Conga Grid

Table of Contents

Conga Grid enables Salesforce customers to manage, visualize, and navigate data more efficiently - and all from a single, actionable grid.

Release Notes

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

- Winter20.01.07 Release Notes
- Winter20.12.11 Release Notes
- Winter20.12.10 Release Notes
- Winter20.12.03 Release Notes
- Winter20.11.12 Release Notes
- Winter20.11.12.01 Release Notes
- Winter '20 Release Notes

Winter20.01.07 Release Notes

These Release Notes contain the following information about Conga Grid Winter20.01.07 Release.

> For more information on new features, enhancements, and document improvements refer to
>
> - What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	Latest Certified Version *(Version Name / Version Number)*
Conga Grid	3.10

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

There are no new enhancements in this release.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-943	When you use the Grid Batch Send Email feature, you are now only able to see and select templates that are marked as available for use.
AG-872	You can now create a Grid Application in Quickstart without installation errors.
AG-1056	Fixed an issue where, upon the first load of the application, if you added inline records and tabbed to continue to another field instead of clicking into them, an error occurred if the field was a combobox.
AG-1055	Enhanced picklist behavior to align with Salesforce's edit form.
AG-1048	Long text area fields now show red bars that indicate a required field in the column header.
AG-1026	Implemented a fix to resolve Date, DateTime, and Time CSV export errors.
AG-1053	When you create a new record from a grid within a grid that has multiple record types, the record type you select is now in the new record instead of reverting to the default.

Known Issues

There are no known issues in this release.

Winter20.12.11 Release Notes

These Release Notes contain the following information about Conga Grid Winter20.12.11 Release.

> For more information on new features, enhancements, and document improvements refer to
>
> - What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	Latest Certified Version *(Version Name / Version Number)*
Conga Grid	3.7

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

There are no new enhancements in this release.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-1049	Updated Conga Grid API from v48.0 to v50.0.

Known Issues

There are no known issues in this release.

Winter20.12.10 Release Notes

These Release Notes contain the following information about Conga Grid Winter20.12.10 Release.

For more information on new features, enhancements, and document improvements refer to

- What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	Latest Certified Version *(Version Name | Version Number)*
Conga Grid	3.5

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

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

Lightning Component Tab Labels

You can now define or edit labels in the Lightning Component tab. For more information, see Lightning Component Tab Labels.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-843	When you make changes in the reading pane, sticky views now remain after switching from one selected record to another.
AG-1047	Non-admin users can now edit field values, remove users from fields, type and search for a new user, and save records without issues.
AG-985	When you have a formula in the detail pane and save the view, it no longer disappears after a soft refresh.

Known Issues

There are no known issues in this release.

Winter20.12.03 Release Notes

These Release Notes contain the following information about Conga Grid Winter '20 Release.

For more information on new features, enhancements, and document improvements refer to

- What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	**Latest Certified Version** *(Version Name | Version Number)*
Conga Grid	3.4

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

There are no new enhancements in this release.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-1034	The default value field with the rich text editor no longer takes up the screen and prevents you from saving when you edit the column settings in a Grid for a rich text field.
AG-763	You are now able to edit a Parental Multi-Picklist field.
AG-893	Filtering Parental and Grandparental Multi-Picklists now returns results.
AG-1043	The Conga Grid View label now displays correctly in the Community.
AG-1041	Filtering a number greater than 10 billion no longer results in an error.
AG-1049	You can now show NewValue and OldValue fields in History Object within your Grid.
AG-1036	The summary no longer displays in a Grandchild Grid field summary when you have Do Not Sum selected for the field.

Known Issues

There are no known issues in this release.

Winter20.11.12 Release Notes

These Release Notes contain the following information about Conga Grid Winter '20 Release.

> For more information on new features, enhancements, and document improvements refer to
>
> - What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	Latest Certified Version *(Version Name / Version Number)*
Conga Grid	3.2

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer

System Requirement	Minimum Supported Version
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

There are no new enhancements in this release.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-1015	You are now able to fill down a value for a Parental Field column without errors in the console and your updated value is applied to all selected records.
AG-1021	The correct value now displays when you set a filter value for a checkbox field.
AG-1027	Searching for a valid user no longer fails when you have a filtered lookup in place that uses OR logic.

Conga Internal ID	Description
AG-1028	Importing data into a Grid view no longer fails if one of the columns has a rich text field.
AG-1029	Email addresses now display as clickable links when you access the Details Grid.

Known Issues

There are no known issues in this release.

Winter20.11.12.01 Release Notes

These Release Notes contain the following information about Conga Grid Winter '20 Release.

> For more information on new features, enhancements, and document improvements refer to
>
> - What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	Latest Certified Version *(Version Name | Version Number)*
Conga Grid	3.3

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

There are no new features in this release.

Enhancements

There are no new enhancements in this release.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-872	You no longer receive an error when creating a Grid Application in Quickstart.
AG-1033	Values no longer disappear when you filter a checkbox field and a Grid is reset.
AG-1023	You no longer receive errors or are blocked from a view when you access a view containing a child column with an object you do not have access to.
AG-1030	You are now able to fill down a value for a Parental Field column if a column is frozen.
AG-1031	Several security updates were made.

Known Issues

There are no known issues in this release.

Winter '20 Release Notes

These Release Notes contain the following information about Conga Grid Winter '20 Release.

For more information on new features, enhancements, and document improvements refer to

- What's New in the Conga Grid Winter '20 Guide

- 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 guides. The packages marked as **(New)** are new packages in this release.

