact (Contact Roles), there is no need to set the &amp;EmailTold. Also, if you use &amp;EmailTemplateld, you must specify the &amp;EmailTold.</p>
4	<p>Delivers the output via Adobe eSign. Requires Adobe eSign parameters.</p>
5	<p>Attaches the output file to a folder on the Documents tab.</p> <p>QMode=5 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
6	<p>Attaches the output file to a workspace in Salesforce Content.</p> <p>QMode=6 saves the output to the specified location only. You cannot attach the document to other locations such as Chatter or Notes and Attachments.</p>
7	<p>Delivers the output via DocuSign. Requires DocuSign parameters.</p>
94	<p>Creates an Adobe eSign Agreement but does not automatically deliver it. Requires Adobe eSign parameters.</p>

## Update an Existing Conga Batch Formula Field Using the Formula Builder

The primary purpose of the Conga Formula Builder is to convert an existing button or link to a Salesforce formula field. However, the Formula Builder can also be used to update an existing Conga Batch formula field.

### To update an existing Conga Batch formula field using the Formula Builder

1. Navigate to the Conga Batch record that lists the formula field you wish to update (in URL Field Name).

⚠ The Conga Batch record must also have a value for either Report ID, Query ID, or Record ID.

2. Click the Formula Builder tab. Notice the message that the button has been automatically loaded since a formula field and Report/Query/Record ID were already entered on the Conga Batch record. The button or link that was converted is in the left pane while the formula field it was converted to is in the right pane.

✔ Disable the Show Converted Formula checkbox to view the button in URL format.

3. Click Edit Button. The button opens in edit mode in a new browser tab.
4. Edit the button and click Save.
5. Return to the Formula Builder and in the Button / Link Name drop-down menu, select any other button, and then select the original button you edited above. This is similar to a browser refresh (alerting Formula Builder that changes were made in the button) but allows the Formula Builder to remember your place in the update process.
6. Formula Builder notifies you that the button and formula field don't currently match – click Update Field.
7. Click OK on the dialog box confirming the formula field has been updated.

The button and the formula field now match exactly and the updates are complete.

## Conga Batch Configuration Steps

First, [choose how to launch Conga Batch](#). You can launch it using a Conga Batch record or a Salesforce list view. The steps for configuring Conga Batch vary for each method.

### Launching via a Conga Batch Record


1. [Determine which records to include](#) in your batch by building a Salesforce report or SOQL query.
2. [Create a Conga Batch record](#).
3. [Convert a button or link to a Salesforce formula field](#) using the Conga Formula Builder.
4. [Test the Conga Batch solution](#) to ensure everything works properly.
5. [Launch the Conga Batch solution](#) from the Conga Batch record.
6. Optional: [Schedule the Conga Batch solution](#) to run automatically at a designated and recurring date and time.

## Launching via a Salesforce List View Button

1. [Create a Conga Batch record](#).
2. [Convert a button or link to a Salesforce formula field](#) using the Conga Formula Builder.
3. [Create a List View button](#).
4. Delete the Conga Batch record. You only need the Conga Batch record temporarily to access the Conga Formula Builder.
5. [Test the Conga Batch solution](#) to ensure everything is working properly.
6. [Launch the Conga Batch solution](#) from the List View button.


## Delete a Conga Batch Record

For Conga Batch solutions that launch from a Salesforce List View button, the Conga Batch record is only needed in order to access the Conga Formula Builder. Therefore, once you create the List View button, you can delete the Conga Batch record so it does not cause confusion later or take up unnecessary space.

 Conga Batch solutions launched from a Conga Batch record instead of a List View button should ignore this section.

### To delete the Conga Batch record

1. Click the Conga Batches tab.
2. Locate the correct Conga Batch record from Recent Conga Batches. Click its Conga Batch Name.
3. You may need to click the Go button next to the View drop-down list to see the correct record in the list.

 Since you were only using this record as a placeholder to access the Conga Formula Builder, the record should say "to be deleted" in its Title or Description.

4. Confirm that you have selected the correct record, then click Delete.
5. Click OK in the confirmation dialog.

## Send Output to Content for Conga Batch

In Conga Batch 8.13 the user has the option of choosing between a ZIP folder or a consolidated PDF when sending output to Content.

To utilize this feature, the administrator needs to manually add the field, "Consolidated PDF Output File", to the Conga Batch Layout. It's suggested that it's added to the Schedule section since that is what it modifies.

See [Saving a consolidated PDF output to a Content Library using Conga Batch 8](#) for more information.

## Launching Conga Batch

Information about how to launch Conga Batch.

- [Benefits of Launching Conga Batch from a Conga Batch Record](#)
- [Launch Conga Batch from a List View Button](#)
- [Launch Conga Batch from a Button on an Object Page Layout](#)
- [Launch Conga Batch from a Salesforce Record](#)
- [Conga Batch Launch Buttons](#)

## Benefits of Launching Conga Batch from a Conga Batch Record

The Conga Batch record provides a place to house all of the details of your Conga Batch solutions and uses a report, query or Salesforce record to [determine which records to include in your batch](#).

The benefits of using a Conga Batch record include:

- You can use a Salesforce report or a SOQL query to determine the list of master object IDs (which records to include in your batch).
- You can leverage Salesforce security to restrict access to the Conga Batch object or each Conga Batch record.
- You can schedule Conga Batch to run on an automatic, repeated basis.

## Launch Conga Batch from a List View Button

Launching your Conga Batch solution from a Salesforce List View button allows you to choose the records in your batch on an ad hoc basis rather than using a report's or query's filter criteria to determine which records are included.

The benefits of using a List View button include:

- Easily choose which records you want to use as the Master Object records in the Conga Batch batch by selecting them manually from the list.
- Select up to 75 records per Conga Batch batch.

- Optionally, you can also use a Related List Button on an object's page layout.

**i** The maximum amount of records you can select and merge with a Conga Batch List View button is 75 records. Use a scheduled Conga Batch solution to merge more than 75 records at once.

If you choose to launch via a List View button, all users who have access to List Views for that object will see the Conga Batch List View button, though a user must have a license to run a Conga Batch process.

- See [Create a List View Button](#) for more information.
- See [Create a List View button in Lightning](#) for more information on how to make this implementation method work with the Salesforce Lightning Experience.
- Alternatively, you can [launch Conga Batch from a Conga Batch Record](#).

## Launch Conga Batch from a Button on an Object Page Layout

Typically, Conga Batch executes a batch of records determined by either a Salesforce report, a SOQL query, or a Salesforce List View. There is another way to execute Conga Batch - from a button on an object page layout.

**i** Launching Conga Batch from a button on an Object Page layout is an advanced solution provided for advanced users. Support is limited.

Typically, Conga Batch executes a batch of records determined by either a Salesforce report, a SOQL query, or a Salesforce List View. There is another way to execute Conga Batch - from a button on an object page layout.

The benefit of launching Conga Batch in this manner is that the records in the batch are determined by either a Salesforce report or SOQL query, filtered by dynamically passing a value from the Master Object (the object on which the Conga Batch button is placed) to the Report or query. Therefore, the Conga Batch batch is executed against a batch of records that are related to the Master Object. Furthermore, this method provides an easy way for end users to execute a batch of records without having to give them access to the Conga Batch object or a Conga Batch List View button.

To launch Conga Batch from a button on an object page layout:

1. Build a custom button on your desired Master Object.
2. Navigate to Setup → Build → Customize → [select object] → Buttons, Links, and Actions → New Button or Link.

3. Click the New Button or Link button.
4. Enter a Label and select either "Detail Page Button" or "Detail Page Link" as the Display Type. The Behavior is "Display in new window" and the Content Source is "URL."

5. Paste the following URL into the button or link URL field:

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=
```

6. Enter the API name of your Conga Batch formula field as the value for the &URLFieldName parameter.

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
```

7. Ensure your Conga Batch formula field includes, at minimum, the &TemplateID and &QMode parameters with appropriate values. Depending on the QMode selected, additional parameters may be required. If you wish solely to send an email and do not want to generate a document, use &CongaEmailTemplateID (and other required email parameters) in place of &TemplateID.
8. Add the &ReportID or &QueryID parameter to the button URL.

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c&ReportID=
```

Or

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c&QueryID=
```

9. Paste the appropriate report ID or query ID as the value for the parameter entered in the previous step. This report or query determines which records are included in your Conga Batch batch. For example, you may use a report that retrieves the Products (from the Pricebook) associated to your Master Object of Opportunity.

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
&ReportID=000j00000020jiz
```

Or

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
&QueryID=a05j0000001FHxo
```

10. Enter “?pv0=” after the Report ID or query ID value, followed by the value by which you want to filter the Report or query. The “?pv0=” instructs Conga Batch to dynamically (without manual user intervention) pass a value to the specified Report or query.

The value entered after “?pv0=” is passed to the first filter of the Report or query and is some value from the Master Object, typically the record ID.

This method ensures that the records in the Conga Batch batch launched from this button are related to the Master Object record.

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
&ReportID=000j00000020jlz?pv0={!Opportunity.Id}
```

Or

```
https://conductor.congamerger.com
?sessionId={!$Api.Session_ID}
&serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
&QueryID=a05j0000001FHxo?pv0={!Account.Id}
```

In Lightning, ensure Conga Schedule is set up as a connected app, and remove the session ID:

```
https://conductor.congamerger.com
?serverUrl={!$Api.Partner_Server_URL_290}
&URLFieldName=Conga_Conductor_API_Field__c
&QueryID=a05j0000001FHxo?pv0={!Account.Id}
```

11. Add the button to the appropriate Master Object page layout(s).
12. Navigate to Setup > Build > Customize > [select object] > Page Layouts > Edit.
13. Test your solution thoroughly.
14. Launch Conga Batch from a Master Object record.



## Launch Conga Batch from a Salesforce Record

Instead of using a Conga Batch record or a list view button, Conga Batch can also be launched from a custom button on a standard or custom object's page layout.

To launch Conga Batch from a custom button, create a new button on your desired object that starts with the following URL:

- **For Salesforce Classic:** *https://conductor.congamerge.com/conductor8/index.html?sessionId={!\$Api.Session\_ID} &serverUrl={!\$Api.Partner\_Server\_URL\_160}*
- **For Salesforce Lightning:** *https://conductor.congamerge.com/conductor8/index.html?&serverUrl={!\$Api.Partner\_Server\_URL\_330}*

Next, reference your Salesforce Report or SOQL query that defines the scope of records to include in a batch process with Conga Batch. Use only ONE of the following parameters:

- **Option One:** *&ReportId=00Oxxxxxxxxxxxxx?pv0={ParameterToPass}*
- **Option Two:** *&QueryId=a0xxxxxxxxxxxxx?pv0={ParameterToPass}*

Where *{ParameterToPass}* is a merge field that will pass a value to your Report or Query. For example, if a Conga Batch button is on an Account and the Salesforce Report is configured to show all Cases for a given Account, you would pass the *{!Account.Id}*.

Finally, reference the formula field that contains the **Conga Batch**

**URL:** *&URLFieldName=Conductor\_URL\_Field\_\_c*

**A complete example of a Conga Batch button is below:**

- **For Salesforce Classic:**  
*https://conductor.congamerge.com/conductor8/index.html?sessionId={!\$Api.Session\_ID}&serverUrl={!\$Api.Partner\_Server\_URL\_160}&ReportId=00OE0000000HM5a?pv0={!Account.Id}&URLFieldName=Conductor\_URL\_Field\_\_c*
- **For Salesforce Lightning:**  
*https://conductor.congamerge.com/conductor8/index.html?serverUrl={!\$Api.Partner\_Server\_URL\_330}&QueryID=a038c00001dHSnmAAG?pv0={!Account.Id}&URLFieldName=Conductor\_URL\_Field\_\_c*

## Conga Batch Launch Buttons

The Conga Batch custom object uses a Launch button to initiate the batch process. There are four versions of the Launch button but only one is included on the Conga Batch page layout by default.

Conga recommends using this default Launch button as it uses Salesforce OAuth (open authorization) session identification, which is the most secure.

Each version of the Launch button is described below. View details by navigating to Salesforce Setup → Build → Create → Objects → Conga Batch → Buttons, Links, and Actions.

Button Label	API Name	Displays by default on Conga Batch Page Layout	Description
Launch	Launch	Yes	This is the default Launch button and the one Conga recommends using. It launches Conga Batch using Salesforce's OAuth session identification and takes into account the Conga Batch Environment and Version fields.
Launch	Launch_Dev	No	This is an alternate version of the default Launch button. It launches Conga Batch Release 8 using a Salesforce session ID instead of OAuth and does not take into account the Conga Batch Environment or Version fields.
Launch C7	Launch_C7	No	An alternate version of the default Launch button, it initiates Conga Batch Release 7 (the older version of Conga Batch).
C8	Launch_C8	No	An alternate version of the default Launch button. This button is the same as the default Launch button but uses querystring parameters instead of the Visualforce page.

## Scheduling Conga Batch

Information on how to run Conga Batch on a schedule.

- [Conga Batch Schedule Buttons](#)
- [Important Note about Professional Edition](#)
- [Run Conga Batch on a Schedule](#)

## Conga Batch Schedule Buttons

The Conga Batch custom object uses a Schedule button to optionally schedule the batch process to run on a recurring date and time. There are three versions of the Schedule button but only one is included on the Conga Batch page layout by default.

Conga recommends using this default Schedule button as it uses Salesforce OAuth (open authorization) session identification, which is the most secure.

Each version of the Schedule button is described below. View details by navigating to Salesforce Setup > Build > Create > Objects > Conga Batch > Buttons, Links, and Actions.

Button Label	API Name	Displays by default on Conga Batch Page Layout	Description
Schedule	Schedule	Yes	This is the default version of the Schedule button and the one Conga recommends using. It opens the Conga Batch Scheduler and uses OAuth session identification and takes into account the Conga Batch Environment and Version fields.
Schedule C7	Schedule_C7	No	This is an alternate version of the default Schedule button and opens the older version (Release 7) of the Conga Batch Scheduler.
Schedule C8	Schedule_C8	No	This is an alternate version of the default Schedule button. It is the same as the default button but uses querystring parameters instead of the Visualforce page. It takes into account the Conga Batch Environment field, but not the Version field.

## Important Note about Professional Edition

How to configure Conga Batch to run on a scheduled basis in Professional Edition.

The Next Run Date and Schedule Description fields (API field names APXT\_BPM\_\_Next\_Run\_Date\_\_c and APXT\_BPM\_\_Schedule\_Description\_\_c) must be included on the Conga Batch page layout for Professional Edition instances. If these fields are not included on the Conga Batch page layout, Conga Batch will not be able to write to them and therefore the scheduled Conga Batch solution will not process.

### To configure Conga Batch to run on a scheduled basis

1. Click the Conga Batch Setup tab and enable the Scheduled Conga Batch Master Switch.
2. Click Save Changes.
3. In Conga Batch Setup, click Create Salesforce Token.  
Important: Scheduled Conga Batch will run under the authority of whoever is logged in and clicks the Create Salesforce Token button so ensure it is an appropriate user (typically one with the System Administrator profile). Scheduled Conga Batch operations run according to the time zone of the user specified in the Salesforce Refresh Token. Each user is allowed four Salesforce Refresh Tokens per connected app. Therefore, a small risk exists that the token saved for Conga Batch will expire due to an excess number of subsequent tokens. This scenario is very likely if Salesforce credentials are shared amongst many people or if one user utilizes multiple internet browsers to create Refresh Tokens.
4. A dialog box appears requesting permission for Conga Composer to access the Salesforce instance - click Allow. This is Salesforce's OAuth (open authorization) feature and creates the Salesforce refresh token for Conga Batch. While the token is created under the name, the configuration is for Conga Batch. The refresh token allows Conga Batch to access Salesforce (only the Conga Batch components) on this user's behalf in order to run the scheduled Conga Batch batches. This dialog may not appear if the Conga-recommended method of configuring as "Admin approved users are pre-authorized" is utilized (Salesforce Setup → Administer → Manage Apps → Connected Apps →).
5. Click OK on the dialog box confirming that the Salesforce refresh token has been saved. The Salesforce Refresh Token area in Conga Batch Setup now displays as Valid Token.

If the Salesforce refresh token needs to be changed to run under a different user, click the Update Salesforce Refresh Token button. Conga Composer also displays in the Connected

Apps OAuth Usage (Salesforce Setup → Administer → Manage Apps → Connected Apps OAuth Usage).

If an end-user blocks Conga Composer or revokes the token (under Salesforce Setup → Administer → Manage Apps → Connected Apps OAuth Usage), all scheduled Conga Batch records will cease to operate on a scheduled basis.

Now Conga Batch batches may be scheduled via the Schedule button on the Conga Batch record.

## Run Conga Batch on a Schedule

A convenient, optional feature of Conga Batch is the ability to schedule a batch process. Scheduled Conga Batch solutions run without user initiation on a recurring, scheduled basis (such as every Monday at 8 AM).

A user does not have to manually launch the batch; instead Conga Batch “looks for” batches that are queued to run on a scheduled basis and runs them automatically. Scheduling is available only from a Conga Batch record (not a List View button).

The ability to run a batch of records on a scheduled basis without user oversight or intervention potentially poses a great risk. For example, once scheduled, hundreds of documents, communications, and/or eSignature requests could be inadvertently sent to external recipients. Please [thoroughly test the solution](#) and confirm it’s working correctly before scheduling it to run automatically.

For more information on setting Conga Batch to run on a schedule, see [Schedule Conga Batch to Run Automatically](#).

## Conga Batch Tips and Tricks

Tips and tricks for using Conga Batch.

- [Accommodating Large Batch Sizes for a Consolidated output with Conga Batch](#)
- [Automate Global Merge Solution with Conga Batch](#)
- [Caveats and Warnings for Conga Batch](#)
- [Conga Batch Operation verse Global Merge Operation: What's the Difference?](#)
- [Integrate Conga Batch with Google Drive](#)
- [Saving a Consolidated PDF Output to a Content Library Using Conga Batch 8](#)
- [Scheduling a QuickMerge Report with Conga Batch \(Release 7\)](#)
- [Upgrade Conga Batch List View Buttons to Run on Release 8](#)
- [Upgrade Conga Batch Records to Run on Release 8](#)
- [Upgrade Conga Batch to Release 8](#)
- [Upgrading Conga Batch to Release 8](#)

- [Set an API Floor](#)

## Accommodating Large Batch Sizes for a Consolidated output with Conga Batch

Conga Batch limits batches to 200 records for a consolidated (download) output and 1,000 for a distributed (email) output. The steps below show you two options for accommodating larger batch sizes in your solution using MFTS with a report or query.

- ✔ These options do not work for Conga Batch batches launched from List View buttons, only from Conga Batch records.

### MFTS

With this method, the idea is that every time a batch is run, a checkbox, date field, or some other field on those records is checked by the MFTS parameter.

The source report or query set to filter out those records so that the next batch will automatically include the next set of records in the batch (for example, those that meet the source report criteria and do not have the checkbox checked).

To set this process up, follow the steps below:

1. Create a checkbox or date field on the record type you are including in your batch. Ensure permissions are set correctly and that the field is added to the page layout.
2. Add MFTS parameters to the solution. Set the MFTS parameters to update the field when the record has been run in a Conga Batch batch.  
Tip: A checkbox is easier to set up but a date field is easier to visualize and easier to re-use next time you run a batch.
3. Ensure your source report or query is filtering to EXCLUDE records that have the checkbox/date field.
4. Set a Row Limit in the source report or query of 200.
5. Manually Launch Conga Batch and simply wait a few moments between clicking the button to ensure the previous process has finished.

### Queries

This method uses queries to designate what records to include in the batch.

Your query should contain an ORDER BY clause ordering the records by a field that makes sense. It should also contain a LIMIT clause to limit the results to 200 to stay under the 200 record limit with consolidated output in Conga Batch.

Below is an example of a SOQL query that would meet these requirements:

```
SELECT Id FROM Object WHERE Invoice_Sent__c is true ORDER BY  
Some_Appropriate_Field__c LIMIT 200
```

Name your record with the query name, 01, and the time it will run (example: Property Reports - Word - 01 (12:00 am) ) for easy scanning on the Recent Conga Batches list.

1. Copy the 15-digit Salesforce ID of the Conga query record from the URL.
2. [Create a Conga Batch Record](#).
3. Paste the Query ID from step 2 into the Query ID field on the Conga Batch Record.
4. Use Conga Batch to [schedule your first batch](#). You should set your schedule to:
  - Frequency: Monthly
  - Repeat: On a day last of every month.
  - Scheduling: Enter Start Date; End date will populate with 1 year from that date.
  - Timing: Preferred Start Time: 12:00 AM
5. Click **Save Changes**.
6. Click **Enable** to enable the new Conga Batch schedule.

## Automate Global Merge Solution with Conga Batch

As with any other Conga Composer solution, a Conga Global Merge solution can be scheduled to run automatically using [Conga Batch](#), an optional, paid add-on to Conga Composer.

First, [create your Global Merge solution](#) and ensure it works just the way you want. Note the solution must be compatible with [Background Mode](#) (but not necessarily in Background Mode) in order to automate with Conga Batch. This means you must pre-select the merge template(s) as well as the desired delivery method (download, email, etc.).

### To automate a Global Merge Solution with Conga Batch

1. Install Conga Batch (if you have not already).
2. Navigate to the **Conga Batches** tab.
3. Click **New**.
4. Name your solution (we recommend using the same name as your Global Merge solution - i.e. Sales Pipeline Report or Quarterly Business Review).

5. Click **Save**.

The screenshot shows the 'Conga Conductor Edit' interface for 'Conductor-0001'. At the top right, there are three buttons: 'Save', 'Save & New', and 'Cancel'. The 'Save' button is highlighted with a red box. Below the buttons is the 'Information' section with fields for 'Conductor Name' (Conductor-0001), 'Title' (QBR), and 'Description'. The 'Master Object Source' section contains fields for 'Report ID', 'Query ID', 'Record ID', 'URL Field Name', 'Version' (set to 'C8'), and 'Conductor Environment' (set to 'prod-'). The 'Schedule' section has a 'Content Workspace ID' field. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

6. Next, navigate to the Global Merge record you want to automate and copy the record id from the address bar (the last fifteen characters of the URL).

7. Paste the Global Merge record id into the Record ID field on the Conga Batch record you just created and click **Save**.

The screenshot shows the 'Conga Conductor Detail' page for 'Conductor-0001'. It includes a toolbar with buttons for 'Edit', 'Delete', 'Clone', 'View Report', 'View Query', 'Formula Builder', 'Schedule', and 'Launch'. The 'Master Object Source' section is expanded, showing fields for 'Report ID', 'Query ID', 'Record ID' (highlighted with a red box and containing the value 'a073600000219dK'), 'URL Field Name', 'Version' (C8), and 'Conductor Environment' (prod-). The 'Schedule' section has fields for 'Schedule Description' and 'Next Run Date'. At the bottom, there is a 'Scheduled Conductor History' section with the message 'No records to display' and a 'Back To Top' link.

8. Click **Formula Builder**.

This screenshot is identical to the previous one, showing the 'Conga Conductor Detail' page for 'Conductor-0001'. The 'Record ID' field in the 'Master Object Source' section is still highlighted with a red box. In this screenshot, the 'Formula Builder' button in the toolbar is highlighted with a red box.



9. Conga Global Merge will populate as the Master Object. Select the desired Output Mode (this will append the corresponding QMode to the formula. field).

Return to Conductor Record

### Formula Builder

Select Object | Select Output Mode | Select Button | Review Formula | Enter Formula Name | Create Field

Master Object: Conga Global Merge

Output Mode: Download

**Button / Link**

Name:  Show API Name

Formula / URL:  Show Converted Formula

**Formula Field**

Name:  Show API Name

QBR

Formula:

The formula field has not been created.

Edit Button | Create Field | Update Conductor

Copyright © 2008-2015 AppExtremes, Inc - dba Conga. All rights reserved. 8.13.4P USW1C-398B281D

10. Next, select the button from the Name drop-down list (this is the name of the Global Merge solution - see below). The converted formula will display.

Return to Conductor Record

### Formula Builder

Select Object | Select Output Mode | Select Button | Review Formula | Enter Formula Name | Create Field

Master Object: Conga Global Merge

Output Mode: Download

**Button / Link**

Name:  Show API Name

QBR

Create Performance Report

QBR

**Formula Field**

Name:  Show API Name

QBR

Formula:

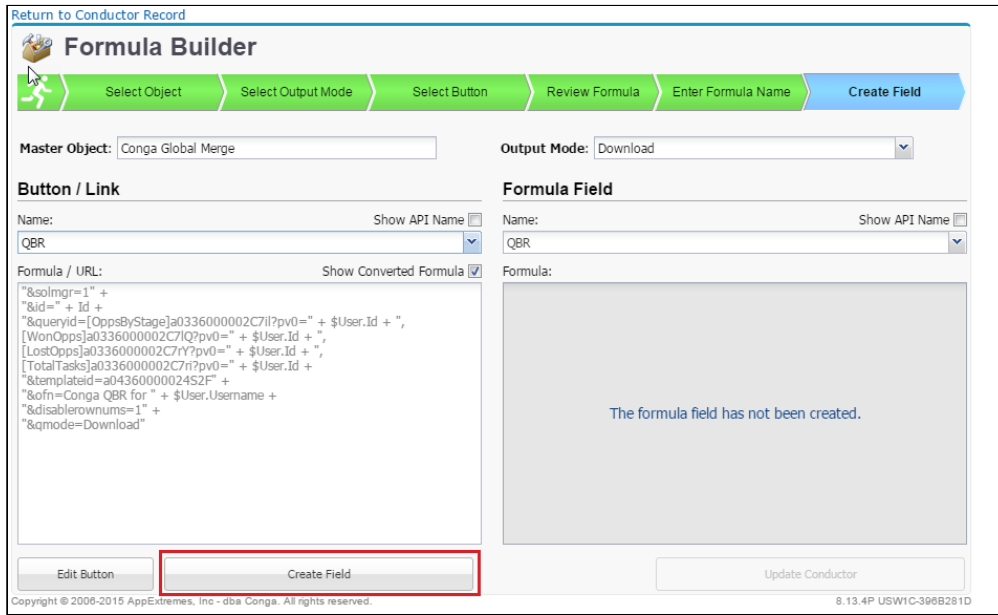
```
&id= + Id +
"&queryid=[OppsByStage]a033600002C7il?pv0=" + $User.Id + ",
[WonOpps]a033600002C7lQ?pv0=" + $User.Id + ",
[LostOpps]a033600002C7r?pv0=" + $User.Id + ",
[TotalTasks]a033600002C7n?pv0=" + $User.Id +
"&templateId=a043600002452F" +
"&ofn=Conga QBR for " + $User.Username +
"&disablerownums=1" +
"&qmode=Download"
```

The formula field has not been created.

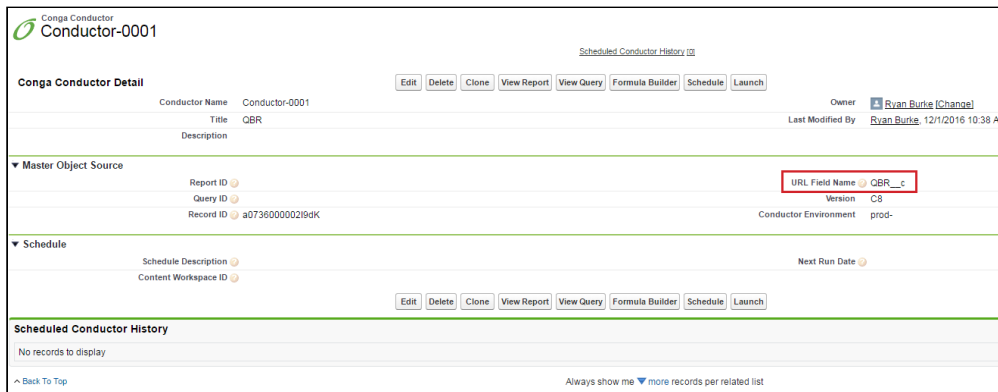
Edit Button | Create Field | Update Conductor

Copyright © 2008-2015 AppExtremes, Inc - dba Conga. All rights reserved. 8.13.4P USW1C-398B281D

11. Click **Create Field** (see below).

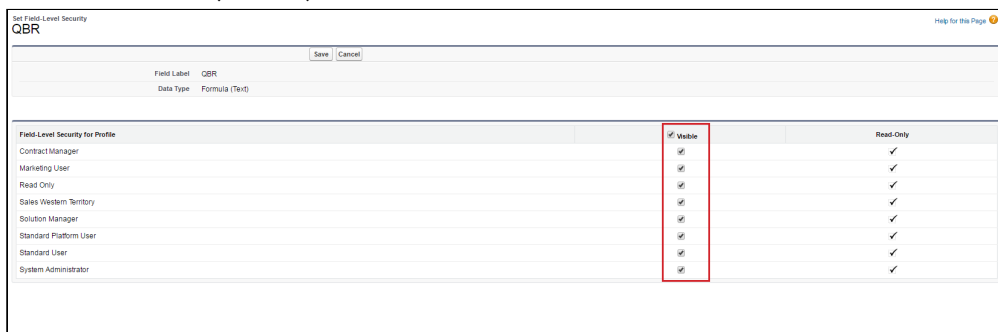


12. A success message will display. Click **Yes** on the success message to update the Conga Batch record with the API Name of the newly created formula field (as in the screenshot below).

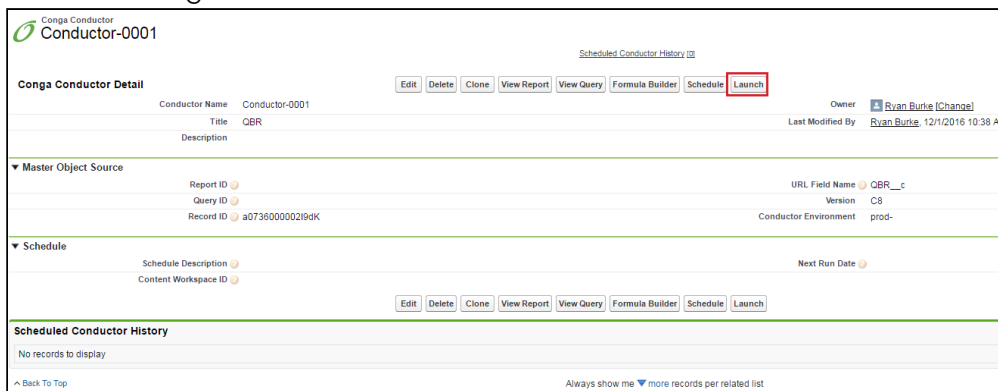


13. Click **OK**.
14. Lastly, ensure the formula field is visible to you and any other user who might run this Conga Batch solution. To do so, return to the Formula Builder and click Edit Field Level Security.

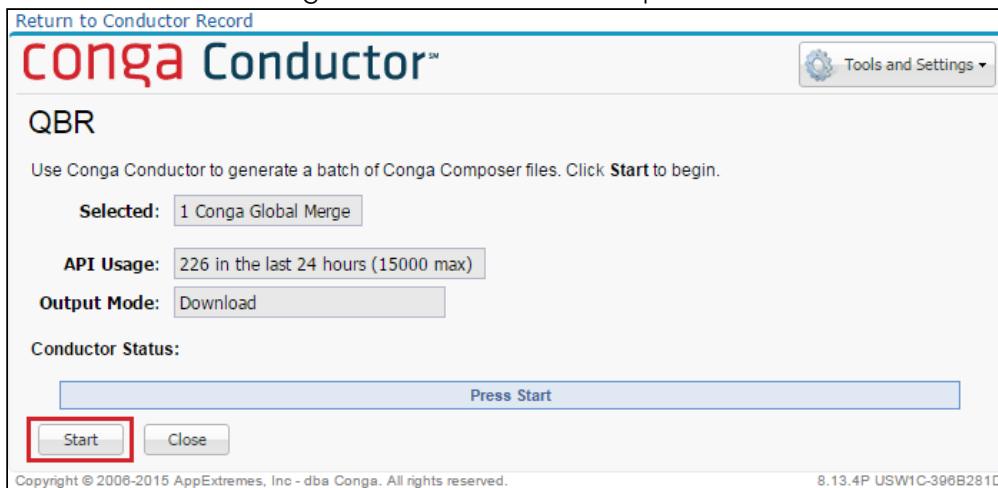
- Check the Visible box next to all of the Profiles of users who need to run the Conga Batch solution (below).



- Click **Save** and close the tab.
- We recommend testing your solution before you schedule it. To do so, click **Launch** from the Conga Batch record.



- Click **Start** in the Conga Batch window that opens.



- Ensure your Global Merge solution runs and delivers the document(s) as expected. You can review the results and any error messages on the Conga Batch Dashboard tab.
- (Optional) The final step is to schedule the Conga Batch solution on whatever recurring intervals you prefer. To do so, follow the steps in [Schedule Conga Batch to Run Automatically](#).

That's it - you have automated a Global Merge solution using Conga Batch.

## Caveats and Warnings for Conga Batch

List of caveats and warnings for Conga Batch.

- Conga Batch is limited to 200 records per batch for consolidated output or when launching from a Salesforce List View button. Up to 1,000 records may be included in the batch for distributed output.
- Each Master Object record in the batch will use one service event. Service events are deducted from your annual subscription plan of service events.
- Each Conga Batch merge operation consumes API requests; the number varies depending on the nature and complexity of the solution as well as how many records are in the batch. Therefore, you'll need to gauge the number of Conga Batch operations that will take place in your instance with the number of API requests you're allowed in a 24-hour period.
- While not generally necessary, set a 12-hour session timeout value when using Conga Trigger or a Scheduled Conga Batch to increase the likelihood that a session is still active when the document processing takes place.
- Ensure that ampersands in text fields passed via Conga Batch have a space before and after the ampersand character. For more information, see Replace Ampersands (&) in Reference Field Data.
- If you have a Conga Batch scheduled that utilizes reports, remain in Classic. If you switch to Lightning, a scheduled Conga Batch will pick this up and not be able to generate data for that report as Conga cannot read report data in Lightning.
- When using Conga products and DocuSign in a fully automated fashion (such as using the background parameter/value pair &DS7=17, Conga Trigger, or Conga Batch) the DocuSign Username and Salesforce User Email must match. If not, you will receive this error: Error: CreateEnvelope in DocuSignAPI. The UserID does not have a valid membership in this Account.

## Conga Batch Operation verse Global Merge Operation: What's the Difference?

Global Merge Formula on Global Merge, started life as Pipeline, Org Wide report
Batch Only, typically created as a Summary report, ONE email

<ul style="list-style-type: none"> <li>• Usually, just Download or Email</li> <li>• Very useful as an Executive Summary, can be sent to Board@mycompany</li> <li>• These are often scheduled, weekly reports</li> <li>• 5 Salespeople, 5 Conga Batches, "weekly Sales status"</li> <li>• Useful to combine a LOT of individual Salesforce reports into ONE Word or Excel doc</li> <li>• Several 1000 rows per report are ok because we're just merging a report into a document</li> </ul>	<ul style="list-style-type: none"> <li>• Cannot update individual records so you won't be able to use AC, SC, MFTS on those Accts, Opptys, whatever</li> </ul> <p>Instead of this:</p> <p>"&amp;QueryID=[Revenue]a0336000000ELxK" +</p> <p>You must do this:</p> <p>"&amp;QueryID=[Rev]a0336000000ELxK?pv0=" +</p> <ul style="list-style-type: none"> <li>• Each Operation is just one service event, even with 1000 rows in report</li> </ul>
--	--

### Conga Batch

The formula field is on a specific object and was converted from a solution (record-based merge).

#### Distributed Output:

- Can use AC, SC, and MFTS on ALL records
- Download or SendEmail or DocuSign, etc
- Common use case: weekly Invoice reminder, upcoming renewals
- Advantage: customer can Download ONE document AND update all child records
- Customers can send high quality Invoices to 1000 records at a time, via email

#### Consolidated Output:

- Not useful as One Email, Executive Summary report
- ONE document, i.e. a Batch document, must be limited to 200 records
- A distributed (i.e. email) output is up to 1000 records
- Can be a ListView
- Each record = One Service Event, even if it's a consolidated output equals multiple service events

### Global Merge

The formula field is located on the Global Merge object and was converted from an Org-wide report.

- Usually, just Download or Email
- Very useful as an Executive Summary, can be sent to Board@mycompany
- These are often scheduled, weekly reports
- 5 Salespeople, 5 batches, "weekly Sales status"
- Useful to combine a LOT of individual Salesforce reports into ONE Word or Excel doc

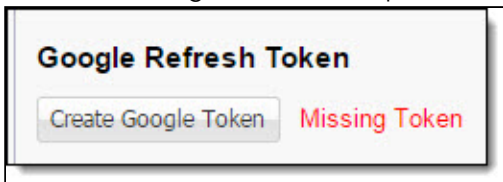
- Several 1000 rows per report are ok because we're just merging a report into a document
- Cannot update individual records so you won't be able to use AC, SC, MFTS on those Accts, Opptys
  - Instead of this: "&QueryID=[Revenue]a0336000000ELxK" +
  - You must do this: "&QueryID=[Rev]a0336000000ELxK?pv0=" +
- Each Operation is just one service event, even with 1000 rows in report

## Integrate Conga Batch with Google Drive

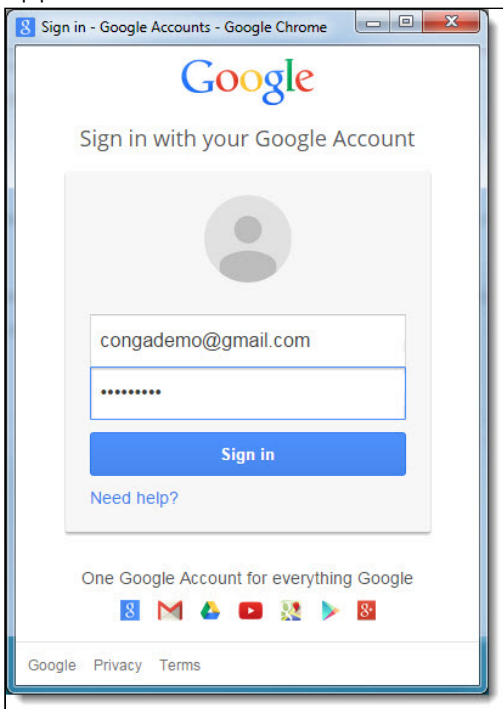
Conga Batch offers many options for saving the merged output files. These can be stored in Salesforce (for example, as attachments, Chatter files, or Content files) or in Google Drive.

### To integrate Conga Batch with Google Drive

1. Click the Conga Batch Setup tab, and then click Create Google Token.

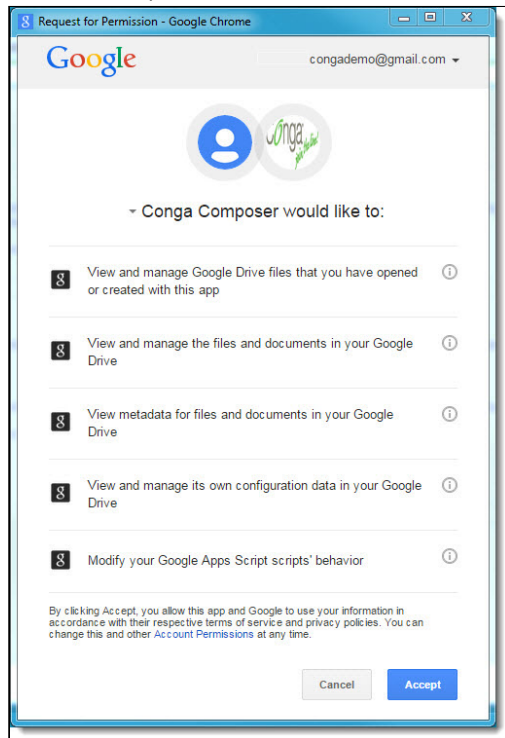


2. Sign into the appropriate Google account when the Google Account dialog box appears.

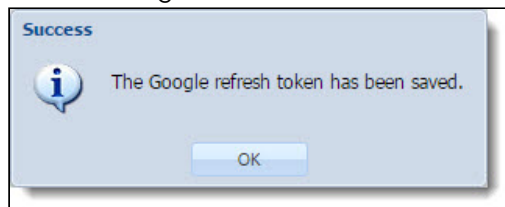


The Google account (in step 2, above) holder must have access to the folders where merged output files will be saved.

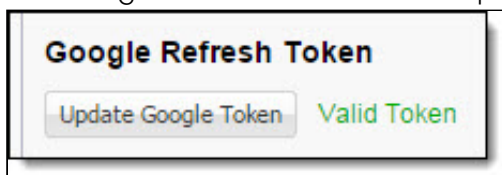
3. Another Google dialog box appears listing the permissions that is requesting. Click Accept.



4. Click OK on the dialog box confirming that the Google refresh token has been saved successfully.



The Google Refresh Token now displays as Valid Token.



## Saving a Consolidated PDF Output to a Content Library Using Conga Batch 8

To save a consolidated PDF output to a Content Library using Conga Batch 8:

1. Add defaultPDF=1 to the [Conga Batch Formula](#) field.

2. Add QMode=1 to the Conga Batch formula field. Do not use QMode=Content (QMode=1 normally translates to a downloaded output, but is required to save a consolidated output to Content).
3. On the Conga Batch record, add a Content Workspace ID and select Consolidated PDF Output File.

The screenshot displays the 'Conga Conductor Detail' page for 'Conductor-0000'. The page includes a header with the Conga logo and 'Conductor-0000'. Below the header are several sections: 'Conga Conductor Detail' with fields for Conductor Name, Title, Description, Owner, and Last Modified By; 'Master Object Source' with fields for Report ID, Query ID, Record ID, URL Field Name, Version, and Conductor Environment; 'Schedule' with fields for Schedule Description, Next Run Date, Content Workspace ID, and Consolidated PDF Output File; and 'Scheduled Conductor History' which currently shows 'No records to display'. The 'Content Workspace ID' field is highlighted with a red box, and the 'Consolidated PDF Output File' checkbox is checked.

4. Save a consolidated PDF output to a Content Library. Conga Batch must be scheduled See [Schedule Conga Batch to Run Automatically](#).

You must remove references to Content in the Formula field as well because they are referenced on the Conga Batch record. This field is not on the Conga Batch record by default. To add the field, click Edit layout and drag the field onto the Page Layout under the Schedule section.

## Scheduling a QuickMerge Report with Conga Batch (Release 7)

In this article, we will walk you through the steps of scheduling a QuickMerge Report with Conga Batch. These steps are for Release 7. For Release 8, it is recommended Conga Global Merge is used instead of QuickMerge.

Conga QuickMerge is an optional, free add-on to that helps automate and Conga Mail Merge solutions by expanding the scope of how they can be launched. Conga Batch is an optional, paid add-on to that launches on-demand or scheduled solutions in batches.

In this article, we will walk you through the steps of scheduling a QuickMerge Report with Conga Batch.

1. Build the QuickMerge Report.
2. [Create a Conga Batch formula field](#) on the QuickMerge object.
3. Create a new Conga Batch record and add two pieces: the API name of your Conga Batch formula field (formula field from Step 2) into the "URL Field Name" field on the



Conga Batch Record, and the Salesforce ID of your QuickMerge record into the "Record Id" field.

⚠ The "Query Id" and "Report Id" fields of the Conga Batch page should be left blank. When pasting the formula field name into the Conga Batch record, be sure to add the full API field name (i.e. Conga Batch\_Formula\_Field\_\_c).

4. Click Launch. Did it work? Test your solution to see if it is working correctly.
5. Schedule the Conga Batch.

If you want to include an email template, you must include the parameter [&EmailTold=]. The "Email To" parameter MUST equal a Contact ID; a User ID will not be accepted.

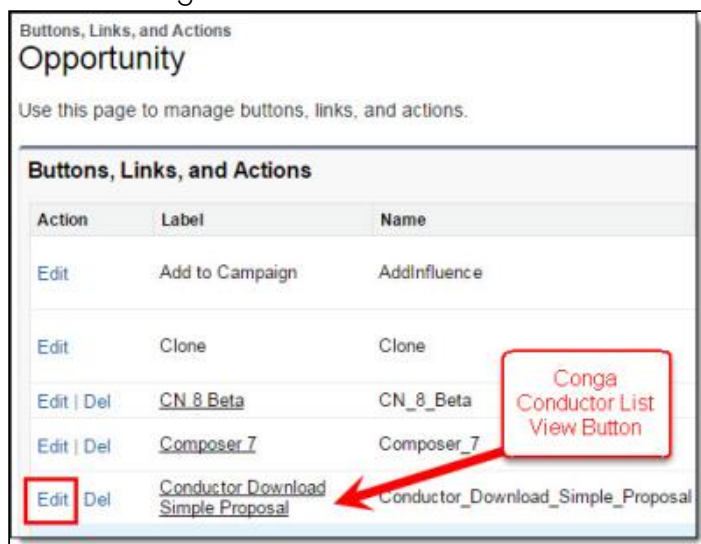
If you want to send the scheduled QuickMerge report to a User and you also want to use an email template, the best solution is to create a Contact record for this particular User.

## Upgrade Conga Batch List View Buttons to Run on Release 8

Conga Batch List View buttons launch Conga Batch from a Salesforce List View. Upgrading the Conga Batch List View button requires updating the button URL to specify that the Conga Batch list view button runs on Release 8 instead of Release 7.

### To upgrade Conga Batch list view buttons to run on Release 8 servers

1. Edit the Conga Batch List View button in Salesforce Setup.



2. Change the Endpoint URL to point to the Release 8 servers.  
Replace

<https://conductor.appextremes.com/apps/Conga/Conductor/LaunchCC.aspx>

with

<https://conductor.congameorge.com>