Product	**Latest Certified Version** ***(Version Name | Version Number)***
Conga Grid	3.0

Product	Latest Certified Version *(Version Name | Version Number)*
Conga Approvals ⚠ This is only required to install the Approval Center.	11.3

System Requirements and Supported Platforms

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

System Requirement	Minimum Supported Version
Salesforce Edition	• Performance / Unlimited • Enterprise • Professional (limitations may exist) • Developer
Browser	• Firefox® (minimum supported version by Mozilla) • Chrome® (minimum supported version by Google) • Internet Explorer® (version 11 or the latest version supported by Salesforce) • Safari® (Mac only) (minimum supported version by Apple) • Microsoft Edge (minimum supported version by Microsoft)

Upgrade Notes

To upgrade Conga Grid, see Install or Upgrade Conga Grid.

New Features

The following features are new to Conga Grid in this release.

The Conga Grid Approval Center

The Conga Grid Approval Center is a dashboard integrated with Conga Intelligent Workflow Approvals. Here you can easily approve, reject, and manage your pending approvals. For information on how to access the Approval Center, see Installing and Setting up the Approval Center.

Enhancements

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

User Interface Updates

Several user interface updates were implemented. For more information, see Conga Grid User Interface Updates.

Lightning Email Templates

When you send an email from a template, you now have the option to select Lightning Email Templates. For more information, see Send Email.

Configuration Options

You can now reorder columns and reset column widths.

Conditional Formatting Legend

A conditional formatting legend is now available in your bottom toolbar to give you an overview of the different conditional formatting rules being applied to a view.

Infinite Scrolling

Infinite scrolling is now available as a feature security setting within Conga Grid setup. Enable it to change from paged viewing to infinite scroll.

Text Area Popover

Clicking into a text area field now opens a popover to allow users to see and edit the entire field more easily.

Field Preview

A preview tooltip is now available for text area fields in Grid Views that have editing disabled to allow users to read the full contents of the cell.

Disabled by Default

In Conga Grid Setup, features turned off by default are now labeled as Disabled by Default.

Resolved Issues

The following table lists the issues resolved in this release.

Conga Internal ID	Description
AG-878	When you fill out a Parental Multi-Picklist field, you are now able to select multiple values.
AG-898	You can now add a record in a new window while in Full Screen mode.
AG-904	Filtered Lookups now return valid values as results.
AG-942	When you batch add tasks to records, the fields now render correctly.
AG-955	You now have the option to leave a picklist field empty by selecting a blank value.
AG-958	When you clear a lookup field, it now correctly displays as blank instead of retaining the old value.

Conga Internal ID	Description
AG-959	When you export to CSV with record IDs, the columns no longer misalign and each lookup has an ID column created.
AG-960	When you edit a parental value in the Reading Pane, clicking out of the cell no longer shows the neighboring value instead of the inputted value.
AG-962	When you edit a parental field on a Grid, you are no longer blocked from clicking back into the field to make additional changes.
AG-975	When you freeze a column, the CSS is now correctly applied the same as when the column is unfrozen.
AG-976	When you reset the reading pane, any conditional formatting that was applied is no longer removed from the reading pane Grid.
AG-977	When Conga Grid times out, you now receive the following error: Unable to Connect to the Server (Transaction Aborted: Timeout) - Learn more here.
AG-979	Resolved an issue where you were unable to update date fields when there was a grandchild column referencing a name related key.
AG-980	Resolved an issue where apostrophes and single quotes in a View Name caused duplicate views to be created.
AG-1020	Resolved an issue where users couldn't add a new Related To record inline when a profile only had access to the Master Record Type while other record types existed for an object.

Known Issues

There are no known issues in this release.

About Grid

Conga Grid replaces Salesforce lists, related lists, and reports with actionable grids to dramatically increase user productivity while improving the user experience and user adoption rates. Its powerful interface enables users to understand and digest their data quickly, to filter and group, to see relationships, and to take action quickly.

Conga Grid offers another level of Salesforce data analysis with dynamic view options like conditional formatting, sorting, mass updates, and other Excel-like features to help capture and accelerate execution on Salesforce-centric activities.

Conga Grid allows users to perform the following tasks:

- Manage Permissions and Licenses
    - Assign permissions, licenses, and sharing settings
    - Configure import and export settings
    - Configure setup options
- Create and Export Grid Graphs
- Configure Reading Pane
    - Add custom Lighting Components
- Build Solutions
    - Configure Grid tabs, lists, and buttons
    - Configure Inline Options
- Create a Multi-Tabular Interface
    - Configure automatic, manual, stacked, and Visualforce markup lists
- Create Conga Grid Actions
    - Batch Add, Clone, and Export Actions
    - Integrate with Conga Composer and Conga Batch
    - Send Emails
    - Create Custom Actions
- Create Conga Grid Views
    - Manage, create, and save views
    - Manage privacy settings for views
    - Create sticky views
- Use Code Examples for Custom Actions

What's New in Grid Documentation

Document	Topic	Description
Winter20.12.10 Patch	Lightning Component Tab Labels	New Topic
Winter 2020	Conga Grid User Interface Updates	New Topic
	Using the Approval Center	New Topic
	Installing and Setting up the Approval Center	New Topic
	Send Email	Updated the topic to include the new option to select Lightning Email Templates when sending an email.

Grid for Administrators

Select one of the following topics for more information:

- Conga Grid Deployment Process
- Install or Upgrade Conga Grid
- Setup Options
- Manage Licenses and Assign Permission Sets
- Assign Permission Sets
- Sharing Settings
- Security and Permission Sets for Visualforce Pages
- Export and Import Settings
- Import an Activities (AG) Tab into an Organization
- Grant Login Access to an Organization
- Track Conga Composer and Grid Usage

Conga Grid Deployment Process

The process for deploying Conga Grid℠ into a production organization ("org") is similar to deploying other Salesforce® custom solutions. These steps may differ based on your own internal processes and deployment strategies. You can modify the steps to meet your specific needs. Please contact Conga Grid Support if you have any questions.