```

Select Field Type      Insert Field
Opportunity             Custom Button or List Editor
// STEP 1: CHANGE THE NEXT LINE TO THE TYPE OF RECORDS YOU'RE ACCESSING ON THE VIEW
var idArray = ({GETRECORDIDS( #ObjectType.Opportunity )});

// STEP 2: CHANGE THE NEXT LINE TO THE API NAME OF THE FIELD THAT CONTAINS THE
// POINTHERE URL
var urlFieldName="Conductor_URL_Simple_Proposal_o";

// DO NOT MODIFY ANYTHING BELOW THIS POINT *****
// YOU MAY PASS UP TO 50 IDS
var CongaURL = "https://conductor.appexchange.com/apps/Conga/Conductor/launchCC.aspx";

*IDMID* = idArray +
*SessionID*({Rapi.Session_ID}) +
*SecretKey*({Rapi.Partner_Secret_URL_140})+
*URLFieldName*+urlFieldName;

window.open( CongaURL, "Conga", "width=
// DO NOT MODIFY ANYTHING BELOW THIS POINT *****
// YOU MAY PASS UP TO 50 IDS
var CongaURL = "https://conductor.congameorge.com";

*IDMID* = idArray +
*SessionID*({Rapi.Session_ID}) +
*SecretKey*({Rapi.Partner_Secret_URL_140})+
*URLFieldName*+urlFieldName;

window.open( CongaURL, "Conga", "width=700,height=450,menubar=0" );
  
```

3. Click **Save**.
4. Test the Conga Batch solution on the Release 8 servers before deploying.
5. Repeat Step 1 - 4 to upgrade any other Conga Batch list view.

New Conga Batch List View buttons (those created after the upgrade) will default the var CongaURL to the version running on the Release 8 servers as long as the JavaScript for the new List View button is copied from the Sample List View Button in Salesforce Setup → Build → Create → Objects → Conga Batch → Buttons, Links and Actions.

## Upgrade Conga Batch Records to Run on Release 8

The Conga Batch record is a custom object that is installed with the Conga Batch app. It serves as a launch point for Conga Batch batch processes. Upgrading the Conga Batch record involves adding a field to the Conga Batch page layout and using that field to specify that Conga Batch record runs on Release 8 instead of Release 7.

### To upgrade a Conga Batch record to run on Release 8 servers

1. Go to your Scheduled Conga Batches and deactivate them.
2. Replace the Launch button with the Launch\_dev button.
3. Add the Formula Builder button to the Conga Batch page layout.
4. Navigate to Salesforce Setup > Build > Create > Objects > Conga Batch > Page Layouts > Conga Batch Layout > Edit.
5. Select the Buttons menu.
6. Click and drag the Formula Builder button to the Custom Buttons area. The Formula Builder is used when creating new Conga Batch solutions - it converts a button to the

required Conga Batch formula field. Don't yet save the change – another modification to the page layout is described in the next step.

7. Add the Version field to the Conga Batch page layout. Navigate to Salesforce Setup → Build → Create → Objects → Conga Batch → Page Layouts → Conga Batch Layout → Edit.
8. Select the Fields menu.
9. Click and drag the Version field beneath the URL Field Name field.
10. Click Save.
11. Navigate to the Conga Batches tab and click Conga Batch Name to open the Conga Batch record you wish to upgrade from Release 7 to Release 8.
12. Edit the Conga Batch record and change the value of the Version field to C8, which indicates Release 8.
13. Click Save. This Conga Batch batch process now runs on the Release 8 servers. However, if the Conga Batch record is configured to run on a scheduled basis, proceed with steps 7-14. To check whether the Conga Batch record is scheduled, see the Schedule section of the Conga Batch record. If the Conga Batch record is not configured to run on a schedule, skip to step 12.
14. Click the Schedule button on the Conga Batch record. A confirmation notice appears.
15. Reschedule the Conga Batch batch in the Release 8 scheduler by selecting the appropriate choices in the Frequency, Repeat, Scheduling, and Timing sections.
16. Click Save Changes.
17. Click Enable.
18. Click Return to Conga Batch Record.
19. Test the Conga Batch solution on the Release 8 servers before deploying. Please see the Conga Batch and Conga Trigger Test Guide for details.
20. Repeat steps 4 - 12 to upgrade any other Conga Batch records.
21. Remove the Version field from the Conga Batch page layout once you've finished upgrading your records.

New Conga Batch records (those created after the upgrade) will default the value of the Version field to C8, indicating those records will run on Conga Batch Release 8.

## Upgrade Conga Batch to Release 8

This topic applies only to customers currently using Conga Batch Release 7 and who wish to upgrade to Conga Batch Release 8.

- ✓ Un-schedule your Conga Batches in Version 7 before upgrading.

Conga Batch is a composite application, which means that some parts of the system are installed into the Salesforce instance in a managed package; however, most of the system runs on Conga's servers.

The portions of the system that run on our servers are updated automatically. However, the portions of the system that are installed into Salesforce are upgraded by installing Conga Batch Release 8 and following the prompts to update the installed components.

Do not uninstall the previous version of Conga Batch before upgrading to Release 8. If Conga Batch is uninstalled before upgrading, all the previously built Conga Batch solutions are lost and must be rebuilt.

## Upgrading Conga Batch to Release 8

You can upgrade the Conga Batch application to Release 8.

1. Log into the Salesforce instance in which Conga Batch Release 7 is installed.
2. Un-schedule the existing Conga Batch 7 schedules.
3. Install Conga Batch or Install Conga Batch in a Sandbox
4. Select a security level for the Conga Batch application.
5. Conga recommends selecting Install for All Users, as anyone who launches batched merges through Conga Batch requires access to the Conga Batch app components.
6. Click Upgrade. Upgrade Complete! appears confirming that the Conga Batch package has been successfully upgraded.

The Conga Batch solutions built in the previous version of Conga Batch (Release 7) are maintained in the Salesforce instance when Conga Batch is upgraded to Release 8. These solutions continue to run on the Conga Batch Release 7 servers unless they are updated to run instead on the Release 8 servers.

See the following topics to learn how to upgrade Conga Batch solutions:

[Upgrade Conga Batch List View Buttons to Run on Release 8](#)

[Upgrade Conga Batch Records to Run on Release 8](#)

## Set an API Floor

Conga Batch can consume a large number of API requests in a short period of time. To ensure that you have enough API requests for other Salesforce interactions, we recommend setting an "API floor" value on the Conga Batch Setup tab.

The API floor prevents Conga Batch from running if your total available API requests are equal to or less than the API floor value. If you do not enter a value for the API floor, Conga Batch will default the API floor value to 10% of your total API requests.

Return

**Conga conductor** Tools

### Conductor Setup

**Welcome to Conga Conductor**  
 Conga Conductor® is an optional add-on to Conga Composer® that launches on demand or scheduled Conga Composer solutions in batches.  
 For complete configuration instructions on Conductor, please see the Conga [Conductor Administrator's Guide](#). [Conductor 7 Setup](#)

Settings

**Scheduled Conductor Master Switch**  
 Enable Conductor to periodically scan your instance of Salesforce. **This must be enabled for Conductor to operate on a schedule.** This can be changed without updating the refresh token.

**API Floor**  
 Enter a value for the API Request Floor. Conga Conductor will not allow a batch job to start if the estimated number of API requests would fall below this value.  
 In the absence of this value, Conga Conductor sets a default floor to 10% of your daily allotment of API requests.

Credentials

**Salesforce Refresh Token**  
 Valid Token

**Google Refresh Token**  
 Valid Token

**Box Refresh Token**  
 Valid Token

Copyright © 2006-2015 AppExtremes, Inc - dba Conga. All rights reserved. 8.11.32-8.11.320882P USW1B-64354D21

## Troubleshooting Conga Batch

Troubleshooting help for Conga Batch.

Select one of the following topics for more information:

- ["There was an Issue Processing the Composer request"](#)
- [Conga Batch Troubleshooting Tips](#)
- [Viewing Errors in Conga Batch and Conga Trigger](#)
- [Test New Conga Batch Solutions on Release 8](#)
- [Test Using the Conga Batch Record](#)
- [Test Using a Conga Batch List View Button](#)
- [Conga Batch Missing Download Button \(Release 7\)](#)
- [Email delivery failed: REQUIRED FIELD MISSING Missing body, need at least one of html or plaintext](#)
- [Conga Batch file size limits](#)
- [Create a Conga Batch List View Button \(Lightning Experience\)](#)
- [Index outside bounds of arrays](#)
- [Problem: Please contact us at 1 \(303\)-465-1616 for subscription details \(Release 7\)](#)
- [Refreshing your Salesforce refresh token for Conga Batch](#)

- [Scheduled Conga Batch Failed to Run](#)
- [Troubleshooting Conga Batch using Hyperlink Formula Fields](#)

## "There was an Issue Processing the Composer request"

Error when launching Conga Batch.

### Cause

There are no record Ids being returned in the query referenced on the Conga Batch record.

### What to Do

Ensure the query is properly returning record Ids of the records you want to include in the batch.

## Conga Batch Troubleshooting Tips

Troubleshooting tips for Conga Batch.

When troubleshooting, one of the first things to review is the Conga Batch formula field. The primary factors that affect this field's behavior are:

- Field accessibility (the field is restricted to certain profiles)
- Whether it contains a value

The field contains, or references, fields containing ampersands that don't have spaces around them when intended as data.

A field value of Ben&Jerry's will cause an error. However, the field value of Ben & Jerry's does not. Conga Batch cannot distinguish between the data and the parameters if the ampersands within the data are not surrounded by spaces.

You can also replace ampersands with a character string Conga recognizes as a substitute for ampersands. If that doesn't work, consider creating a [hyperlink formula field](#) for testing.

## Viewing Errors in Conga Batch and Conga Trigger

In Conga Batch and Conga Trigger Release 8, error messages are logged on the Conga Batch Dashboard or Conga Trigger Dashboard.

1. Click the appropriate dashboard (custom Salesforce tab) to review batches and individual records.
2. Rows listed with Status of failed or Error encountered problems.
3. Select a failed batch or Error record in the left pane and then hover on the Error column in the Error List (right pane) to view the specific error message.

## Test New Conga Batch Solutions on Release 8

Follow these instructions if you have newly installed Conga Batch Release 8 (not upgraded from Conga Batch Release 7).

To test a solution for a newly installed Conga Batch Release 8:

**Before You Begin:** You must have Conga Batch Release 8 installed and a Conga Batch formula field and solution (a Conga Batch record or Salesforce list view button) built, but not yet deployed.

1. Gather sample Master Object records for the test batch. If you are using the Conga Batch record to launch a Conga Batch solution, you can either:
  - Write a Salesforce report or SOQL query that includes a handful of records and reference it using the Report ID or Query ID field on the Conga Batch record.
  - Reference one Master Object record at a time using the Record ID field on the Conga Batch record.

If you are using a Salesforce list view button to launch Conga Batch, then use the Salesforce list view to select a few records. Sample records are used so that the testing phase won't send documents or communications to actual recipients (customers, partners, employees, etc.). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request.

2. Launch the test batch. Launching the test batch is determined by which method you used to gather sample records.
  - Salesforce report or SOQL query: Launch from the Conga Batch record
  - One Master record at a time: Launch from the Conga Batch record
  - Salesforce list view: Launch using a Conga Batch list view button
3. Analyze the results.

This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot.

You should confirm that the following aspects of the solution are working properly:

- Reports and queries are retrieving data to be used in the template.
- Data is merging to the template, creating the merged output file.
- Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
- Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.

If any of the above components are failing or inaccurate, troubleshoot the solution and then re-test. Visit the Conga Batch Dashboard or Conga Trigger Dashboard to view errors. If you need assistance, please see [Conga Batch Troubleshooting Tips](#). If everything works correctly, deploy the solution.

## Test Using the Conga Batch Record

Follow these instructions if you are currently running Conga Batch solutions on Release 7 and wish to upgrade to Conga Batch Release 8.

To test a Release 7 solution by using the Conga Batch record:

Before You Begin: [Upgrade to Conga Batch Release 8](#). The upgrade process will retain all existing Conga Batch solutions from Release 7. These solutions can continue to run on Release 7 or can be upgraded to run on Release 8.

1. Clone the existing Conga Batch record to prevent interference with your existing solution. Include Test Release 8 in the Title of the new, cloned record.
2. Gather sample Master records for the test batch. Sample records are used so that the testing phase won't send documents or communications to actual recipients (customers, partners, employees, etc.). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request. Using the Conga Batch record to launch the Conga Batch solution, you can either:
  - Write a [Salesforce report or SOQL query](#) that includes a handful of sample records and reference it using the Report ID or Query ID field on the Test Release 8 Conga Batch record.
  - Reference one Master Object sample record at a time using the Record ID field on the Conga Batch record.
3. Ensure the proper Conga Batch formula field (API name) is referenced in URL Field Name. It should have copied over when you cloned the original Conga Batch record. Click Save.



4. Add Version to the Conga Batch page layout. We recommend that the Version field is accessible only to the System Administrator profile.
5. Change Version to C8, ensuring that the batch process runs on Conga Batch Release 8.
6. Click Launch to initiate the Conga Batch process. If you are prompted to allow or deny permissions for Conga Composer, we recommend clicking Allow, which uses a Salesforce refresh token and may be revoked at any time (Salesforce → My Settings → Personal → Advanced User Details → OAuth Connected Apps). If you select Deny, a session ID will be used for security settings instead of a refresh token.
7. Start the Conga Batch process and analyze the results. This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot. You should confirm that the following aspects of the solution are working properly:
  - Reports and queries are retrieving data to be used in the template.
  - Data is merging to the template, creating the merged output file.
  - Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
  - Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.
8. Once you confirm that the Conga Batch solution works as desired, you may delete this test record, update your original Conga Batch record to C8 for Version and use that Conga Batch record going forward.

## Test Using a Conga Batch List View Button

Follow these instructions if you are currently running Conga Batch solutions on Release 7 and wish to upgrade to Conga Batch Release 8.

To test a Release 7 solution by using the Conga Batch record:

Before You Begin: [Upgrade to Conga Batch Release 8](#). The upgrade process will retain all existing Conga Batch solutions from Release 7. These solutions can continue to run on Release 7 or can be upgraded to run on Release 8.

1. Edit the Conga Batch list view button you created in Conga Batch Release 7.
2. Change var CongaURL to <https://conductor.congamerger.com/>. This instructs Conga Batch to run the list view button on Conga Batch Release 8.
3. Create a test list view on the appropriate Master Object in order to gather sample records to use for testing. We recommend naming the list view Test CN Release 8 – [Conga Batch Solution Name]. For example, Test CN Release 8 – Oppty Invoice Email. Sample records are used so that the testing phase won't send documents or

communications to actual recipients (customers, partners, employees, etc.). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request.

4. Select a few sample records from the test list view and click the Conga Batch list view button you just updated to Release 8.
5. Analyze the results. This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot. You should confirm that the following aspects of the solution are working properly:
  - Reports and queries are retrieving data to be used in the template.
  - Data is merging to the template, creating the merged output file.
  - Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
  - Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.
6. Edit the Conga Batch list view button you created in Conga Batch Release 7.
7. Change var CongaURL to <https://conductor.congamerger.com/> This instructs Conga Batch to run the list view button on Conga Batch Release 8.
8. Create a test list view on the appropriate Master Object in order to gather sample records to use for testing. We recommend naming the list view Test CN Release 8 - [Conga Batch Solution Name]. For example, Test CN Release 8 - Oppty Invoice Email. Sample records are used so that the testing phase won't send documents or communications to actual recipients (customers, partners, employees, etc.). You may want to create new, sample records with fake data to ensure that documents or communications are not sent to actual recipients when testing. For solutions that send emails or eSignature requests, ensure that you have access to the recipient's email inboxes (or use your own email address) so you can confirm receipt of the email message/eSignature request.
9. Select a few sample records from the test list view and click the Conga Batch list view button you just updated to Release 8.
10. Analyze the results. This is perhaps the most important step in the testing process, as you will determine if everything is working correctly and you can deploy the solution or if there are inconsistencies or issues and you need to configure further or troubleshoot. You should confirm that the following aspects of the solution are working properly:

- Reports and queries are retrieving data to be used in the template.
- Data is merging to the template, creating the merged output file.
- Activity logging, field updates, saving a copy of the merged output file, and any other parameters are acting as desired.
- Fields on the email or eSignature request are referencing the proper information and the email or eSignature request is sending.

Once you confirm that the Conga Batch list view works as desired, you may use it with list views going forward.

## Conga Batch Missing Download Button (Release 7)

When running a Conga Batch solution, the Download button should appear in the Conga Batch dialog box once the merge process is complete. You will also see a message saying, "If your merge operation was configured to allow you to download files, click Download."

If the Download button does not appear there is something wrong with your template.

Conga Batch does not produce error messages. To find what is causing the error in the template run the original solution that was created for the Conga Batch routine. When running the solution an error message should appear indicating what is causing the error in the template.

## Email delivery failed: REQUIRED FIELD MISSING Missing body, need at least one of html or plaintext

ERROR MESSAGE. This error occurs when running a Conga Batch using Conga Email templates and the &Templateld parameter is left blank in the button url.

### Cause

This error occurs when running a Conga Batch using Conga Email templates and the &Templateld parameter is left blank in the button url.


### What to Do

Add the Template ID as the value for the &Templateld parameter in the button URL and then update the formula field to match. This can be done using the Formula Builder. For more information see [Update an Existing Button URL using the Conga Batch Formula Builder](#).

## Conga Batch file size limits


Conga Batch Limits	Maximum
Records in a batch	200 Records for consolidated output. 1000 Records for distributed output.

## Create a Conga Batch List View Button (Lightning Experience)

 Conga cannot support customized code. Due to the complicated nature of supporting programmers and programming languages, the integration of Conga with Visualforce, Apex, and Javascript (or other programming languages) is outside the scope of Conga support.

Use Conga Grid to integrate Conga Batch with Salesforce Classic, Lightning and Salesforce One (SF1).

Ensure that you created a Conga Batch formula field. For more information, see the [Conga Batch Product Guide](#). You will need to reference the API Name of the formula field in subsequent steps.

 You cannot create a new Apex class in a Production environment. Create the new class in a Sandbox org and then deploy the Apex class to a Production org after testing. To learn more, see [Create Apex classes in a production organization](#).

To create a Conga Batch List View Button in Lightning:

1. Create an Apex Class.
  - a. Navigate to Salesforce Setup and search for *Apex* in the Quick Find search box.
  - b. Click **Apex Classes** under the Develop section.
  - c. Click **New** to create a new Apex class.
  - d. Copy and paste the code below into the Apex Class Edit section.

```
/**
 * @description Builds list of sObject Ids from recordSetVar and constructs
 the Conductor URL
```

```

*/
public class ConductorFromListViewController {
    private final static String urlFieldName = 'test__c';
    private String partnerServerUrl;
    private ApexPages.StandardSetController controller;
    private List<Id> recordIds = new List<Id>();

    public ConductorFromListViewController(ApexPages.StandardSetController
controller) {
        this.controller = controller;
        for (sObject sObj : controller.getSelected()){
            recordIds.add(sObj.Id);
        }
        partnerServerUrl = System.URL.getOrgDomainUrl().toExternalForm()+'/
services/Soap/u/26.0/'+UserInfo.getOrganizationId();
    }

    public PageReference prepareConductorUrl() {
        PageReference conductorUrl = new PageReference('https://
conductor.congamerge.com?');
        conductorUrl.getParameters().put('MOID', String.join(new
List<Id>(recordIds), ','));
        conductorUrl.getParameters().put('SessionId',
UserInfo.getSessionID());
        conductorUrl.getParameters().put('ServerUrl', partnerServerUrl);
        conductorUrl.getParameters().put('UrlFieldName', urlFieldName);

        return conductorUrl;
    }
}

```

- e. Replace the test\_\_c field referenced in the code with the API name of your formula field.
  - f. Click **Save**.
2. Create a second Apex class after completing Step 1.

 This second Apex class is the Test class.

- a. Copy and paste the content below into the Apex Class Edit Section.

 The specific code below is for the standard Account object.

```

/**
 * @description Test Class for the ConductorFromListViewController Class
 */
@isTest
public with sharing class ConductorFromListViewControllerTest {
    /**
     * @description setup - test data setup method
     */
    @testSetup
    public static void setup(){
        List<Account> testAccounts = new List<Account>();

        Account a = new Account();
        a.Name = 'ABC Company, Inc.';
        testAccounts.add(a);

        Account acct = new Account();
        acct.Name = 'XYZ Company, Inc.';
        testAccounts.add(acct);


        insert testAccounts;
    }
    /**
     * @description testMethod1 - Test Happy Path of code
     */
    public static testMethod void testMethod1() {
        List<sObject> accts = (List<Account>)[Select Id From Account];
        System.debug(accts);
        ApexPages.StandardSetController ssc = new
ApexPages.StandardSetController(accts);
        ssc.setSelected(accts);
        ConductorFromListViewController cn = new
ConductorFromListViewController(ssc);
        cn.prepareConductorUrl();
        Test.startTest();
        PageReference pr = cn.prepareConductorUrl();
        String fieldName = pr.getParameters().get('UrlFieldName');
        System.assertEquals('test__c',fieldName, 'The UrlFieldName
parameter is not set' );
        Test.stopTest();
    }
}

```

```
}

```

- b. Replace the `test__c` field reference in the code with the API name of your formula field.
  - c. Click **Save**.
3. Create a Visualforce page.
    - a. Navigate to Salesforce Setup and search for Visualforce Pages in the Quick Find search box.
    - b. Click **Visualforce Pages**.
    - c. Click **New** to create a new Visualforce page. Label the Visualforce page as *BatchList View*.
    - d. Delete the existing code and then copy and paste the code below into the new Visualforce page.

 This example demonstrates a List View button on the Account Object. If your formula field is located on a different object, update the bold section (the value of the Standard Controller) with the correct object.

```
<apex:page standardController="Account" extensions="ConductorFromListViewController" recordSetVar="records" action="{!prepareConductorUrl}"> </apex:page>
```

4. Create a new button or link on your object.
  - a. Navigate to your object in Salesforce Setup and select Button, Links, and Actions. Click New Button or Link.
  - b. Fill out the required Label and Name fields to name the List View button.
  - c. Select List Button as the Display Type value and check Display Checkboxes (for Multi-Record Selection).
  - d. Choose Visualforce Page as Content Source value.
  - e. Choose the newly created Visualforce page as the Content Source value.
5. Add the new List View button to the list in the object's Search Layouts.
  - a. Navigate to the object in Salesforce Setup and click Search Layouts for Salesforce Classic.
  - b. Click **Edit** next to the object's List View Search Layout.
  - c. Select and move the new List View button from the Available Buttons column to the Selected Buttons column.
  - d. Click **Save**.

## Index outside bounds of arrays

ERROR MESSAGE: Ampersands (&) in Conga Batch and Conga Trigger fields.

### Cause

The formula fields for Conga Batch and Conga Trigger use ampersands (&) to identify parameters. If you pass to a parameter a field that might contain an ampersand, such as passing an ACCOUNT\_NAME field to the OFN ("Optional File Name") parameter (for example, "Ben & Jerry's"), within the formula, it has the effect of prematurely terminating the formula and returns this error: Index outside bounds of arrays

### What To Do

To prevent this from happening, Conga recommends you use the Salesforce SUBSTITUTE function to automatically substitute any ampersand with a special character string that recognizes as a substitute for ampersands (will later output the original value).

The formula below passes the Account Name field to the OFN parameter. Since some Account Name fields might contain an ampersand, the best practice is to use the SUBSTITUTE function to replace the ampersands with three ~~~ (tilde) characters:

Incorrect:

```
"&OFN=Executive+Brief+--+" + Name
```

Correct:

```
"&OFN=Executive+Brief+--+" + SUBSTITUTE( Name, "&", "{~~~}")
```

After processing the URL, will generate the output file with the original value, including any ampersand.

## Problem: Please contact us at 1 (303)-465-1616 for subscription details (Release 7)

ERROR MESSAGE: Problem: Please contact us at 1 (303) 465-1616 for subscription details (Release 7).



## Cause

Conga Batch is an add-on for Conga Composer, so it requires an active trial or subscription. When the Conga Batch trial or subscription is active but the trial or subscription has expired, the following message will be displayed: Problem: Please contact us at +1 (303)-465-1616 for subscription details.

## What to Do

Please contact our support team and explain the error message you're receiving and ask for details on your trial or subscription. This error often occurs in a Salesforce sandbox instance where the subscription may not have been used in quite some time and needs to be reactivated for Conga Batch to run.

Our support team will provide you with the necessary details for your trial or subscription and will reactivate it as applicable.

## Refreshing your Salesforce refresh token for Conga Batch

If you upgrade your version of Conga Batch from the Salesforce App Exchange, you must refresh your Salesforce token.

Conga Batch is a composite application, which means that some parts of the system are installed into the Salesforce instance in a managed package; however, most of the system runs on Conga servers. The portions of the system that run on our servers are updated automatically. Other portions of the system are installed into Salesforce.

If you upgrade your version of Conga Batch from the Salesforce App Exchange, you must refresh your Salesforce token.

Important: You must be a system admin to produce a refresh token.

## To refresh your Salesforce token

Classic	<ol style="list-style-type: none"> <li>1. Select the Conga Batch app from the Salesforce app drop-down then navigate to the Conga Batch Setup tab or click All Tabs</li> <li>2. Conga Batch Setup</li> <li>3. In the Credentials section, under Salesforce Refresh Token, click Update Salesforce Token.</li> </ol>
---------	---

Lightning	<ol style="list-style-type: none"><li>1. Use the Salesforce Lightning App Launcher and navigate to the Conga Batch Setup tab.</li><li>2. In the Credentials section, under Salesforce Refresh Token, click Update Salesforce Token.</li></ol>
-----------	---

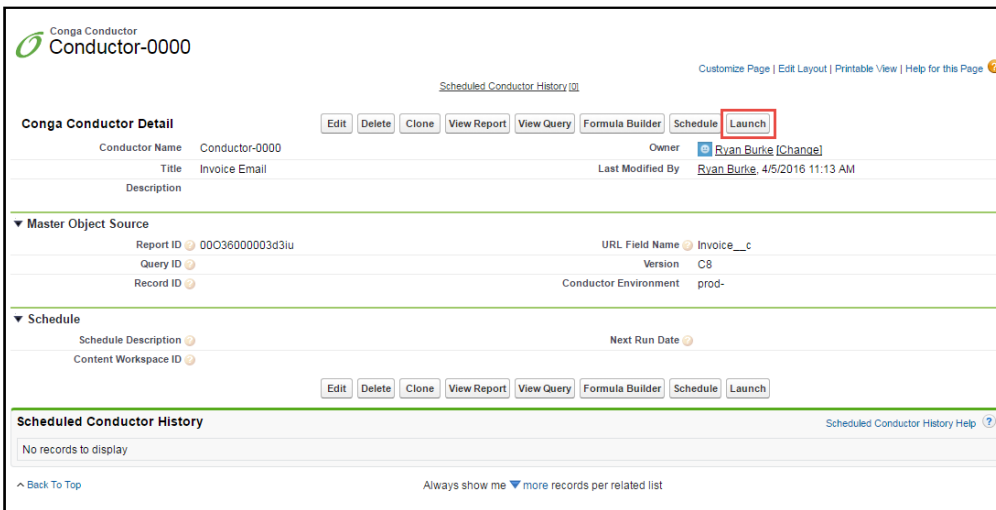
For more information about Conga Batch setup, see the [Conga Batches Tab](#).

## Scheduled Conga Batch Failed to Run

If your scheduled Conga Batch fails to run, the first thing to check is whether the Master Switch on the [Conga Batch Setup Tab](#) is enabled (which is required for Conga Batch schedules to run). If it is not enabled, it is possible the Master Switch was disabled due to a Salesforce limitation in which a user can only have four OAuth tokens per application.

### Why Does This Happen?

The default Launch button on the Conga Batch record Page Layout uses OAuth. Thus, manually launching Conga Batch with this button can "kick out" the OAuth token used to run the schedule because of the limitation mentioned above.



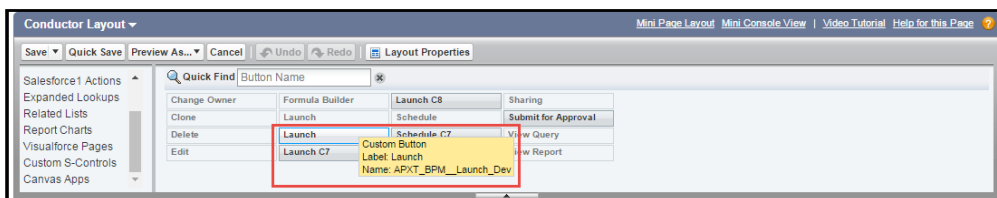
This can also happen if you click Allow too many times when generating a debug URL since it also creates an OAuth token.

## What to Do

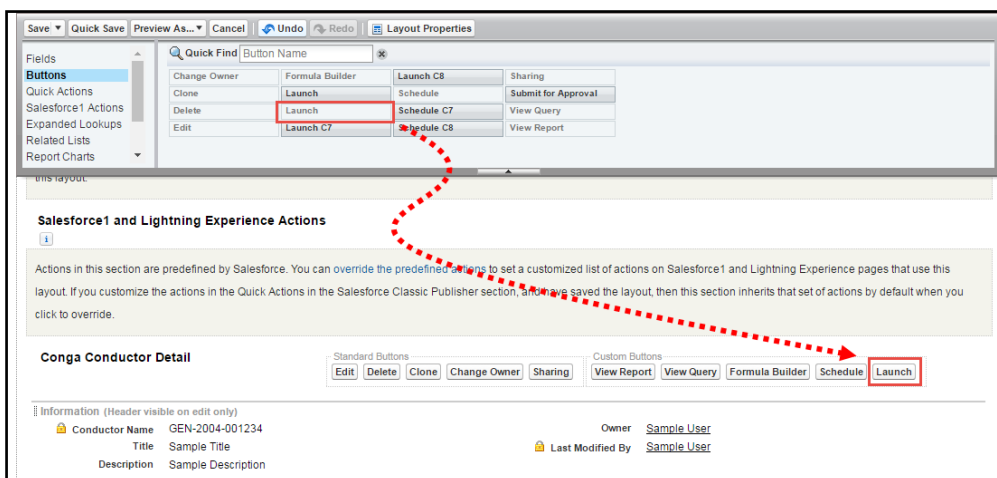
One thing you can do to help ensure the Master Switch is not disabled from too many OAuth tokens being created is to change the default Launch button on the Conga Batch record to use the non-OAuth Launch button that came with the package (the one that says "APXT\_BPM\_\_Launch\_Dev" when hovering over it. See screenshot below).

To change the button:

1. Click Edit Page Layout on the Conga Batch record.
2. Select the Buttons section, drag the default button off the page and drag the new Launch button into the Custom Button region on the page.
3. Click Save. (See screenshot below).

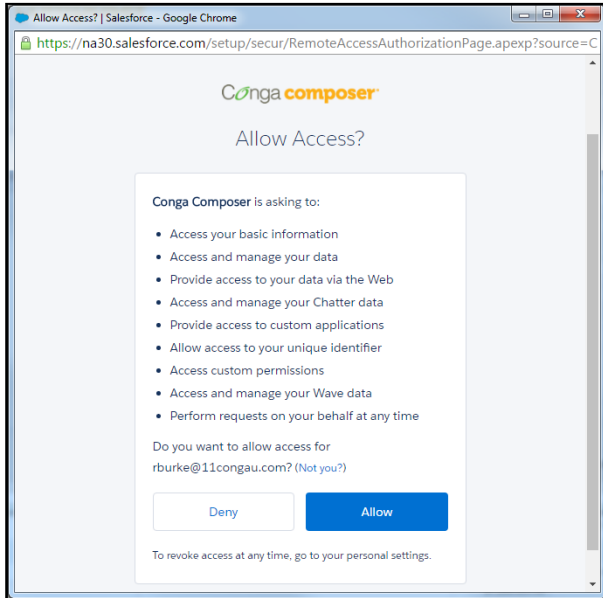


### APXT\_BPM\_Launch\_Dev Button (non-OAuth)



Drag the non-OAuth button on the layout

**i** When generating a debug URL, we suggest only clicking Allow when you are on the phone with a Conga Business Analyst. If you are generating a debug URL for your own use, click Deny (you can still get the debug URL, it just won't create an OAuth token).



Access Composer Debug URL.

## Troubleshooting Conga Batch using Hyperlink Formula Fields

Conga recommends the use of the Hyperlink Formula Field technique as a first step whenever challenges arise in the use of Conga Batch.

This hyperlink field will be placed on the Master Object page layout and allows the system administrator to run the Conga Batch process without having to invoke the Salesforce batch process. The hyperlink field launches the user interface, but runs the Conga Batch solution, thus providing all of the user interface capabilities of (such as the View Data workbook, templates, user license status, and error messages) while running Conga Batch.

This approach also helps identify problems associated with access to templates, reports and queries that can cause Conga Batch process failures resulting from Salesforce session timeouts.

## To create a hyperlink formula field to troubleshoot a Conga Batch solution

1. Create a custom formula field on your Master Object.

**New Custom Field**

**Step 1. Choose the field type**

Specify the type of information that the custom field will contain.

**Data Type**

- None Selected
- Auto Number
- Formula
- Roll-Up Summary

2. In the Field Label box, enter Test and the solution name.

**Step 2. Choose output type**

Field Label: **Test Conga Workflow Invoic** Field Name: **Test**

**Formula Return Type**

- None Selected
- Checkbox
- Currency
- Date
- Date/Time
- Number
- Percent
- Text

Select one of the data types below:

- Calculate a boolean value**  
Example: `TODAY() > CloseDate`
- Calculate a dollar or other currency amount and automatically format the field as a**  
Example: `Gross Margin = Amount - Cost__c`
- Calculate a date, for example, by adding or subtracting days to other dates.**  
Example: `Reminder Date = CloseDate - 7`
- Calculate a date/time, for example, by adding a number of hours or days to another**  
Example: `Next = NOW() + 1`
- Calculate a numeric value**  
Example: `Fahrenheit = 1.8 * Celsius__c + 32`
- Calculate a percent and automatically add the percent sign to the number**  
Example: `Discount = (Amount - Discounted_Amount__c) / Amount`
- Create a text string, for example, by concatenating other text fields.**  
Example: `Full Name = LastName & ", " & FirstName`

3. In the Formula Return Type area, click Text.
4. Click the Advanced Formula tab and, in the Functions area, select HYPERLINK, and click Insert Selected Function.

**Step 3. Enter Formula** Step 3 of 5

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `Full Name = LastName & ", " & FirstName` More Examples...

Single Formula **Advanced Formula**

Insert Field: `Test Conga Workflow Invoice Email (Text) = HYPERLINK(url, friendly_name [, target])`

Insert Operator: `+`

Functions:

- All Function Categories
- EXP
- FIND
- FLOOR
- GEOLOCATION
- GETPERSONID
- HYPERLINK**

Insert Selected Function: `HYPERLINK(url, friendly_name [, target])`

Create a hyperlink

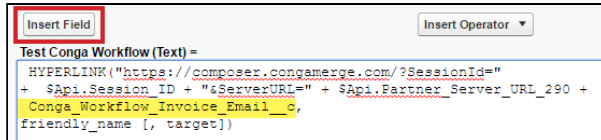
5. In the HYPERLINK function string, replace url with the following:  
For Conga Batch:  
Lightning:

```
"/apex/APXTConga4__Conga_Composer?ServerURL=" + $Api.Partner_Server_URL_370 +
```

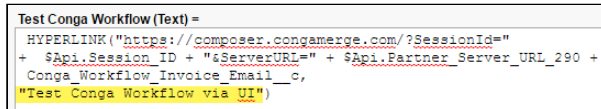
Classic:

```
"https://composer.congamerge.com/composer8/index.html?SessionId=" +
$Api.Session_ID + "&ServerURL=" + $Api.Partner_Server_URL_290 +
```

6. Enter a new line after the URL and enter the API name of the formula field.



7. Click Insert Field to insert the field.
8. In the HYPERLINK function string, replace `friendly_name` with the name of your hyperlink as it will appear on the page layout.



9. In the HYPERLINK function string, delete `[, target]`.

If you're using Salesforce Professional Edition, the hyperlink troubleshooting field will be visible to all users due the lack of field-level security. Therefore, we recommend including the `&DC` (Disable Conga Composer) parameter with an IF function to limit access to this field to System Administrators only:"

```
&DC=" + IF( $Profile.Name = "System Administrator", "0", "1")
```

Your full hyperlink formula field would then be:

Lightning:

```
HYPERLINK("/apex/APXTConga4__Conga_Composer?ServerURL="+
$Api.Partner_Server_URL_370 + Conga_Workflow_Invoice_Email__c,"Test Conga
Trigger via User Interface")
```

Classic:

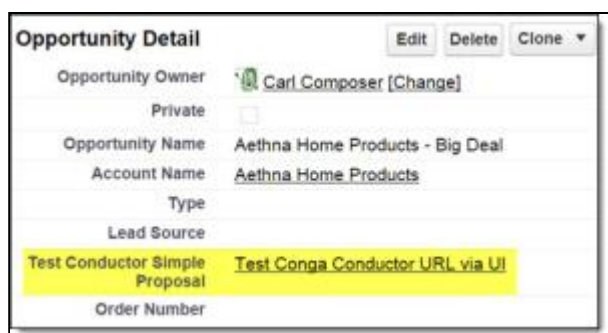
```
HYPERLINK("https://composer.congamerge.com/?SessionId=" + $Api.Session_ID + "&Se
rverURL=" + $Api.Partner_Server_URL_290 + Conga_Workflow_Invoice_Email__c,"Test
Conga Trigger via User Interface")
```

10. Click Next to establish field-level security. Note: We recommend making this field visible to only the System Administrator profile.

11. Add the field to the appropriate page layouts.
12. Click **Save**.

You now have a hyperlink field on your Master Object page layout to run your solution without having to initiate the Salesforce batch process.

You can use this troubleshooting link to launch Conga Batch or Conga Trigger through the user interface to view error messages, download the View Data workbook, and so on.



QMode parameters will not function when you run Conga Batch or Conga Trigger using this technique, so you must run your solution using the appropriate action from the user interface (such as Merge & Download or Email).

## Conga Mail Merge

### Important

After many years of success in the market, Conga has decided to officially retire the **Mail Merge** product. You will no longer be able to use **Mail Merge** on or after **1 August 2022**. For more information, refer to [Mail Merge Retirement Announcement](#).

Conga Mail Merge® is an optional, free add-on to Conga Composer® that simplifies mass mailings. Use Conga Mail Merge to generate hundreds of printable letters, labels, and envelopes at once.

[Step-By-Step Guide To Using Mail Merge](#)

## Mail Merge Product Guide

### Learn Mail Merge Basics

[What is Conga Mail Merge?](#)

## Use Mail Merge

- [Gather Salesforce Data](#)
- [How to Use Salesforce Reports with Conga Mail Merge](#)
- [What Field Names are Available in Mail Merge?](#)
- [Create Merge Fields in Mail Merge](#)
- [Mail Merge Product Guide](#)

## Launch Mail Merge from QuickMerge

[Launch Mail Merge from a Custom Button, Link or Tab](#)

## Troubleshooting tips

[Template Troubleshooting Tips](#)

### Important

- Conga Mail Merge does not currently work in Salesforce Lightning. You must use Salesforce Classic.
- Conga Mail Merge is not supported in Chrome v91 or Edge browsers.
- Conga Mail is supported in Firefox and Internet Explorer browsers.
- Conga Mail Merge does not support the use of custom queries.
- You can use only one template at a time with Conga Mail Merge.

## What is Conga Mail Merge?

Simplify mass mailings by generating hundreds of printable letters, labels, and envelopes all at once. Log activities, update fields, and create follow-up tasks for every record in a mail merge. Automate the process to a single click with Conga QuickMerge.

Conga Mail Merge is a step-by-step wizard that guides you through the process of creating batches of documents and logging the associated activity in Salesforce. You access Conga Mail Merge by clicking the Conga Mail Merge tab.

With Conga Mail Merge, you can retrieve data from Salesforce by entering search criteria, using specially configured Salesforce reports, or by employing Conga Queries. After the data retrieval, Conga Mail Merge then merges the data into a Word template to produce



tens, hundreds or thousands of documents. (Conga Mail Merge supports only Word templates.)

You can launch Conga Mail Merge directly from a Campaign. For information, see [Launching Conga Mail Merge from a Custom Button, Link or Tab](#).

You can streamline (or even bypass) the Conga Mail Merge wizard with Conga QuickMerge.

## Using Mail Merge

The information in this section relates to Mail Merge, including configuring, using, gathering data, and building templates.

## Gather Salesforce Data for Mail Merge Solutions

In Conga Mail Merge, you have several choices for searching Salesforce data. You can:

- Enter search criteria for Contacts, Leads, or Accounts
- Select a Campaign
- [Select a Salesforce Report](#)

### Contacts/Leads/Accounts

If you search for Contacts, Leads, or Accounts, Conga Mail Merge retrieves the desired records and makes all fields on those records available for merging.

### Campaigns

If you select a Campaign, Conga Mail Merge retrieves all Contacts and Leads associated with the Campaign and likewise makes all fields on those records available for merging.

### Reports

If you select a Salesforce Report as the data source, the fields available for merging depend, in part, on the columns defined in the Report. In the next section, we'll examine several possible scenarios for defining a Report for use with Conga Mail Merge.

## Searching for objects

### How do I search for objects other than Contacts, Leads, or Accounts?

To access objects other than Contacts, Leads or Accounts, use a Salesforce Report. In doing so, you can access virtually any information in Salesforce. You can find more information about using Reports with Conga Mail Merge in the next section.

## How to Use Salesforce Reports with Conga Mail Merge

When using a Report as the data source for a Conga Mail Merge operation, Conga Composer evaluates the contents of the Report and makes certain decisions based on that content. The key decision is whether the report is a Master Object Report or a Master/Detail Report.

What is a Master Object Report? A Master Object Report is useful when you want to create documents that require data from a single object. For example, correspondence or brochures might be the kind of documents you would produce with data retrieved from a Master Object Report. This kind of report has only one column, namely the Salesforce Id of the records in Salesforce you want to retrieve. This column is known as the Master Object Id.

From this kind of report, Conga Mail Merge automatically retrieves all fields from each record listed on the Report.

For the reports to appear in the list of available reports, you must save Conga Mail Merge reports to the Conga Reports folder. If you wish to use other folders to store Conga Mail Merge reports, the Conga Reports folder must be empty and you must include Conga in the Report Description field when you save the report.

To create a Master Object Report:

1. From the Reports Tab, click **New Report**.
2. Select the type of data you want to report on and click **Create**.
  - For the report format, select **Tabular**.
  - For the columns to include, click **Remove All Columns**, then include the Id of the master object.
  - For the Report Filters, define the Standard and Advanced Filters as desired.
3. Save the report to the Conga Reports folder.

Say you'd like to send a document to a group of Accounts. In that case, the report you create for Conga Mail Merge would contain a single column: Account Id.

When Conga Mail Merge discovers the Account Id column in the report, it retrieves all fields from each Account listed on the Report. Therefore, it isn't necessary for you to include any other columns from the Account on the Report, except for the Account Id. A report with a single Id column isn't very useful for people to read, but for Conga Composer, that single column tells volumes! Users will only see the name of the report from the Conga Mail Merge wizard, and therefore they won't need to open the report directly.

What if I include other columns that belong to the master object? You can make your Salesforce reports more readable for people by including other columns from the Master Object, but Conga Mail Merge doesn't need them. For example, if your Master

Object Id is the Account Id column, columns such as the Account Name or the Account's Billing City are irrelevant to Conga Mail Merge, since it will have already retrieved all columns from the Accounts on the report. The additional columns make the report look nice though, in case people need to view the report.

We commend that the ID column be placed on the farthest right and include no OTHER Id columns. You can sort the report by clicking a column header, then click Save.

I want to create invoices or quotes based on Opportunities. How should I set up my report? Conga Composer automatically retrieves more information than you might expect when using either the Opportunity Id or Contact Id as the master object. In those cases, Conga Composer automatically retrieves the following:

Master Object ID Column	Bonus Fields
Opportunity ID	In addition to the Opportunity fields, Conga Composer retrieves: All Opportunity Line Item fields for all related line items (as the "OpLineItems" dataset) All Account fields for the related Account All Contact fields for Primary Contact All Account fields for Primary Contact
Contact ID	In addition to the Contact fields, Conga Composer retrieves: All Account fields for the related Account

Therefore, when creating a report for Conga Mail Merge that uses Opportunity Id as the Master Object Id, it is not necessary for you to include Opportunity Line Items on the report. Likewise, for reports that use Contact Id as the Master Object Id, you don't need to include columns from the Contact's related Account.

What if I don't include an Id column at all? Although Conga Mail Merge can use a report that has no Master Object Id, we don't recommend it because Conga Mail Merge's ability to log activities against those records won't work. (Conga Mail Merge requires a master object Id column for activity logging to function properly.)

What if I include columns that aren't from the master object in my report? Say the master object is the Account Id, but I also include Contact columns in my report. If your report includes columns from an object other than the master object, you've created a "Master/Detail Report", which is discussed in the next section.

What is a Master/Detail Report? A Master/Detail Report is useful when you want to create a batch of documents that require data from more than one object. For example, say you want to print a batch of Quotes with Quote Line Items, where the Quotes and Quote Line Items are custom objects. You could use a Master/Detail report to represent Quotes (the master object) and their related Quote Line Items (the detail object). This kind of report

builds on the concept of a Master Object Report. A Master/Detail Report has more than one column, one of which is the Salesforce Id of the Master Object. The other columns in the report would be from a related object.

The Master Object Id column is treated in the same manner as in a Master Object Report, namely, Conga Mail Merge automatically retrieves all fields from the objects listed on the report. The remaining columns of the report are made available for merging as a separate dataset. This is useful in that a single Report can define two distinct sets of data, namely the Master data and the Detail data.

1. From the Reports Tab, click **New Report...**
2. Select the type of data you want to report on. Choose a type that includes two or more objects.

 **Tip**

Focus the scope of the Conga Mail Merge report to pull only the objects and fields necessary.

3. For the report format, select **Tabular**.
4. For the columns to include, click **Remove All Columns**, then include the Id of the master object.
5. Drag any columns from the related object you want to include in your merge operation. Leave the Master Object Id is at the left end of the list. It doesn't matter whether the Master Object Id is at the left or right, but the order can affect the column headers in View Data, so select left or right and stick with it!
6. For the Report Filters, define the Standard and Advanced Filters as desired
7. Save the report to the Conga Reports folder. Say you'd like to generate documents addressed to each of your Accounts. On each document, you'd like to list the names of the Contacts associated with that Account. Here, you could use a Master/Detail Report to define the data you need. In this case, you'd create a Report based on "Contacts & Accounts". The Account Id would be the Master Object Id, and the remaining columns of the report would come from Contacts (Salutation, First Name, Last Name, etc.)

When Conga Mail Merge encounters the Account Id, it retrieves all fields from the Accounts listed on the Report. Those values appear on the Master sheet in the View Data workbook. The Contact details are retrieved from the report itself. Those values appear on the ReportData sheet in the View Data workbook.

The ReportData sheet shows the Contact data, taken directly from the report: Congratulations! The data is now ready to be merged into a Word template to generate multiple documents. The next step is to create your Word template and

upload it to Salesforce. See [Create and Store a Mail Merge Template](#) for more information.

## Launching Conga Mail Merge from a Custom Button, Link or Tab

You can launch Conga Mail Merge from a button, link, or custom tab. This is particularly useful for launching a merge operation from a Campaign in that the campaign can be pre-selected. To implement this, add a custom button or link to a page layout with the following details.

### Display Type:

Detail Page Link or Detail Page Button Behavior: Display in the existing window with sidebar URL:

```
https://massmerge.appextremes.com/apps/Conga/Welcome.aspx?SessionId={!API.Session_ID}&ServerUrl={!API.Partner_Server_URL_80}
```

### Example URL

```
https://massmerge.appextremes.com/apps/Conga/Welcome.aspx?SessionId={!API.Session_ID}&ServerUrl={!API.Partner_Server_URL_80}&id={!Campaign.Id}&ReportID=000i0000006IF4C?pv0={!Campaign.Id}&TemplateID=a0461000001LyhM
```

## Further Refine Your Solution

You can include any of the following parameters to further refine your solution:

Parameter	Description
&ReportId=[Id]	Pre-selects the specified Report on the Search Criteria page Requires: A 15-character Salesforce Id of a Report. May optionally include filter criteria.  Do not use this in combination with any other "data source" parameter (such as CampaignId).
&CampaignId=[Id]	Pre-selects the specified Campaign on the Search Criteria page. Requires: A 15-character Salesforce Id of a Campaign. Do not use this in combination with any other "data source" parameter (such as ReportId).

&TemplatelD=[Id]	Pre-selects the specified template on the Document/Letter page Requires: A 15-character Salesforce Id of a template record.
&LabelTemplatelD=[Id]	Pre-selects the specified template on the Label page Requires: A 15-character Salesforce Id of a template record.
&EnvTemplatelD=[Id]	Pre-selects the specified template on the Envelope page Requires: A 15-character Salesforce Id of a template record.
&BPW=[0 or 1]	Bypass Wizard Navigates directly to the Download page Requires: <ul style="list-style-type: none"> <li>• ReportId or CampaignId</li> <li>• TemplatelD, LabelTemplatelD and/or EnvTemplatelD</li> </ul>

### Important

You can use only one template at a time with Conga Mail Merge.

## How to Add Page Numbers to your Conga Mail Merge documents

This article details a method in which you can add page numbers to your Mail Merge letters and documents. These page numbers will be specific to each record in your batch and reset when the letter for your next record begins. The page number must be in the header or footer.

### Header/Footer Page Numbers

The field needed to display the current page number is called a Page field (Insert > Quick Parts > Field > Page). After inserting the Page field, highlight the number and right click in the highlighted area, then choose Format Page Numbers. Once in the page number formatting menu deselect Continue from previous section. This will ensure the page number resets for each record in your batch.

To achieve the format, "Page 1 of 1", we need to insert an additional field which displays the current number of pages in the current section. The field to display total pages in current section is called a SectionPages field (Insert > Quick Parts > Field > SectionPages).

# Conga Mail Merge - Using a Salesforce Report for a Label in a Word Template

When using fields from a Salesforce Report to populate your label template for Conga Mail Merge, follow these best practices.

The report should have the master object ID field in the far right column. When the ID is placed in far left column or anywhere in the middle, all of the field columns after it are renamed which can cause errors in the merge process.

If you need a combination of Master fields and report fields on your label, add the Master fields to this report (unlike other Conga Mail Merge or Conga Composer processes, labels only support fields from one data source - either Master or report, not both). This report must be saved in the reports folder, Conga Reports, that was installed with the Conga Mail Merge app.

 DO NOT use TableStart or TableEnd fields in your Word template, only the merge fields should be present. Click **Update Labels** anytime you make changes to the top left label.

Upload the Word template into Conga Templates. Note: When using a label with report data, you will not be able to merge from a local template; you must use the template from Conga Templates.

## Template Type

Must be set to Label

## Label Template: Use Detail Data

Must be checked

In the Conga Mail Merge wizard, select:

1. Select Labels.
2. Select the report you built.
3. Choose a label template from Salesforce.com and click the template you uploaded.
4. Click the Labels button to download your output labels.

When using label templates in Conga QuickMerge Links, follow the same instructions as above and also paste the Conga Template ID (as it appears in the Salesforce URL on the Conga Templates record) into the Label Template Id field.

## Printing Batches of Word Documents

When generating a batch of documents with Mail Merge, you can choose to generate a single (potentially large) output file, or separate files (zipped into a single download).

In some cases, Mail Merge will automatically generate separate files (specifically if the output file would exceed 40 MB). After downloading the zip file, unzip it to a directory of your choice. You may then print the files by selecting **Edit > Select All > File > Print**. Word will open for each document and the printed output will be directed to your computer's default printer. This option is good if you have a small number of documents.

For more control over the printed output, you may choose to create the following Word macro. The benefit of doing so is that you can set printer options before printing, and Word does not have to open and close for each document.

To create a Batch Print Macro:

1. Open the template and create a macro.

Product	Steps
Word 2003	Click Tools > Macro > Macros > ...
Word 2007/2010:	Click Tools > Macro > Macros > ...

2. Click the **Developer** ribbon.
3. In the Code group, choose **Macros**.
4. Enter **BatchPrint** as the Macro Name and select **NORMAL.DOT** in the "Macros In:" list box.
5. Click **Create**.
6. Paste or type the following code on a blank line between the Sub BatchPrint() and End Sub:

```
Dim MyPath As String, MyName As String
MyPath = "c:\my documents\" ' == CHANGE TO YOUR OWN FOLDER LOCATION ==
MyName = Dir$(MyPath & "*.doc")
Application.PrintOut _
Background:=False, _
FileName:=MyPath & MyName
MyName = Dir
Do While MyName <> ""
Application.PrintOut _
Background:=False, _
FileName:=MyPath & MyName
```