Below is a typical deployment lifecycle:

1. Create a sandbox or use an existing sandbox environment.
   Refer to the Create or Refresh a Sandbox topic in the Salesforce documentation. Creating a sandbox based on a production org after Conga Grid has been installed in the production org will replicate the Conga Grid settings and configuration.
2. Install Conga Grid in the sandbox.
   See the Conga Grid installation instructions. Click **Install in sandbox** when prompted for an installation option.
   a. Click **Install in the sandbox** when prompted for an installation option.
   b. Assign permission sets to users separately. This will restrict most users (non-administrators) from accessing Conga Grid Setup.
3. Configure Conga Grid.
   During this process, please contact the Conga Grid team as needed to resolve any questions.
   Typical configurations Conga Grid include:

- Creating Views
- Creating multi-tabular Conga Grids for both page layouts and Visualforce® pages
- Restricting security of Conga Grid features to users or profiles
- Restricting security of Conga Grid fields or objects to users or profiles
- Creating Conditional Formatting rules, Formula Fields, Custom Actions, and other advanced configurations

4. Determine security settings for users
   The views and features that you set up will depend on your organization's needs. Conga Grid is flexible in letting you choose which features to expose to your end-users. The default setting is to expose all features, but with a few clicks, you can easily turn off (hide) features globally, per profile, or per individual user. Here are some simple guidelines:
   - Use Conga Grid Setup to set feature security
   - Use license management and permission sets to control security for the Conga Grid app
   - Use sharing settings and the Manage Views tool to control Conga Grid view
   - Use Conga Grid Setup to grant the user access to Visualforce pages
5. Test the configuration in the sandbox.
   Consider the following scenarios when testing the configuration:
   - Access to the Conga Grid app in general
   - Access to Visualforce pages
   - Logging in as users of different profiles
   - Verifying settings and functionality as both administrative and non-administrative users install Conga Grid in the Production Org
6. Install Conga Grid in the production org.
   Refer to the Conga Grid installation instructions. When installing Conga Grid into a production org, the default license type is a 15-day trial for five users. Contact your Conga Grid account representative to activate licenses and to set the seat count appropriately.
7. Move configured items from the sandbox to production org.
   See the topic Export and Import Conga Grid Settings for instructions. Conga Grid versions 1.49.83 and earlier require you to also export Conga Grid settings and to create a change set for custom actions. We recommend upgrading to the latest version of Conga Grid to avoid this process.
8. License a subset of users in the production org.
   Select a team of Conga Grid power users to internally test the initial deployment, and assign them licenses as needed. Ensure they have duplicate security settings as defined in the sandbox org.

9. Test the production org functionality to ensure it is the same as the sandbox. Consider loading two separate screens: one for the sandbox instance and another for the production instance. Gather feedback, and adjust accordingly. The team may even consider writing internal documentation that describes key workflows.
10. Expand the base of licensed users until all licenses are consumed. Continue to add users in larger pools. Here are some scenarios that have worked well in the past:
    Conga Grid deployment within a department of 50 users
    - Select one to five power users.
    - Power users train a larger set of 10 to 20 users.
    - Power users train the entire department.

    Conga Grid deployment within a company of three departments, 150 users total
    - Determine a core team of 1 to 10 cross-departmental power users.
    - Power users train a larger set of users within their departments.
    - Power users train the entire respective department.
    - Host a company-wide training session.
11. Timeline for Deployment. The timeline for deployment will vary based on your organization. If you need help determining a timeline for your specific organization, please contact your Conga Grid account representative.

Refer to the Salesforce Development Lifecycle Guide for more information. Conga Grid follows a similar deployment path.

Install or Upgrade Conga Grid

Conga Grid℠ is supported on the following Salesforce® editions:

- Professional Edition
- Enterprise Edition
- Performance Edition
- Unlimited Edition

You must have a Salesforce account to install the Conga Grid app.

To install or update Conga Grid

1. Log in to your Salesforce account.
2. Go to the Salesforce AppExchange.
3. Search for the Conga Grid app.
4. On the Conga Grid summary page, click **Get It Now**.

5. When prompted to choose a login option, click **Log in to the App Exchange**.
6. Choose an installation option:
   - Install in Production
   - Install in Sandbox
7. On the next screen, select the check box for **I have read and agree to the terms and conditions**. Then click **Confirm and Install**.
8. Log into your Salesforce account again if prompted.
9. Choose Install for Admins Only as your installation option. Click **Install** or **Upgrade**.
10. When the installation is complete, go to the App menu in Salesforce and click Conga Grid. Or, click **Conga Grid** in the App Launcher.

Setup Options

To get started, click Conga Grid Setup and select the Global Defaults option in the object list. You will see three tabs: Global Setup, Feature Security, and Object Visibility.

Global Setup

Use the Global Setup tab to do the following:

- Create new Multi-Tabular Conga Grids℠ that users can access from tabs in Salesforce®. See Multi-Tabular Conga Grid for instructions.
- Export and import Conga Grid settings between different organizations. See Export and Import Conga Grid Settings for instructions.

Feature Security

Use the Feature Security tab to globally set which actions and features will be available for end-users across the entire organization.

Select the **Global Defaults** option in the object list, and click the **Feature Security** tab.

The Enable option enables the feature. The Disable option disables the feature. The Inherit option will inherit the feature security from the user's Salesforce permissions.

- To set feature security at a global level, select the appropriate radio button in the field header at the top of the feature list.
- To set the security of individual features, select the appropriate radio button in the field header within the feature list.

Some features are disabled by default. You can enable them by clicking the **Enable** option.

- Lightning UI Beta

- Mass Update Large Batch
- Paste to Batch Update
- Batch Delete

Next, choose to apply feature security to **All Users** in an org, by **Profile**, or by individual **User**. If you select Profiles or Users, a selection menu appears that contains a list of the profiles or users to choose from.

The Default Visible check box controls the Inherit functionality for all features. If you clear the Default Visible check box, Inherit will be interpreted as Hide by default.

Click **Save** after making changes.

Feature Security for Individual Objects

You can also set feature security for any object in the object list, rather than using Global Defaults. The following figure shows an example of the User object. Under the User Features tab, you can disable all users within your organization from adding new records.
Click Save after making any changes.

Object Visibility

Use the Object Visibility tab to set which objects will be available for end-users in the Conga Grid Explorer.

Select the **Global Defaults** option in the object list, and click **Object Visibility**.

- To set object visibility at a global level, select the appropriate radio button in the field header at the top of the object list.
- To set object visibility for individual objects, select the appropriate radio button for the individual object within the object list.

To show, hide, or inherit visibility from the user's Salesforce permissions for individual objects, click **Show**, **Hide**, or **Inherit** for each object. To show, hide, or inherit visibility for all objects at once, select the appropriate radio button in the field header at the top of the Object Visibility list.

Next, choose to apply object visibility to **All Users** in an org, by **Profile**, or by individual **User**.

The Default Visible check box controls the Inherit functionality for all objects. If you clear the Default Visible check box, Inherit will be interpreted as Hide by default.

Field Visibility

This section describes how administrators can set which fields will be visible or read-only for specific objects.

In Conga Grid Setup, select an object from the object list, then click **object Fields**. Each check box can be toggled in three different ways:

- A checked box means the field will be:
- Editable when using inline edit mode
- Available to use with the Mass Update action
- Visible to the end-user
- Summed in the bottom columns of the grid (for fields that can be summed, such as currency)
- A clear box means the field is disabled.
- A grey box means that the visibility and edit permissions will be inherited from the user's Salesforce security profile. If field-level security is already enabled in Salesforce for a given field, Conga Grid will honor that security.

If you can edit a field in the Record Detail page, then you can edit it in the grid unless you explicitly disable it in the object Fields tab. You can also apply field visibility to **All Users** in an org, by **Profile**, or by individual **User**.

Manage Licenses and Assign Permission Sets

Now that Conga Grid is installed in your Org, you need to give your Users access to the app. To give your Users access, you must assign Permission Sets to the Users at the Profile level and grant them a Conga Grid license.

> ⚠ Conga Grid is not compatible with the Guest User profile in Salesforce Communities.

Prerequisite:

- API must be enabled to use Conga Grid.

To assign a license, take the following steps

1. In the Salesforce Setup menu, under Build or App Setup section, select the **Installed Packages option.** You see the option to Manage Licenses next to the Conga Grid package name.
2. Click **Add Users** and select the Users you want to the License assigned to. If you want all users to have access, click **Add All Users**.
3. To add specific users, click the checkbox to the left of their name. After they are added to Selected Users, click **Add**.
4. Now you must assign the Conga Grid User permission set to the Users. Go to the Setup menu, and under the Manage Users drop-down, click **Permission Sets**.
5. Select the Conga Grid Users permission set and click **Manage Assignments**.
6. Click **Add Assignments**, and click the checkbox to the left of the Users that need access in the Active Users list, and then click **Assign**.
7. To assign a Permission Set to an individual User, go to Salesforce Setup, Manage Users, Users, (User), and go to the Permission Set Assignments related list.
8. Click **Edit Assignments** and add the Conga Grid Users permission set. Click **Save**.

Now your Users are ready to use Conga Grid!

Assign Permission Sets

Before using Conga Grid, you must enable permission sets for yourself or end users in your organization.

Conga Grid Users - This Permission Set is required to use Conga Grid.

Conga Grid Visualforce Pages - This Permission Set is required to use Conga Grid.

To enable permission sets

1. Click **Conga Grid Explorer**.
2. Select **User** from the object list.
3. Select one or more user records for which you want to assign permission sets.
4. Click **Actions** and click **Batch Add > Permission Set**.
5. In the Batch Add Permission Set screen, select **Conga Grid Users** from the drop-down list.
6. Click **OK**. A message displays that the permission set has been assigned to the selected users.

7. Repeat these steps for the **Conga Grid Visualforce Pages** permission set.

You must create a new Grid with the Conga Grid Quick Start wizard before assigning the Conga Grid Visualforce Pages permission set.

If you use the Quick Start options or if you select the **Add Page to AG Users Permission Set** check box in the following configuration wizards, Conga Grid automatically adds the resulting Visualforce pages to both permission sets:

- Assign Permission Sets
- Multi-Tabular Related List (Automatic)
- Multi-Tabular Related List (Manual)
- Single Related List

For instructions on manually assigning Visualforce pages to the Conga Grid Visualforce Pages permission set, see Security and Permission for Visualforce Pages.

Sharing Settings

The first time you access Conga Grid Setup after installation, sharing settings will be automatically created. These include Privacy settings for views, such as the Public Read Only rule. If sharing settings are not automatically created, you can create them manually using the following steps:

1. In Salesforce®, click **Setup**.
2. In the App Setup section, click **Create** and click **Objects**.
3. Under the Label column, click the **Conga Grid View** link.
4. Scroll down to the Custom Fields & Relationships related list and click the **Privacy** link under the Field Label column.
5. Scroll down to the Picklist Value Set related list, and click **New**.
6. Type **Public Read Only**, then click **Save**.