```
MyName = Dir  
Loop
```

7. Choose **File > Close** and Return to Microsoft Word.
8. To run the macro: Choose Tools > Macro > Macros, select BatchPrint and then Run. (Alternatively, you can create a button to execute the macro.)  
The macro will prompt you to select a folder on your computer. (Select one that contains Word documents.) It then opens each Word document within the folder and prints it to whatever printer is selected from the **File > Print** dialog box.

## Create and Store a Mail Merge Template

After configuring Mail Merge to retrieve the desired data from Salesforce, your next task is to create a merge template using Microsoft Word.

 Conga Mail Merge supports Word templates only.

Refer to the following resources for information on creating Mail Merge Word templates:

- [What Field Names are Available?](#)
- [Create Merge Fields in Mail Merge](#)

## Mail Merge Word Template Guide

After creating your Mail Merge Word template, you can store your template in any of three locations:

- The Salesforce template repository  
The Conga Template Manager
- Your local hard drive or network drive

### **Important**

The Mail Merge template must be uploaded as an Attachment. If it is uploaded as a File, it will not work as expected.

We recommend you store your templates in the Conga Template Manager to make it easily accessible from the Mail Merge wizard. Text-based Merge Fields DO NOT work with Mail Merge.

## Field Names Available in Mail Merge

An essential tool in the development of Conga Mail Merge solutions is the View Data button. The View Data button is located at the lower-right corner of the third step of the Conga Mail Merge wizard.

To generate a View Data workbook:

1. Click the **Conga Mail Merge** Tab in Salesforce.
2. Select the type of document you want to create.
3. Enter your selection criteria or select a Salesforce Report as your data source.
4. Scroll to the bottom of the page and click the bottom labeled **View Data**.

Conga Mail Merge assembles the data and presents it in an Excel workbook. Each dataset comprises a sheet within the workbook.

### **Tip**

Locate fields quickly within the View Data workbook by using Excel's Find command, **Edit > Find**. Use it to look for known field values, thus revealing the Salesforce field name. Then click the column header and copy it to the clipboard so you can use it when creating a field in your template.

Since Conga Composer displays data across several sheets, when using Excel's Find command, you can search the entire workbook.

### **Tip**

If Opportunities are your Master Objects, Conga Mail Merge automatically makes the Opportunity Line Items available for merging on the OppLineItems sheet.

Excel displays up to 256 columns in a sheet. If your dataset has more than 256 columns, it is represented on multiple sheets in the View Data workbook.

## Create Merge Fields in Mail Merge

With the [View Data Workbook](#) close at hand, you're ready to begin creating merge fields.

For columns from the Master dataset, the Org dataset, or the User dataset, create merge fields wherever you want them according to the instructions below. For columns from other sheets, such as ReportData, in addition to creating the merge fields, you'll need to place those fields within a Detail Region using TableStart and TableEnd special merge fields.

To create a Word merge field:

1. In Word, create or open a document to use as your template
2. Switch to the View Data Workbook in Excel. Locate the field you'd like to use and copy the column header to the clipboard **Edit > Copy or Ctrl-C**.
3. Switch to Word and move to the location where you'd like the merge field.
4. Create the field.
5. In the Field Names list, select **MergeField**.
6. In the Field Name text box, paste the field name that was copied to the clipboard in step 2.
7. Click **OK**. If the field just created is from a detail data set such as ReportData, you'll need to position the field within a Detail Region.

You can display the field codes in Microsoft Word by pressing **Alt-F9**.

Let's say you want to create a field for the LastName field. We'll select the field from the View Data Workbook to copy it to the clipboard. Then, in Word, we'll create a new merge field and paste the field name (LastName) into the Field name: text box.

The resulting field looks like:

«LastName»

## Creating Label Merge Templates in Word

Label merge templates differ slightly from other document templates. Creating envelope templates is similar.

To create label templates in Word 2007/2010:

1. Create a blank document.
2. On the Mailings ribbon, select **Start Mail Merge > Labels**.
3. Select the Avery Label style and click **OK**.
4. OPTIONAL: Click **Table Tools > Layout > View Gridlines** to show gridlines in the label template.
5. From the Template Builder window, add merge fields to the upper-left cell only.
6. On the Mailings ribbon, click **Update Labels**.
7. Update Labels copies the fields from the upper-left cell to the other cells on the page and automatically inserts Next Record fields.

## Troubleshooting Mail Merge

Troubleshooting help for Mail Merge. You can find topics to help you troubleshoot issues and resolve problems. See various issues in this section.

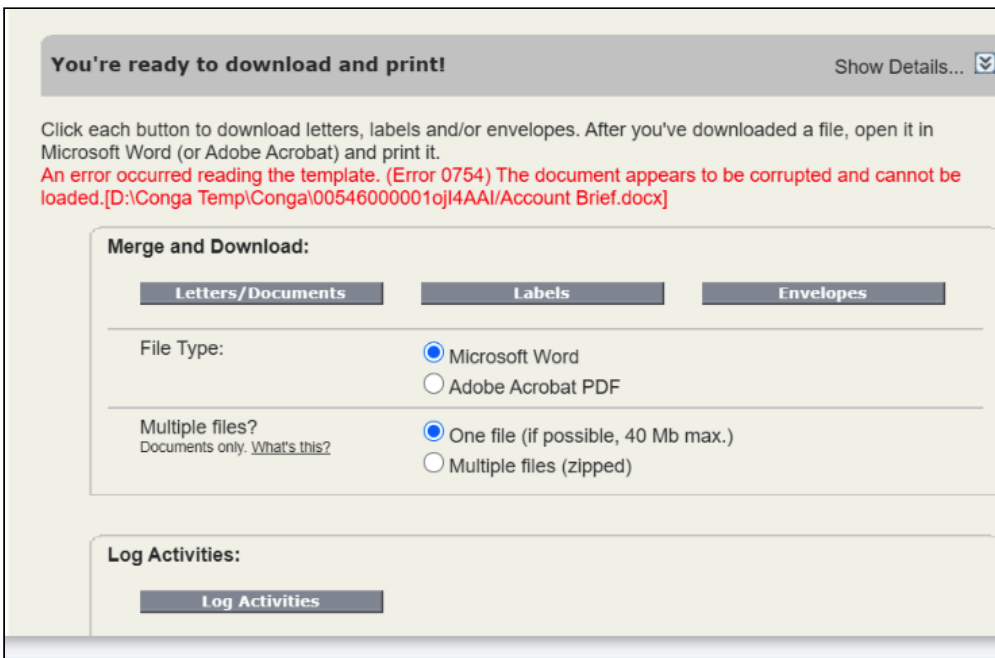
## Mail Merge in Salesforce Lightning

### Warning

Conga Mail Merge is not compatible in Salesforce Lightning.

When attempting use Mail Merge in Lightning, Mail Merge cannot retrieve reports or templates designated in a solution.

Users typically encounter the following error message when attempting to generate a document with Mail Merge in Lightning: **An error occurred reading the template. (Error 0754) The document appears to be corrupted and cannot be loaded.**



### Important

Ensure that you use Mail Merge in Salesforce Classic to avoid the aforementioned error.

## Template Troubleshooting Tips for Conga Mail Merge

Ensure Conga Template Manager (Conga Templates) is enabled.

This allows Conga Mail Merge to use it as the template source rather than Salesforce's Mail Merge Templates. Upload the template as a .doc Word file and Salesforce Attachment, rather than a Salesforce File.

To ensure the template is enabled:

1. Click the **Conga Mail Merge** tab then click the Setup link.
2. Check the box to **Enable the Conga Template Manager** and click **Save**.

Confirm Users have access to Conga Templates.

### **Is the Conga Templates object deployed?**

Check in Salesforce Setup > Build > Create > Objects > Conga Template. The Deployment Status field should be set to Deployed.

### **What are the Sharing Settings?**

Check in Salesforce **Setup > Administration > Security Controls > Sharing Settings > Conga Template**.

Public Read Only is the setting Conga recommends which allows all users to access the Conga Templates object by default but only the owner of the Conga Template record or a System Administrator can edit the records. Public Read/Write allows all users full access (including the ability to edit) to all Conga Template records by default.

Private allows access to Conga Template records only to the owner of the Conga Template record. Records then must be shared to Groups, Profiles, or individual Users on record-by-record bases or through Sharing Rules.

### **Does the Profile have access to the Conga Templates object?**

Profiles can be used to control who has access to objects, beyond the Organization-Wide Defaults.

Navigate to Salesforce **Setup > Administer > Manage Users > Profiles > Desired User Profile] > Custom Object Permissions > Conga Templates**.

The minimum level of access required to access templates stored in Conga Templates is "Read."

## **Access View Data Workbook**

On the last screen of the Mail Merge wizard, select Download View Data. This is an Excel Workbook that contains all of the data available to Conga Mail Merge. If data is not in your View Data Workbook, it will not be in your template. If data is not in your View Data workbook this means the report used to gather the data is not properly returning results. Common issues to check:

- Are you using the correct report type?
- Are the correct filters applied to your report?

If data is in your View Data workbook but not in your template, then the problem is the template. Double check that the merge fields were added correctly.

✓ **Tip**

Do not use a Colon: in the document file names.

## An error occurred logging activity: Character reference is an invalid XML character

### Issue

An error occurred logging activity: Character reference "" is an invalid XML character.

### Cause

This error occurs when trying to include the body of the letter in the activity logging when the Mail Merge template contains column breaks. Column breaks cannot be added to Salesforce activities.

### What to Do

Omit the column breaks from the Mail Merge template OR select **No (saves space)** in the "Include body of the letter?" section in Step 7 of the Mail Merge Wizard.

## Connection to Salesforce Failed. Unable to open object of type System.Web.HttpCookie to type SystemJConvertible (Release 7)

### Issue

Connection to Salesforce Failed. Unable to open object of type 'System.Web.HttpCookie' to type 'SystemJConvertible' (Release 7).

## Cause

This error message occurs using Conga Mail Merge.

## What to do

To correct this error, clear your browser cookies and restart the browser.

# Master Object query failed: ID column not included in SOQL select statement. Please add it. (Release 7)

## Issue

Master Object query failed: ID column not included in SOQL select statement. Please add it. (Release 7).

## Cause

This is best explained with an example. Below, Conga Mail Merge is expecting the first value to be "ID" rather than "Product\_\_c.ID".

```
SELECT Product__c.IDProduct__c.Name,Product__c.Type,FROM Purchase_Order_Product__cWHERE Product__c.Vendor='001T000000mABSn'
```

## What to do

This error would be corrected by using the following:

```
SELECT IDProduct__c.NamFROM Purchase_Order_Product__WHERE Product__c.Vendor='001T000000mABSn'
```

## QuickMerge Product Guide

### Important

After many years of success in the market, Conga has decided to officially retire the **Quick Merge** product. You will no longer be able to use **Quick Merge** on or after **1 August 2022**. For more information, refer to [Quick Merge Retirement Announcement](#).

## Learn QuickMerge Basics

- [About Conga QuickMerge](#)
- [Launching Conga Mail Merge from Conga QuickMerge](#)

## Install QuickMerge

- [Install Conga QuickMerge](#)

## Use QuickMerge

- [Use Conga QuickMerge to Automate Conga Mail Merge](#)
- [Use Conga QuickMerge to Automate Conga Composer \(Release 7\)](#)

## About Conga QuickMerge

Conga QuickMergeSM is an optional, free add-on to Conga Composer that helps automate Composer and Conga Mail Merge solutions by expanding the scope of how they can be launched.

## Automate Mass Mailings

- Predefine your template, the records to merge, and activity logging options.
- Bypass the Mail Merge wizard to quickly create letters, labels, and envelopes.

## Flexible Organization-Wide Reporting

- Create documents, reports, and presentations that are not associated with a specific, individual record in Salesforce.
- Leverage all the standard functionality of Composer.

## Easily Control Access & Permissions

- Exercise granular control over access to Mail Merge solutions.



# Installing Conga QuickMerge

The following steps provide an overview of how to install Conga QuickMerge and the general steps involved in deploying the application.

## To install Conga QuickMerge into a Salesforce instance:

1. Click the appropriate link to install Conga QuickMerge in your Salesforce environment: Conga QuickMerge is an add-on for Release 7 of Conga Composer and Conga Mail Merge. With Release 8.8 and higher of Conga Composer, this functionality is automatically included in the Conga Composer package in the Conga Global Merges object. If you want to install an earlier version of Conga QuickMerge, please contact a Conga Business Analyst.  
Follow the on-screen prompts to install Conga QuickMerge. (You must provide system administrator credentials to install an application from the AppExchange.) Be sure to Deploy the object at the end of the installation.
2. Set the default access to the object. Remind me of the path.
3. In Salesforce, navigate to Setup > Security Controls > Sharing Settings.
4. Click **Edit**.
5. Set the Conga Query object to Public Read Only.
6. Click **Save**.
7. Set the page layouts for each record type. Remind me of the path.
8. In Salesforce, navigate to Setup > Create > Objects > QuickMerge Links.
9. Scroll to the Record Types list and click Page Layout Assignment.
10. Click **Edit Assignment**.
11. In the PointMerge Record Types column, select all layout entries for the profiles. To select all, click the first cell, scroll to the last cell, and shift-click the last cell.
12. In the Page Layout to Use box, select PointMerge Layout.
13. Click **Save**.

The installation is complete. You're now ready to build a QuickMerge record.

## Launching Conga Mail Merge from Conga QuickMerge

With QuickMerge, you can streamline selections made in the Conga Mail Merge wizard, or even bypass the wizard completely.

Conga QuickMerge is a custom object available through a private AppExchange listing, used as a launchpad for Conga Mail Merge operations. With QuickMerge, you can streamline selections made in the Conga Mail Merge wizard, or even bypass the wizard completely. In many cases, you can download batches of documents in as few as two clicks.

Install QuickMerge into your Salesforce environment and add the custom tab to the Conga Composer application. Then create a QuickMerge Link record that pre-defines the choices that correspond to the Conga Mail Merge wizard, including a ReportId or QueryId, TemplateIds, and output parameters. From there, launch Conga Mail Merge by clicking the Launch button, and immediately download the results.

## Use Conga QuickMerge to Automate Conga Composer (Release 7)

The Conga Suite makes it easy to generate documents of all kinds, by merging templates with your Salesforce data. Conga QuickMerge automates both Conga Mail Merge and Conga Composer operations. By documenting your preferences in a QuickMerge button, you can launch your merge operations with a single mouse click.

### To create a QuickMerge Link record:

1. Select the QuickMerge Links tab.
2. Select PointMerge as the record type.
3. Click Continue to go to the next page.
4. Document your Conga Composer preferences.
5. Add the title of the QuickMerge action.
  - Link Type - You can perform PointMerge (Conga Composer) or MassMerge (Conga Mail Merge) operations. Select PointMerge.
  - The Salesforce ID of the report(s) you are using as a data source. This is the last 15 characters of the URL for that Report.

- The Salesforce ID of the template you are using. This is the last 15 characters of the URL for that template.
6. Launch the QuickMerge link by using one of these methods:
    - Click the Launch button on the object's detail page.
    - Click the Launch button on the QuickMerge Links home page.

When you run QuickMerge, it takes you directly to the Conga Composer dialog box to generate your documents. The parameters that you entered in the QuickMerge link are applied to your solution.

## Use Conga QuickMerge to Automate Conga Mail Merge

The Conga Suite makes it easy to generate documents of all kinds, by merging templates with your Salesforce data. Conga QuickMerge automates both Conga Mail Merge and Conga Composer operations.

By documenting your preferences in a QuickMerge button, you can launch your merge operations with a single mouse click.

### To create a QuickMerge Link Record for Conga Mail Merge:

1. Select the QuickMerge Links tab.
2. Select MassMerge as the record type.
3. Press Continue to go to the next page.
4. In the Title field, type the name of the QuickMerge link.
5. In the Link Type field, select MassMerge.  
You can perform PointMerge (Conga Composer) or MassMerge (Conga Mail Merge) operations.
6. In the MassMerge Settings area, complete the appropriate fields.
  - a. In the Template ID field, enter the Salesforce ID of the template you are using.
  - b. In the Report ID field, enter the Salesforce ID of the report you are using as a datasource.
  - c. (Optional) If you are using a custom template for the label or envelope type the Salesforce IDs in the Envelope Template ID.
  - d. You can choose to go through each step of the wizard or bypass it. In your usual usage, you will bypass the wizard although you may expose the steps during testing.
7. In the Output Settings area:

- Default to PDF
  - Force the output as PDF
8. In the Log Activity Settings area, specify the settings you want to use to log the merge action.
  9. Click **Save**.
  10. Document your preferences for the Mail Merge operation on the QuickMerge page.
    - The name of the QuickMerge action (MassMerge).
    - Link Type - you can perform PointMerge (Conga Composer) or MassMerge (Conga Mail Merge) operations. Select MassMerge.
    - The Salesforce ID of the template you are using. This is the last 15 characters of the url for that template.
    - The Salesforce ID of the report you are using as a datasource. This is the last 15 characters of the url for that Report.
    - If you are using a custom template for the label or envelope store those IDs here.
    - You can choose to go through each step of the wizard or bypass it. In your usual usage, you will bypass the wizard although you may expose the steps during testing.
    - Default the output to PDF.
    - Force the output to always be PDF.
    - Select your activity logging settings.

Once you have completed the setup you can now run QuickMerge. For each QuickMerge you add, a Launch button is created on the QuickMerge Links home page.

11. To run the QuickMerge Link for Conga Mail Merge, click the appropriate QuickMerge Launch button.

QuickMerge takes you directly to the Conga Mail Merge download page to generate your documents.

# Composer for Users

This section describes how *Conga Composer Solution* works and how to generate the document.

Topic	Description
What's Covered	This documentation walks the end-users through conceptual information, set-by-step instructions, use cases.
Primary Audience	Composer End Users (such as Sales Manager, Sales Representative)
IT Environment	Refer to the latest <i>Conga Composer Release Notes</i> for information on System Requirements and Supported Platforms.
Updates	For a comprehensive list of updates to this documentation for each release, see <a href="#">What's New in Composer Documentation</a> .
Other Resources	Refer to <i>Composer Release Notes</i> for information on new features and enhancements, fixed issues, and known issues for a specific release.

Before using Composer, you must be familiar with the following:

- Basic Salesforce administration
- Salesforce and Conga terms and definitions

Select one of the following topics for more information:

- [Generating a Document](#)
  - [Using Composer Button](#)
  - [Using Composer Lightning Component](#)
  - [Using a Local Template](#)
- [Using Live Edit](#)
- [Using Composer Advanced Features](#)
  - [Send a Document as a Link](#)
  - [Email Notification](#)
  - [View Document History](#)
  - [PDF Viewer](#)
- [Launching Conga Batch from a List View Button](#)

- [Localization of Classic Composer UI](#)
- [Localization of Lightning Component UI](#)

## Generating a Document

Select one of the following topics as per your configuration to generate the document:

- [Using Composer Button](#)
- [Using Composer Lightning Component](#)
- [Using a Local Template](#)

## Using Composer Button

This topic describes the use case for generating a document using the Composer Button.

This use case gives one example of how to generate an *Account Brief* document using the Composer Button. You might use this functionality differently to generate different documents depending on your business case. Contact your administrator to configure the Composer Solutions as per your business use case.

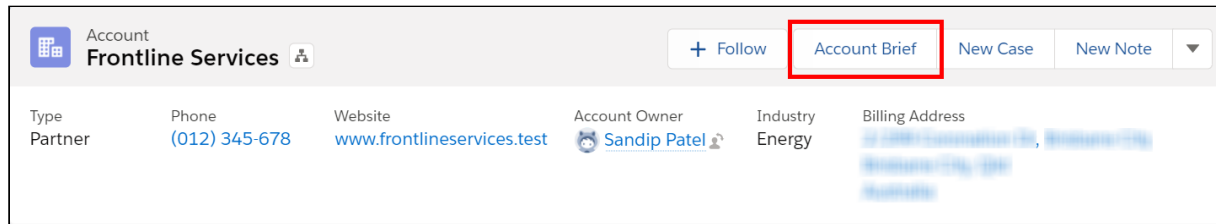
For example, if the administrator would like to enable you (as a sales representative) to generate account briefs based on specific fields from an account record, the administrator builds the Composer Solution and place the button on an account object's page layout. You then click the button to run the solution and generate the document.

The administrator can configure the solution in the following two ways as per the business use case to **generate the document in PDF format**:

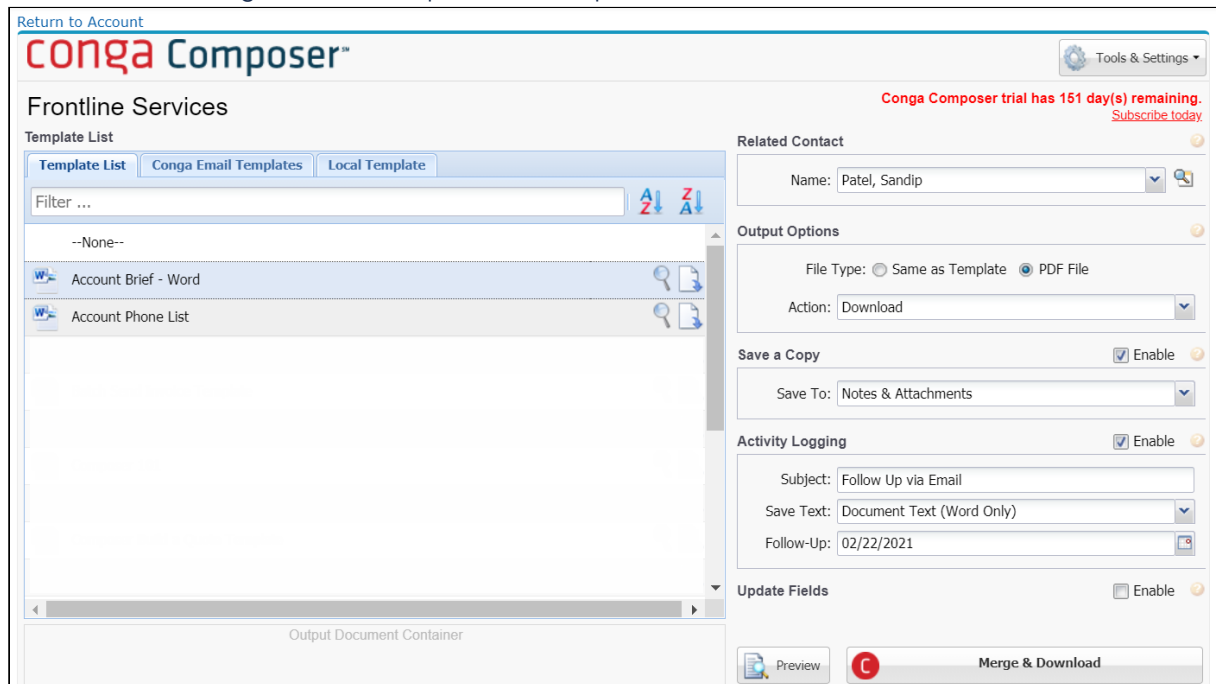
1. Provide selections, such as which template to use for the final merged files or other options to streamline the process.
2. Run the solution in background mode, which means with a single button click, the document will be generated according to the predetermined output method, behind the scenes. (For this, you only need to click the button, the rest of the process runs in the background and the document or email template is automatically merged and delivered in the desired manner. Because the user interface is skipped, you are not able to choose templates, output file options, or activity logging options. Instead, the administrator pre-defines these options in the Conga Composer solution using parameters.)

## To generate a document with Conga Composer User Interface

1. Click the **App Icon** (☰) > Search and select **Accounts**.
2. Click the respective account from the account list.
3. Click the **Account Brief** button.



4. The *Conga Composer User Interface* screen is displayed if the administrator defines this behavior for you to select particular options.




5. Select the respective **Template** from the list or it is preselected if the Composer Administrator has defined a particulate template as a default in the solution.

⚠ You can also upload a local template for the merge process. For more information, refer to the [Using a Local Template](#) section.

6. If required, select or update the options in the following sections:
  - Related Contact
  - Output Options
  - Save a Copy

- Activity Logging
  - Update Fields
7. Click **Preview** if you want to preview the document before the final merge.
  8. Click **Merge & Download**.

 For the **generate a document and send it via email** use case, refer to the [Using Composer Lightning Component](#) section.

The Account Brief document is downloaded into your system. If you have set the **Activity Logging** in the *Conga Composer User Interface* or *configured it into the solution by the administrator using Activity Logging parameters*, you can see the **Upcoming & Overdue** details under the *Activity* section as shown in the image below.



The screenshot displays the 'Activity' tab of a user interface. At the top, there are two tabs: 'Activity' (selected) and 'Chatter'. Below the tabs is a navigation bar with buttons for 'New Task', 'Log a Call', 'New Event', and 'Email'. A text input field labeled 'Create a task...' is followed by a blue 'Add' button. Below this is a filter section showing 'Filters: All time · All activities · All types' with a dropdown arrow, and links for 'Refresh', 'Expand All', and 'View All'. The main content area is divided into sections: 'Upcoming & Overdue' (with a dropdown arrow), 'February · 2021' (with a dropdown arrow and 'This Month' label), and 'December · 2020' (with a dropdown arrow and 'Last Year' label). Each section contains a list of tasks with expandable details (indicated by a right-pointing chevron and a green task icon), a task title, a description, and a date with a dropdown arrow. The tasks in the 'Upcoming & Overdue' section are 'Follow-Up:New Customer Checkin' with dates 'Dec 7, 2020' and 'Dec 6, 2020'. The 'February · 2021' section has one task: 'Follow Up via Email' with the date 'Today'. The 'December · 2020' section has three tasks: 'New Customer Checkin' with dates 'Dec 29, 2020', 'Dec 2, 2020', and 'Dec 1, 2020'. A 'View More' button is located below the 'Upcoming & Overdue' section. At the bottom, it says 'No more past activities to load.'

## Using Composer Lightning Component

This topic describes the use case for generating a document using the Composer Lightning Component.

This use case gives one example of how to generate a *Client Welcome Letter* using the *Composer Lightning Component*. You might use this functionality differently to generate different documents depending on your business case. Contact your administrator to configure the Composer Solutions as per your business use case.

For example, you (sales representative) are sending out new client welcome letters to every customer. These are often the same copy and pasted documents with some slight changes for personalization. The administrator (Sales Operations Manager) of your company always looking for ways to speed up and simplify the sales process, so he automates this process and builds a *Composer Solution* so that you can generate a welcome letter for new accounts directly from the Salesforce using Composer Lightning Component.

The administrator can configure the solution in the following two ways as per the business use case to **generate the document** and **send it via email**:

1. Provide selections, such as which template to use for the final merged files or other options to streamline the process.
2. Run the solution in background mode, which means with a single button click, the document will be generated according to the predetermined output method, behind the scenes. (For this, you only need to click the button, the rest of the process runs in the background and the document or email template is automatically merged and delivered in the desired manner. Because the user interface is skipped, you are not able to choose templates, output file options, or activity logging options. Instead, the administrator pre-defines these options in the Conga Composer solution using parameters.)

## To generate a document with Conga Composer Lightning Component UI

1. Click the **App Icon** (☰) > Search and select **Accounts**.
2. Click the respective account from the account list.

- Click the **Client Welcome Letter** button.

The screenshot shows the Salesforce account page for 'Frontline Services'. The account owner is Sandip Patel. The 'Conga Document Generation' sidebar is on the right, and the 'Client Welcome Letter' button is highlighted with a red box.

- The *Conga Composer Lightning Component UI* screen is displayed if the administrator defines this behavior for you to select particular options.

The screenshot shows the Conga Document Generation UI. The 'Client Welcome Letter' option is selected in the 'Email Templates' section. The 'DETAILS' section includes a search for related contacts, a PDF file selection, and an output action of 'Email'.

- Select the respective **Email Template** from the list or it is preselected if the Composer Administrator has defined a particulate template as a default in the solution.

6. If required, select or update the options in the following sections:
  - Related Contact
  - Output Options
  - Save a Copy
  - Activity Logging
7. Click **Preview Solution** if you want to preview the document before the final merge operation.
8. Click **Merge and Email**. The *Conga Email User Interface* or standard *Salesforce Email Task* screen is displayed with an attachment of the generated document as per the configuration.

You can customize the formatting and delivery options before sending the email. When an outbound email is generated from a Conga HTML Email Template, SendGrid is enabled, or when merging multiple documents, Composer automatically assembles the email and presents the *Conga Email User Interface*. If none of those conditions are met, the standard *Salesforce Email Task* screen is used when emailing a Composer generated document.
9. Click **Send Email**.

The email is sent with the Welcome Letter. If you have set the **Activity Logging** in the *Conga Composer Lightning Component UI* or configured it into the solution by the administrator using *Activity Logging* parameters, you can see the **Upcoming & Overdue** details under the *Activity* section as shown in the image below.