Next, you must give the Public Read Only and Public options criteria-based values to make them work properly. Follow these steps:

1. In Salesforce, click **Setup**.
2. In the **Quick Find** box, type **sharing** and select the **Sharing Settings** result.
3. From the **Manage sharing settings** list, click **Conga Grid View**.
4. In the Conga Grid View Sharing Rules related list, click **New**.
5. Enter the following values, then click **Save**:
    - Label: **Conga Grid View**
    - Rule Name: **Conga Grid_View**
    - Rule Type: **Based on Criteria**
    - Field: **Privacy**

   - Operator: **Equals**
   - Value: **Public Read Only**
   - Share With: **Public Groups, All Internal Users**
   - Access Level: **Read Only**
6. In the Conga Grid View Sharing Rules related list, click **New**.
7. Enter the following values, then click **Save**:
   - Label: **Conga Grid View1**
   - Rule Name: **Conga Grid_View1**
   - Rule Type: **Based on Criteria**
   - Field: **Privacy**
   - Operator: **Equals**
   - Value: **Public**
   - Share With: **Public Groups, All Internal Users**
   - Access Level: **Read/Write**

Security and Permission Sets for Visualforce Pages

This topic describes how administrators can grant security to Visualforce pages after they have been created.

Here are some different options for setting the security of Visualforce® pages:

- Create a Permission Set
- Add Visualforce Pages to a User Profile
- Enable Profile Access for Visualforce Pages

Create a Permission Set

This option allows global access to all Visualforce pages that you create. Follow these steps:

1. In Salesforce®, click Setup.
2. In the Administration Setup section, click Manage Users and click Permission Sets.
3. Click New.
4. In the Label box, type Conga Grid Visualforce Pages. The API Name field will automatically populate based on the Label entry.
5. From the User License list, select the type of users who will use this permission set. Or, leave this section blank to make the Visualforce pages available to multiple profiles.

6. Click Save.
7. In the next screen, click VisualForce Page Access.
8. Click Edit.
9. In the Available Visualforce Pages section, select the Visualforce pages you created and click Add to add them to the Enabled Visualforce Pages section.
10. Click Save.
11. Add individual users to the permission set. See Assign Permission Sets for instructions.

Add Visualforce Pages to a User Profile

This option grants permission for specific user profiles to view certain pages. Follow these steps:

1. In Salesforce, click Setup.
2. In the Administration Setup section, click Manage Users and click Profiles.
3. Select a profile by clicking its Name link.
4. Move the cursor over the Enabled Visualforce Page Access link and click Edit.
5. In the Available Visualforce Pages section, select the Visualforce pages you created and click Add to add them to the Enabled Visualforce Pages section.
6. Click Save.

Enable Profile Access for Visualforce Pages

These steps are similar to the previous section.

1. In Salesforce, click Setup.
2. In the App Setup section, click Develop and click Visualforce Pages.
3. Click the Security link next to the desired Visualforce page.
4. Select profiles from the Available Profiles list, and click Add to move them to the Enabled Profiles list.
5. Click Save.

Export and Import Settings

> Due to the ever-changing complex nature of programming languages, Conga does not support Visualforce or any elements related to APEX.

System administrators only:

At the end of this topic are instructions for exporting and importing settings without the workbench.

Follow these steps to export and import the following:

- Conga Grid℠ settings
- Conga Grid views
- Conga Grid static resources and custom actions
- Visualforce pages

To export and import:

1. Click **Conga Grid Setup** in the organization your exporting from.
2. Click **Global Defaults** at the top of the object list on the left.
3. Click **Export**.
4. Select the Pages you would like to export.
   These are the pages that are imported into your secondary org. Only select the pages you would like to migrate.

   > ⚠ The Export wizard cannot tell Conga Grid VisualForce pages apart from other VisualForce pages. Keep this in mind if you select all the options using the Select All Checkbox in the top left.

5. Select the custom Visualforce Tabs you would like to export. If you have not created any tabs, or do not want to migrate any tabs, select **Continue**.
6. Select the Settings you would like to export. If you already have settings in the org you are migrating to that you would like to keep, you can choose to leave these options unchecked and select **Continue**.
7. Select the Views you would like to export. Ensure that any views which are hard coded into your Conga Grid Visualforce Pages are selected.
8. Select any Custom Actions or Panes to export. If you do not have any Custom Actions or Panes, you can select **Continue**.
9. You receive a Conga Grid Zip File containing your selections that can be imported into another org.
10. Click **Conga Grid Setup** in the organization you would like to migrate to.
11. Click **Global Defaults** at the top of the object list on the left.
12. Click **Import**.
13. Select the zipped Conga Grid file in the Downloads folder and click **Open**.
14. Select the Settings you would like to import. If you did not choose any settings, for example, during the import, select **Continue**.
15. Click **Continue** and Select the Custom Actions you would like to import.
16. Click **Continue** and Select the Views you would like to import.

17. Click **Continue** and Select the Pages you would like to Import.
18. Conga Grid detects if views and pages already exist in the organization you are migrating to. The Duplicates screens show a list of duplicate views and pages. Select the **Use Existing** option next to each page or view to use the preexisting option, or clear the option to overwrite that page or view with the new one. Click **OK**.
19. Select the Tabs you would like to import.
20. Click **Continue**.

> ⚠ The Import tool does not import sharing settings for objects or permission sets. It does not use the change set system from Salesforce and does not communicate with Salesforce or change your data. Also, you cannot set views for new pages during the import.

Export and Import Settings Without the Workbench

To export and import setting without using the Workbench:

1. Open a console in a web browser by pressing the F12 key.
2. To export settings, run a query in the console using the following Javascript code:
```
sforce.connection.query("SELECT crmc_pp__JSON__c,Name,crmc_pp__Owner_Label__c,crmc_pp
__Owner_Type__c,crmc_pp__Owner__c,crmc_pp__Setting_Name
__c,crmc_pp__Setting_Type__c,crmc_pp__Table_Name__c
FROM crmc_pp__Grid_Perm_Setting__c");
```
3.

   This yields an array of JavaScript objects. Copy them into a text file and name it **Company Name - AG Settings**. (Replace **Company Name** with the actual company name.)
4. To import the settings, click **Conga Grid Setup**.
5. Open a console in a web browser by pressing the **F12** key.
6. Run the following JavaScript code in the console. Replace Company Name - AG Settings with the name of the text file you just created. Do not run it more than once or the settings are interpreted twice.
```
var settingsList = "Company Name-AG Settings"
var sObjList = settingsList.map(function (setting) {
 var newSetting = new sforce.SObject('crmc_pp__Grid_Perm_Setting__c');
 Object.each(setting, function (field, name) {
 if (field != null) newSetting[name] = field
 });
 return newSetting;
});
sforce.connection.create(sObjList);
```

Import an Activities (AG) Tab into an Organization

1. Download the zip file Conga Grid_TaskTab.zip.
2. Click **Conga Grid Setup**.
3. Click **Global Defaults** in the object list.
4. Click the **Global Setup** tab.
5. Click **Import**.
6. In the File Selection dialog, locate the zip file that you just downloaded. Click **Open**.
7. Dismiss the Import Complete dialog.
8. Create a tab for the imported Visualforce page. See Single Conga Grid Tab for instructions.

Grant Login Access to an Organization

1. Click your user name in the Salesforce® header and click **My Settings**.
2. Click Personal and click **Grant Account Login Access**.
3. Select Public from the drop-down list next to **AppExtremes**, **dba Conga Support**. You can change this back to **No Access** after your technical support issues have been resolved.

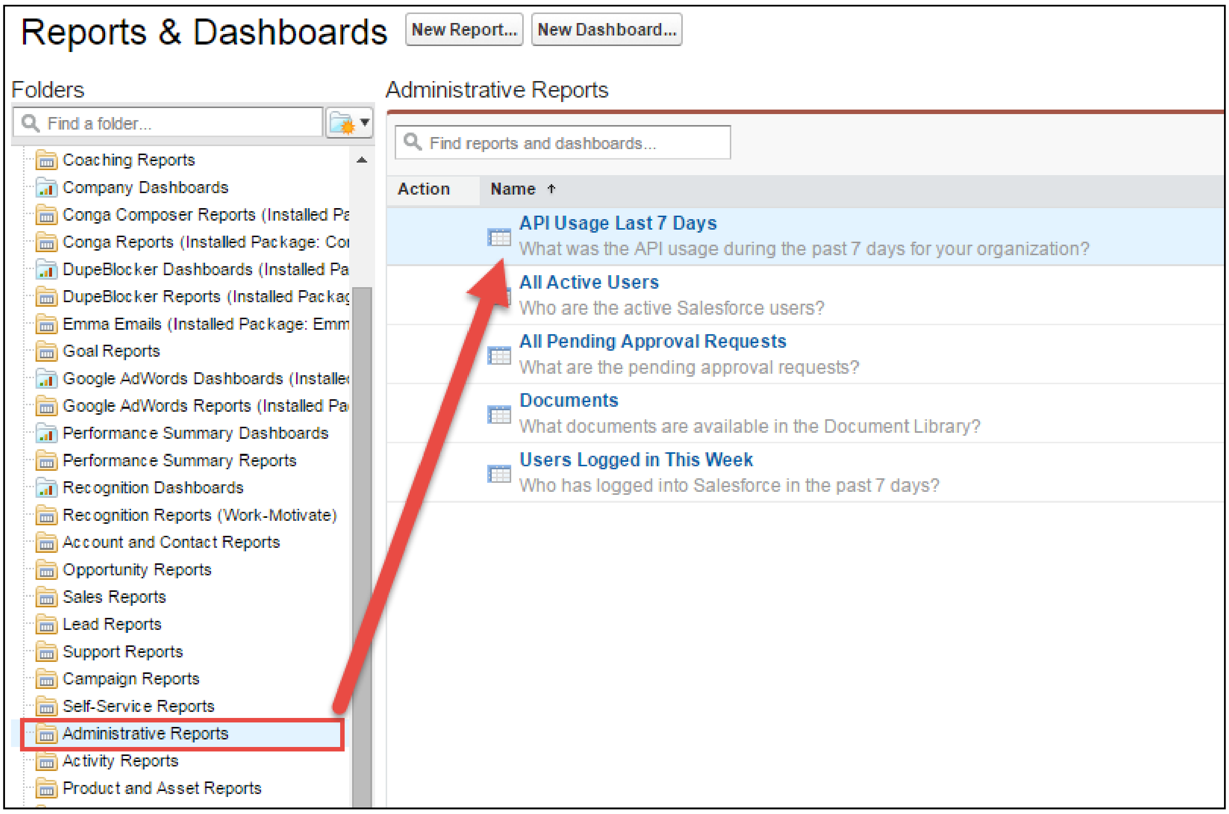
Finding your Organization ID

Conga Grid℠ Technical Support may ask for your organization ID when granting login access to your organization. Follow these steps to locate your ID:

1. In Salesforce, click **Setup**.
2. In the Administration Setup section, click **Company Profile** and click **Company Information**.
3. On the Company Information screen, locate the Salesforce.com Organization ID.

Track Conga Composer and Grid Usage

To track API usage for Conga Products, create a standard report in Salesforce® called API Usage Last 7 Days.