The screenshot displays the Salesforce Activity page. At the top, there are tabs for 'Activity' (selected) and 'Chatter'. Below the tabs is a navigation bar with 'New Task', 'Log a Call', 'New Event', and 'Email'. A search bar contains the text 'Create a task...' and an 'Add' button. Below this, there are filters: 'Filters: All time • All activities • All types' with a dropdown arrow, and links for 'Refresh', 'Expand All', and 'View All'. The main content is organized into sections:

- Upcoming & Overdue:** Contains two tasks, both titled 'Follow-Up:New Customer Checkin'. The first is due on 'Dec 7, 2020' and the second on 'Dec 6, 2020'. Each task includes a description: 'You have an upcoming task'. A 'View More' button is located below these tasks.
- February • 2021:** Labeled 'This Month'. Contains one task titled 'Follow Up via Email' due 'Today'. The description is 'You had a task with Sandip Patel'.
- December • 2020:** Labeled 'Last Year'. Contains three tasks, all titled 'New Customer Checkin'. The dates are 'Dec 29, 2020', 'Dec 2, 2020', and 'Dec 1, 2020'. Each task has the description 'You had a task'.

At the bottom of the December 2020 section, it says 'No more past activities to load.'

## Using a Local Template

By default and as a recommended practice, only users with the Salesforce System Administrator profile are permitted to upload a template from their local hard drives. This feature is always enabled for administrators because of its utility in creating and testing templates. The Composer Administrator may enable this feature for all other user profiles as per your business use case. You can then upload your local template to merge and download/email the document.

## To use a Local Template

1. Click the *Composer Solution Button*.
2. Click the **Local Template** tab.
3. Click **Select File** and upload the template.
4. Click **Preview** if you want to preview the document before the final merge.
5. Click **Merge & Download** or **Merge & Email** as per the composer solution configured.

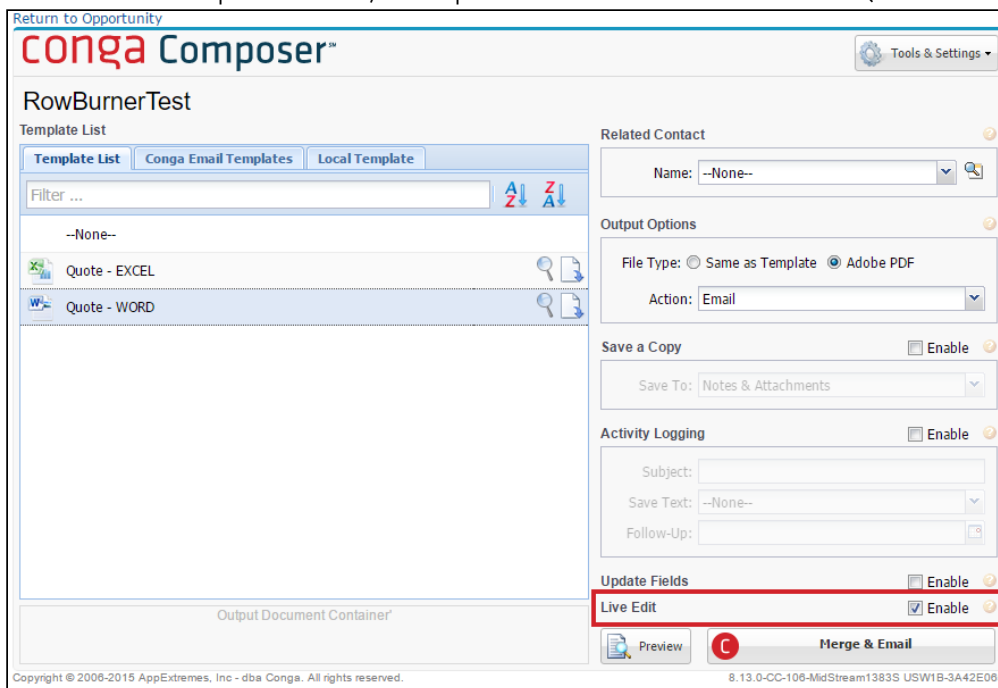
For more information, refer [Using Composer Button](#) and [Using Composer Lightning Component](#) sections.

## Using Live Edit

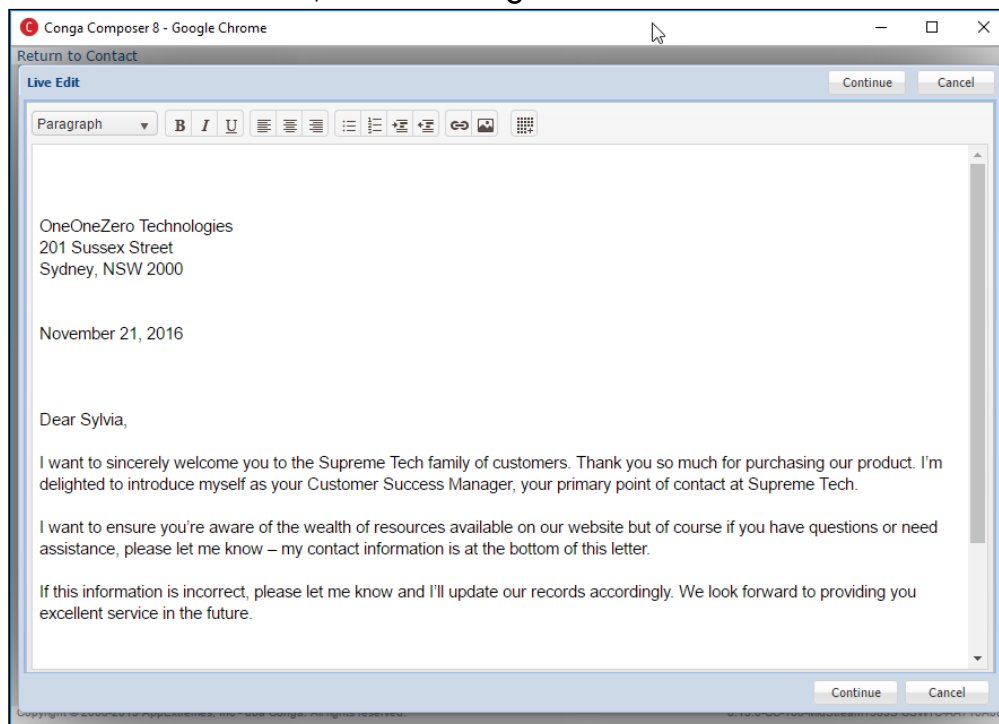
Live Edit allows you to preview and edit your document before it is delivered according to the specified method like email and download.


## To use Live Edit

1. Click the *Composer Solution* button to launch your solution. When the Conga Composer user interface loads, you will see an option for Live Edit. If you used the LiveEditEnable parameter, the option will be checked as well (see screenshot below).



2. With Live Edit checked, click the **Merge** button. The *Live Edit* window launches.



 Some templates do not display accurately in the Live Edit window. Trigger a full merge to see the final results.

In the Live Edit window, you can make changes to your document. The toolbar (shown below) allows you to make formatting changes, including adding links, images, and tables. Be aware that formulas and other data have already been calculated when using Live Edit, so edits to calculations need to be made in all applicable locations. For example, changing a value in a list does not automatically update the total amount; that value needs to be updated manually as well.



3. After making changes, click the **Continue** button. The merge process and deliver the document finalizes.

## Considerations

The following limitations apply:

- Microsoft Word templates are the only supported template type.
- If a non-Word template is predefined in the URL or selected in the interface, the UI presents a warning message and disables the Live Edit checkbox.
- Only single Word templates are supported.

- The template must be saved in Salesforce (local templates do not work).
- Text boxes are not supported.
- Resized images may not carry over into the output document as expected.
- Some advanced formatting might not carry over into the output document as expected.
- There is no "write back" functionality between Salesforce and Live Edit. For example, if you change information like a phone number or name in Live Edit that same information still needs to be updated in Salesforce in order to match.
- Headers are not supported with Live Edit.
- Live Edit might negatively impact Watermarks.
- When IF statements are present in the Word template, it is advised to use the parameter, `wordStripIfs=1`.

Live Edit is a simple copy/paste of the content out of the Word document, edited in Rich Text, and pasted back to the Word document.

#### **Important**

- Live Edit does not support all Composer processes such as headers, footers, watermarks, or background colors. For example, headers and footers do not display in the Live Edit window and consequentially do not display in the final merged document. Live edit does not support multiple templates in a Composer solution.
- The Live Edit feature is not supported for the Composer Lightning Component.

## Using Composer Advanced Features

Composer Advanced Features is a new functionality that provides users with access to Composer Link Service, Notification Emails, Document History, and the Conga PDF Viewer.

- [Send a Document as a Link](#)
- [Email Notification](#)
- [View Document History](#)
- [PDF Viewer](#)

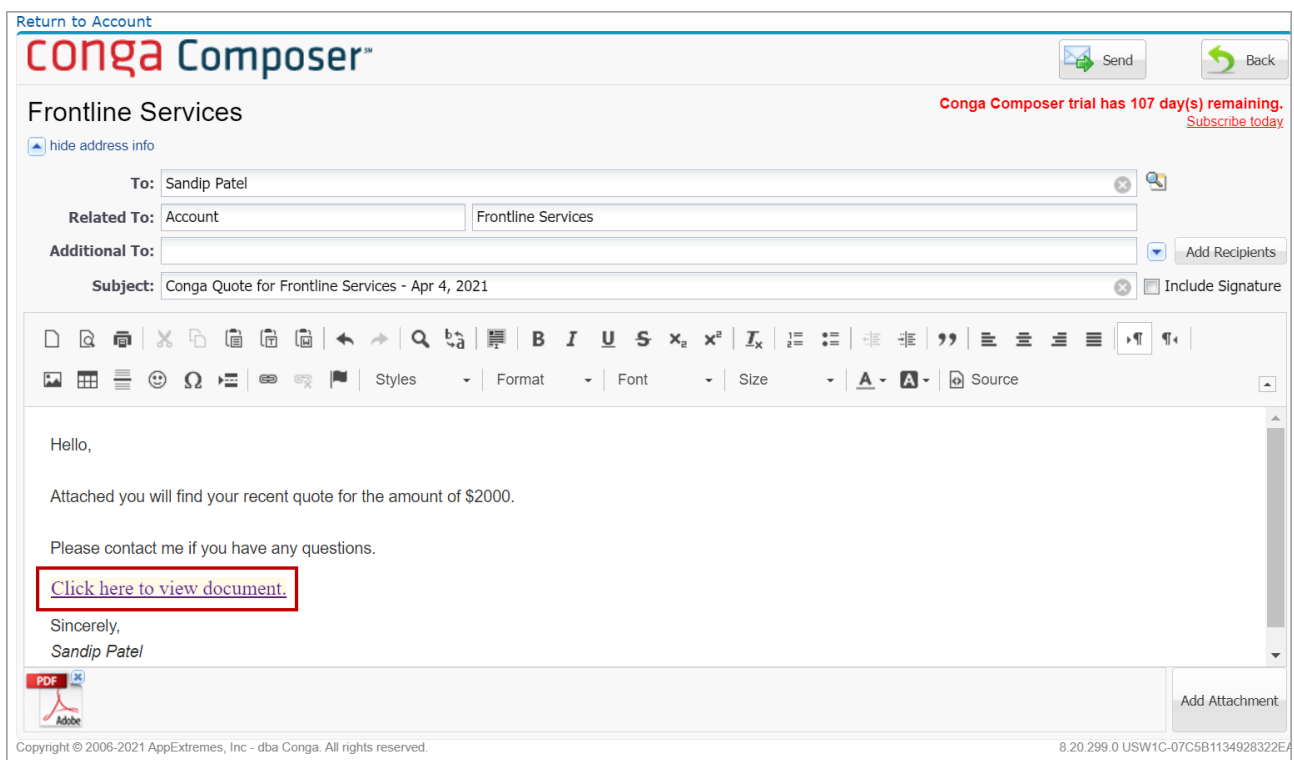


## Send a Document as a Link

Composer Link Service is a Composer Advanced Feature that allows end-users to create and send documents through email. Composer solutions using Composer Link Service create a unique link for each document, making documents easily shareable and accessible. Word, Excel, and PowerPoint files sent with Composer Link Service are downloaded to the browser tray once the recipient clicks the document link in the email. PDF files are automatically displayed in the Conga PDF Viewer.

⚠️ Composer Link Service supports sending PDF, Word, Excel, and PowerPoint files. The Conga PDF Viewer only supports PDF files.

Refer to the following image to see what the generated link looks like:



The Composer document link is kept on the [Document History](#) object in Salesforce. Each online document generated with Composer Link Service has its own Document History record.

## Email Notification

⚠️ Contact your Administrator to enable the Email Notification feature.

Notification Emails are emails sent to the running Composer user when recipients interact with documents sent as a link. A Notification Email will be sent to the Composer user who executes and sends the document. When a document is opened for the first time, a Notification Email is sent to the user. After the initial notification email, no additional notification emails are sent to the user.

You can use this feature to track the progress of your task and ensure that it is completed on time, depending on your business use case.

## View Document History

The **Document History** object is a custom object that contains the link to a document generated with Composer Link Service. Composer automatically creates a unique Document History record specific to each document. Each Document History record is generated with a related Document History Detail child record.

The **Document History Detail** object is a child object of the Document History object. It contains key information such as the number of document views and the date/time that the document was last viewed.

## To view Document History


1. Click the **App Icon** (⋮) > Search and select **Document Histories**.
2. Click the respective document history name hyperlink to view the details.


The screenshot shows a 'Document History' record with ID 'DH-0002'. It features a 'Details' tab and various fields including 'Document History Name', 'Owner' (Mark), 'Conga Solution' (Linkservice), 'URL', 'Link Text', 'Last Viewed' (4/4/2021, 10:27 PM), 'Master Object ID', 'Number of Views' (2), 'Created By' (Mark, 4/1/2021, 9:38 AM), and 'Last Modified By' (Mark, 4/4/2021, 10:28 PM).

Document History	
DH-0002	
Document History Name	DH-0002
Owner	<a href="#">Mark</a>
Conga Solution	<a href="#">Linkservice</a>
URL	<a href="https://content.congaplatform.com/PdfViewer/1.0.0/index.html?linkId=8cadbe4d-9ffd-4a3d-8da0-27293b938fa7&amp;signature=982cd5">https://content.congaplatform.com/PdfViewer/1.0.0/index.html?linkId=8cadbe4d-9ffd-4a3d-8da0-27293b938fa7&amp;signature=982cd5</a>
Link Text	Click here to view document.
Last Viewed	4/4/2021, 10:27 PM
Master Object ID	0014x000005ctKwAAI
Number of Views	2
Created By	<a href="#">Mark</a> , 4/1/2021, 9:38 AM
Last Modified By	<a href="#">Mark</a> , 4/4/2021, 10:28 PM

Field	Description
Document History Name	Auto-generated record name, such as <i>DH-0001</i> .
Conga Solution	Lookup field that displays a link to the related Conga Solution that generated the document.
URL	Displays the click-able hyperlink to the document.
Link Text	Displays the hyperlink text used in the Notification Email Template to access the document. By default, Link Text is set to <i>Click here to view document</i> .
Last Viewed	Displays the date/time for when the document was last viewed.
Master Object ID	Displays the specific record ID that the document is generated from.
Number of Views	Displays the number of times the document is viewed.
Created By	Displays the user that created the Document History record and the date/time of the record creation.
Owner	Displays the current record owner.
Last Modified By	Displays the user that last modified the Document History record and the date/time of the record modification.

## To view Document History Details

1. Click the **App Icon** (  ) > Search and select **Document Histories**.
2. Click the respective document history name hyperlink to view the details.
3. Click the **Related** tab.
4. Click the respective document history detail name hyperlink to view the details.

 Document History Detail  
**DHD-0002**


Related
Details

Document History Detail Name  
DHD-0002


---

Document History  
[DH-0002](#)


---

Document Status  
**Generated** 


---

Email Address  
[REDACTED] 


---

SMS Number  
[REDACTED] 


---


Number of Views  
3 

---

Last Viewed  
4/4/2021, 11:11 PM 

---

Created By  
 [Mark](#), 4/1/2021, 9:38 AM

Last Modified By  
 [Mark](#), 4/4/2021, 10:28 PM

Field	Description
Document History Detail Name	Auto-generated record name, such as <i>DHD-0001</i> .
Document History	Lookup field to the related parent Document History record.
Document Status	Lookup field that displays a link to the related Conga Solution that generated the document.
Email Address	Displays the email address the document is sent to.
Number of Views	Displays the number of times the document is viewed.
Last Viewed	Displays the date/time for when the document was last viewed.

## PDF Viewer

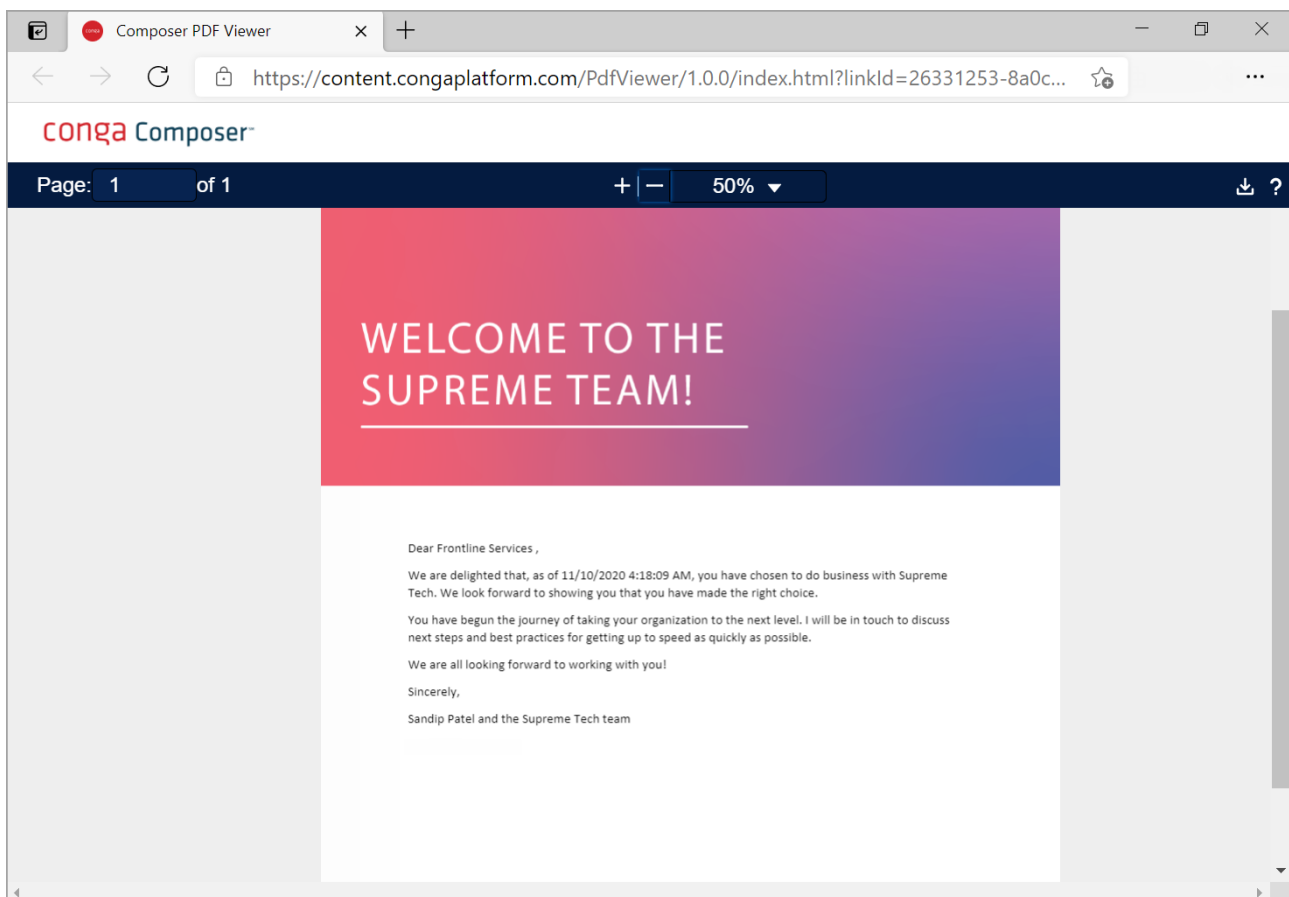
The Conga PDF Viewer makes it simple to view documents created with Composer Link Service for both you and the recipient. When you click the document link in an email sent from Composer Link Service, it automatically opens in the Conga PDF Viewer. Salesforce

users can also use the Conga PDF Viewer to view online documents by clicking the hyperlink in the URL field on the [Document History](#) object.

**⚠ The Conga PDF Viewer only works with PDF files.**

You can easily zoom in and out by percentage or page fit, skip to specific pages within the document, and download the file locally with the Conga PDF Viewer.

Refer to the following image to see how the generated document opens in the Conga PDF Viewer:



## Launching Conga Batch from a List View Button

Launching your Conga Batch solution from a Salesforce List View button allows you to choose the records in your batch on an ad hoc basis rather than using a report's or query's filter criteria to determine which records are included.

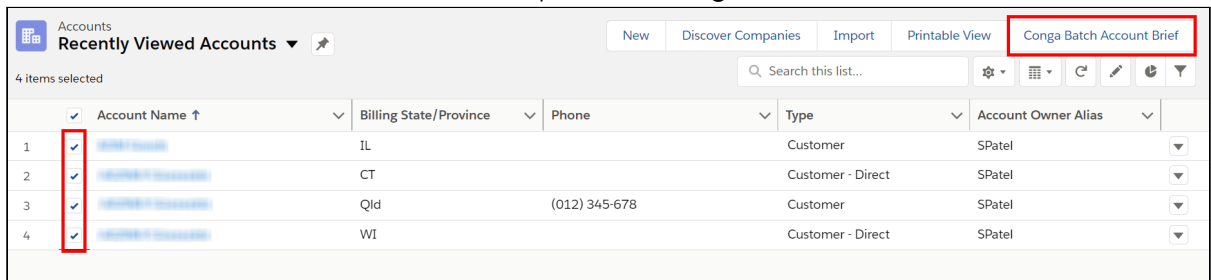
The benefits of using a List View button include:

- Easily choose which records you want to use as the Master Object records in the Conga Batch batch by selecting them manually from the list.
- Select up to 75 records per Conga Batch batch.
- Optionally, you can also use a Related List Button on an object’s page layout.

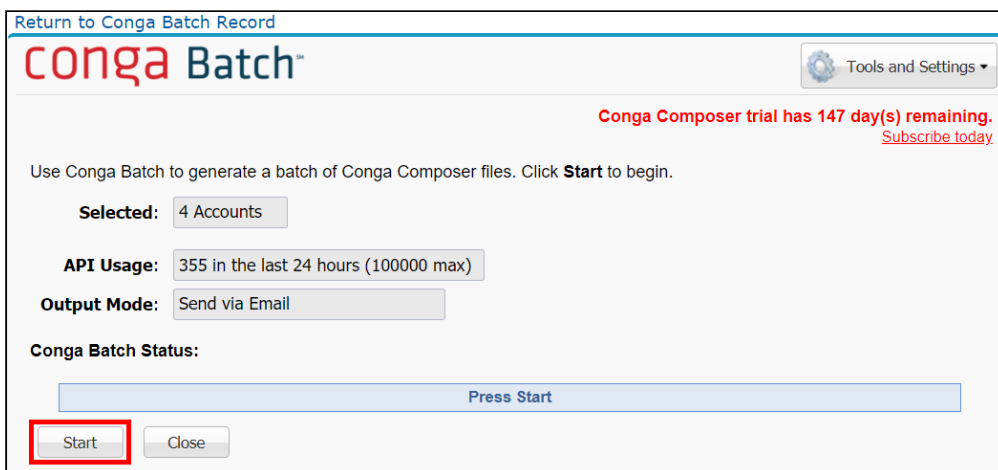
**i** The maximum amount of records you can select and merge with a Conga Batch List View button is 75 records. Use a scheduled Conga Batch solution to merge more than 75 records at once. For more information on Conga Batch, contact your Composer Administrator.

## To launch Conga Batch from a List View Button

1. Click the **App Icon** (☰) > Search and select a respective object where the List View button is configured. In this example, it is configured on the **Accounts** object.
2. Select the records for which you want to run the batch.
3. Click the *List View Button*. In this example, it is **Conga Batch Account Brief**.



You are redirected to the *Conga Batch User Interface* screen. Click **Start** to start the merge process.

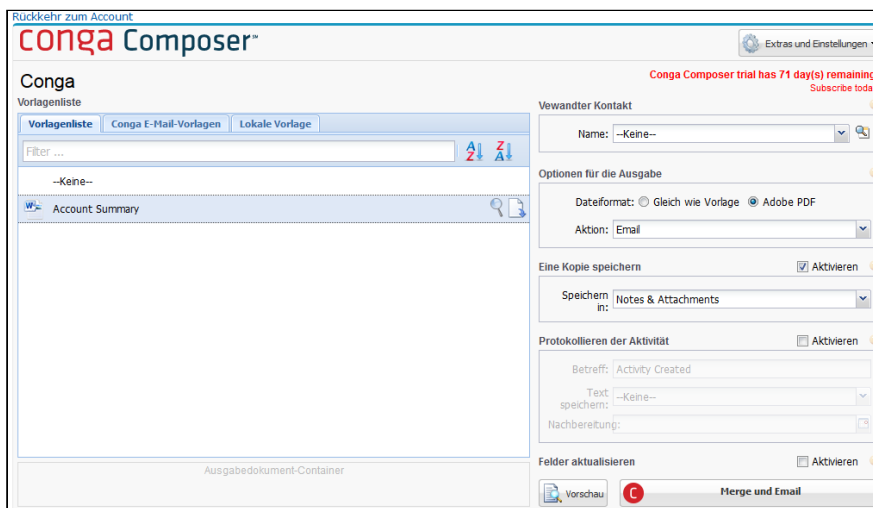


# Localization of Classic Composer UI

This documentation review how to enable German and French dynamic translations in the Composer user interface.

⚠ You must have Conga Composer version 8.20.71 or higher installed in your Salesforce org to leverage German, French, and Polish dynamic translations of the Composer user interface.

Conga Composer's user interface currently supports dynamic translation for German, French, and Polish, in addition to the English language default.

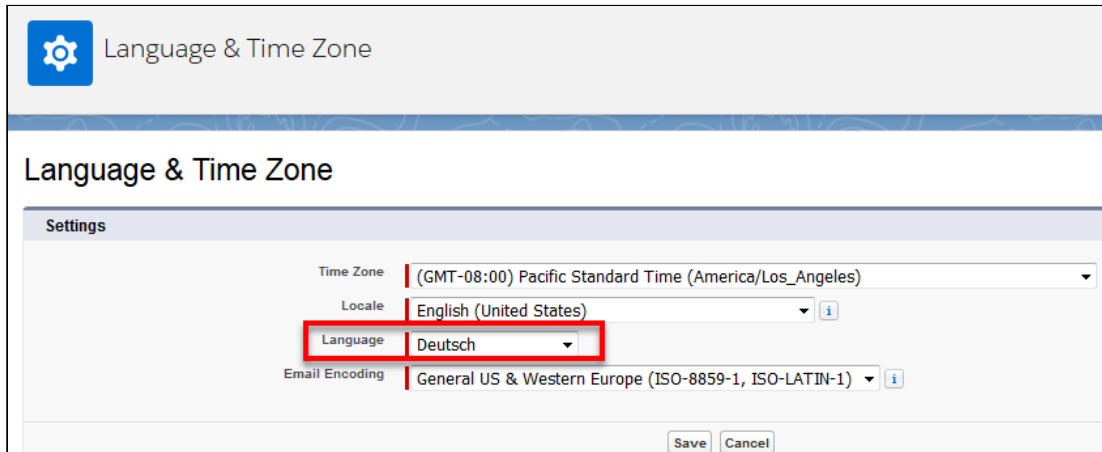


The Composer user interface automatically translates to German or French for Salesforce users that have their local language in Salesforce set to German, French, or Polish.

## To enable dynamic translation in the Composer user interface

1. Under your Salesforce Profile, select **Settings**.

2. Click **Languages & Time Zone**.



3. Change the Language field to Deutsch (German), Français (French), or Polski (Polish).
4. Click **Save**.

## Localization of Lightning Component UI

**Conga Composer Lightning Component** user interface supports dynamic translation for German, French, Dutch, Polish, and Spanish, in addition to English as a default language. It automatically translates to German, French, Dutch, Polish, or Spanish for the Salesforce Users based on their local language set in Salesforce.

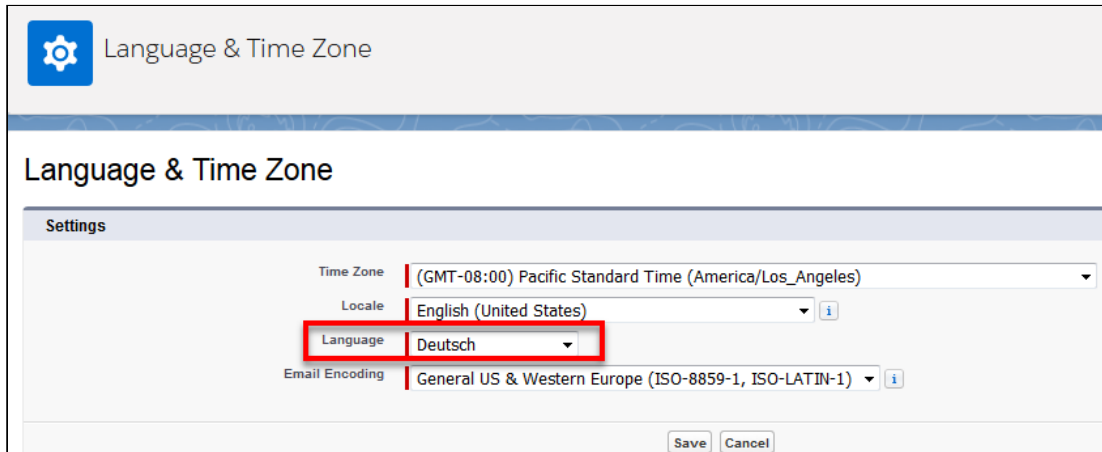
⚠ You must have a minimum Conga Composer version 8.198-Longs installed in your Salesforce org to leverage German, French, Dutch, Polish, and Spanish dynamic translations.

To set dynamic translation

1. Under your Salesforce Profile, select **Settings**.



2. Click **Languages & Time Zone**.



The screenshot shows the 'Language & Time Zone' settings page. The page title is 'Language & Time Zone'. Below the title, there is a 'Settings' section with four dropdown menus: 'Time Zone' (set to '(GMT-08:00) Pacific Standard Time (America/Los\_Angeles)'), 'Locale' (set to 'English (United States)'), 'Language' (set to 'Deutsch', highlighted with a red box), and 'Email Encoding' (set to 'General US & Western Europe (ISO-8859-1, ISO-LATIN-1)'). At the bottom right, there are 'Save' and 'Cancel' buttons.

3. Change the Language field to Deutsch (German), Français (French), Nederlands (Dutch), Polski (Polish), or Español (Spanish).
4. Click **Save**.

# Composer Features by Release

Review the latest Composer Features by Release document.

- [Features by Release](#)

## Features by Release

This document contains an overview of features introduced in each major release of Conga Composer. For more information, see [Conga Composer Features by Release](#).

# Community & Learning Center Resources

The Conga Customer Community is your one-stop shop for success!

After registering as a new member you'll gain access to a wide variety of resources, including: further details regarding the onboarding process, access to the knowledge base, and product-specific discussion groups.

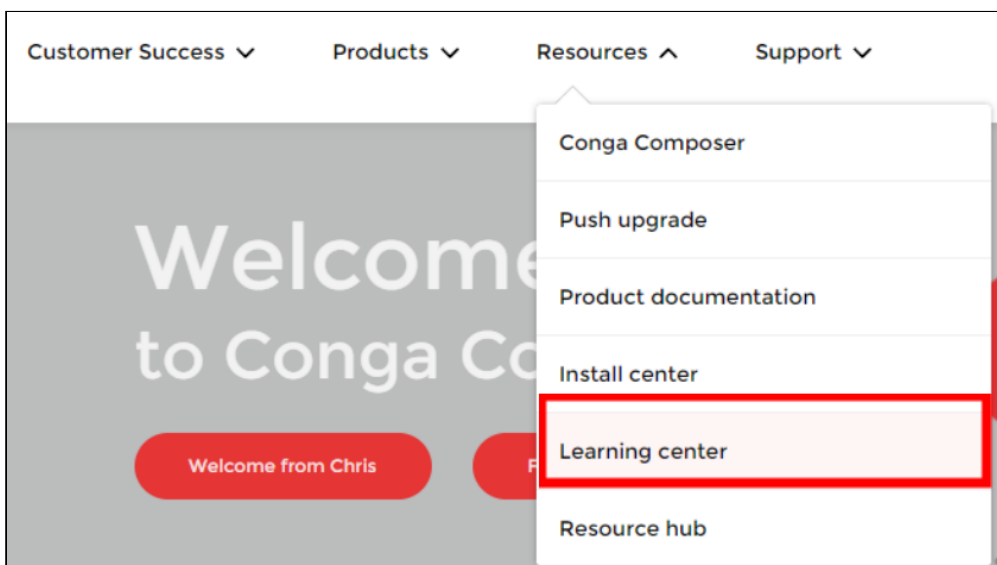
In addition, just a click away from Community is the Conga Learning Center: our comprehensive training portal well stocked with FREE self-paced courses complimented by virtual instructor-led sessions for more advanced and intensive training!

## Ready to get started?

First download the [Salesforce Authenticator app](#); you'll need it as part of the login procedure.

Then create a Community account, or log in [here](#)!

Once you're logged in, navigate to the Resources dropdown menu and click Learning Center to reach the [Conga Learning Center](#)!



## **Conga Copyright Disclaimer**

Copyright © 2022 Apttus Corporation (“Conga”) and/or its affiliates. All rights reserved.

No part of this document, or any information linked to or referenced herein, may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written consent of Conga. All information contained herein is subject to change without notice and is not warranted to be error free.

This document may describe certain features and functionality of software that Conga makes available for use under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not, in any form, or by any means, use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part of the software. Reverse engineering, disassembly, decompilation of, or the creation of derivative work(s) from, the software is strictly prohibited. Additionally, this document may contain descriptions of software modules that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document.

U.S. GOVERNMENT END USERS: Conga software, including any operating system(s), integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Neither the software nor the documentation were developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Conga and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Apttus, AI Analyze, Conga, Conga AI, Conga AI Discover, Conga Batch, Conga Collaborate, Conga Composer, Conga Conductor, Conga Connect, Conga Courier, Conga Grid, Conga Mail Merge, Conga Merge, Conga Orchestrate, Conga Sign, Conga Trigger, Digital Document Transformation, True-Up, and X-Author are registered trademarks of Conga and/or its affiliates.

The documentation and/or software may provide links to web sites and access to content, products, and services from third parties. Conga is not responsible for the availability of, or any content provided by third parties. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Conga is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Conga is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

For additional resources and support, please visit <https://community.conga.com>.