Reports & Dashboards

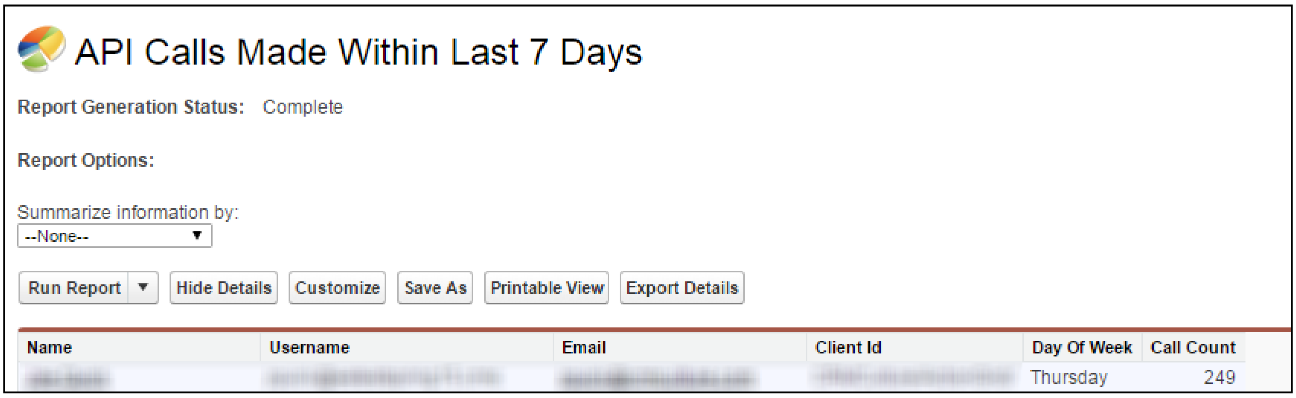
Folders

- Coaching Reports
- Company Dashboards
- Conga Composer Reports (Installed Pa
- Conga Reports (Installed Package: Co
- DupeBlocker Dashboards (Installed Pa
- DupeBlocker Reports (Installed Packag
- Emma Emails (Installed Package: Emm
- Goal Reports
- Google AdWords Dashboards (Installe
- Google AdWords Reports (Installed Pa
- Performance Summary Dashboards
- Performance Summary Reports
- Recognition Dashboards
- Recognition Reports (Work-Motivate)
- Account and Contact Reports
- Opportunity Reports
- Sales Reports
- Lead Reports
- Support Reports
- Campaign Reports
- Self-Service Reports
- Administrative Reports
- Activity Reports
- Product and Asset Reports

Administrative Reports

Action	Name ↑
	API Usage Last 7 Days What was the API usage during the past 7 days for your organization?
	All Active Users Who are the active Salesforce users?
	All Pending Approval Requests What are the pending approval requests?
	Documents What documents are available in the Document Library?
	Users Logged in This Week Who has logged into Salesforce in the past 7 days?

You can modify this report to see the API usage per user per day. The higher the API call, the more times the users accessed Conga Grid.



API Calls Made Within Last 7 Days

Report Generation Status: Complete

Report Options:

Summarize information by: --None--

Name	Username	Email	Client Id	Day Of Week	Call Count
[redacted]	[redacted]	[redacted]	[redacted]	Thursday	249

Grid for Users

Select one of the following topics for more information:

- Getting Started with Conga Grid
- Guided Tour of Conga Grid
- Building Conga Grid Solutions
- Advanced Conga Grid Solutions
- Code Examples for Custom Actions in Conga Grid
- Secure Mass Update
- Troubleshooting Conga Grid
- Conga Grid Product Links

Getting Started with Conga Grid

Learn how to install, set up, and use basic features in Conga Grid℠.

- Conga Grid Quick Start
- Conga Grid Basics
- Build Conga Grid from existing Salesforce data

Conga Grid Quick Start

Set up Conga Grid℠ using pre-configured templates or by importing existing Salesforce® data from your organization.

Conga Grid Quick Start Options

Use the Conga Grid℠ Quick Start tab to automatically create a Conga Grid using Quick Grid Templates or by importing existing Salesforce® data from your organization. The options in this tab are meant for new users who want to get started using Conga Grid for the first time.

Three ways to get started:

Quick Start: Pre-Built Conga Grids

This option creates a Conga Grid layout based on one of four common roles:

- Sales - One place for a Sales Rep covering Leads, Opportunities, and Accounts.

- Sales Executive - Opportunity and pipeline management, geared toward sales leaders and directors.
- Case Management - For Support Reps who make calls, emails, and drive cases to closure.
- Salesforce Admin - Tools for admins who need to manage users, Visualforce pages, permission sets, and Conga Grid views.
- CPQ - Meant for admins using CPQ.
- Conga Contracts for Salesforce - Contracts Hub where users can view Contracts, Opportunities, and Quotes with Managed Clauses, Content Documents, and Redlines in the Reading Pane. Requires Conga Grid to be installed in the org. Quick Start: Create a Custom Conga Grid

Use these grids if you have apps, list views, or custom setups.

- Create from Salesforce App - Allows you to select an existing app with list views and convert those views into a multi-tab Conga Grid.
- Create from List Views - Allows you to select a single list view and turn it into a Conga Grid.

The Quick Start options were designed for new Conga Grid users who are not familiar with configuring Conga Grid. You can still build a new grid or customize an existing grid from the Conga Grid Setup tab. See Conga Grid Configuration for more information.

You may need to add the Conga Grid Quick Start tab to the row of tabs if it is not already visible. Follow these steps:

1. Click **All Tabs (+)** in the row of tabs.
2. In the upper-right corner of the All Tabs page, click **Customize My Tabs**.
3. Click Conga Grid - Quick Start in the Available Tabs list.
4. Click **Add** to move it to the Selected Tabs list.
5. Click **Save**.

Quick Grid Templates in Conga Grid

You can install pre-built Conga Grid templates optimized for a role in Salesforce.

To select a template:

1. Click the **Conga Grid℠ Quick Start** tab.
2. Click **Install Pre-Built Conga Grid**.
3. Select the type of team for which you are setting up Conga Grid:
    - **Sales**: This option creates a Multi-Tabular Conga Grid based on the Leads object with tabs for My Leads, My Opportunities, My Accounts, and My Tasks. A

Reading Pane is displayed with additional tabs to provide more details about selected Leads records.

- **Sales Executive**: This option creates a Conga Grid based on the Case object. A Reading Pane is displayed with a Multi-Tabular Conga Grid with tabs for Account Details, Tasks, Chatter, and Detail Grid. These tabs provide more details about the selected Case records.
- **Case Management**: This option creates a Conga Grid based on the Account object. A Reading Pane is displayed with a Multi-Tabular Conga Grid with tabs for Opportunities, Contacts, Cases, Tasks, Chatter, and Detail Grid. These tabs provide more details about the selected Account records.
- **Salesforce® Admin**: This option creates a Conga Grid layout similar to that of Sales.

4. Click **install**.
5. If duplicate fields are found, a Duplicates screen appears. Select or uncheck any of the duplicate records listed, then click **OK**.
6. When processing is complete, an Import Complete screen appears. Click **Go To Tab** to view the new grid.

The tabs that comprise each Conga Grid are saved as individual views. You can rename the views, share them, or set their privacy by following the steps in Manage Views.

Conga Grid Basics

Learn how to use basic features in Conga Grid℠. We recommend that you start with the Conga Grid Product Guide.

Conga Grid Product Guide

Learn Conga Grid

- About Grid
- Conga Grid Getting Started Video

Install Conga Grid

- Conga Grid Deployment Process
- Install Conga Grid
- Manage Licenses

- Assign Permission Sets

Configure Conga Grid

- Conga Grid Quick Start Options
- Options for Starting Conga Grid
- Manage Views
- Configure Conga Grid for Lightning

Use Conga Grid

- Using Conga Grid for the First Time
- Choose Fields to View
- Add Records Inline
- Edit the Conga Grid
- Filter and Group Records
- Copy and Paste Records
- Sort Records in Conga Grid
- Conditional Formatting
- Reading Pane
- Add Formula Fields
- Formula Functions
- Date Formula Fields
- Actions

Troubleshoot Conga Grid

- Conga Grid Frequently Asked Questions
- Conga Grid Known Issues and Workarounds

Conga Grid Getting Started Video

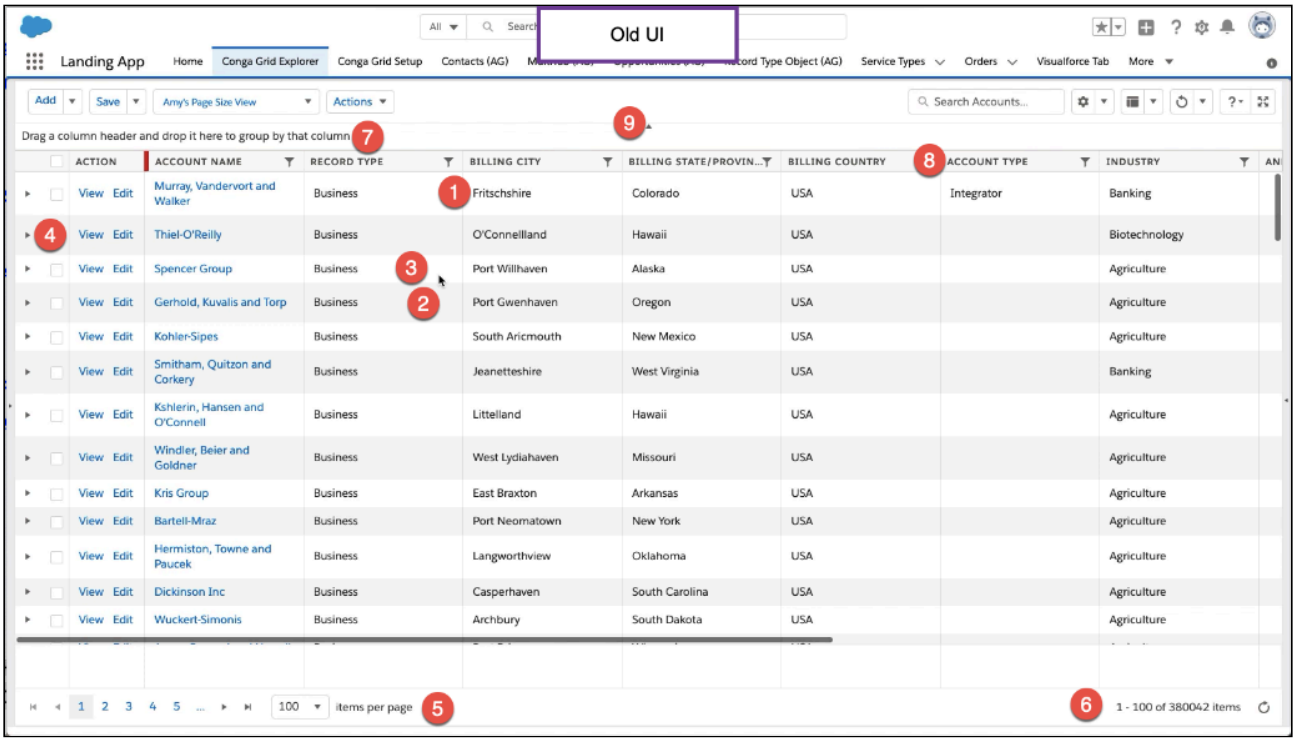
This video shows you how to get started with Conga Grid.

How to Get Started with Conga Grid

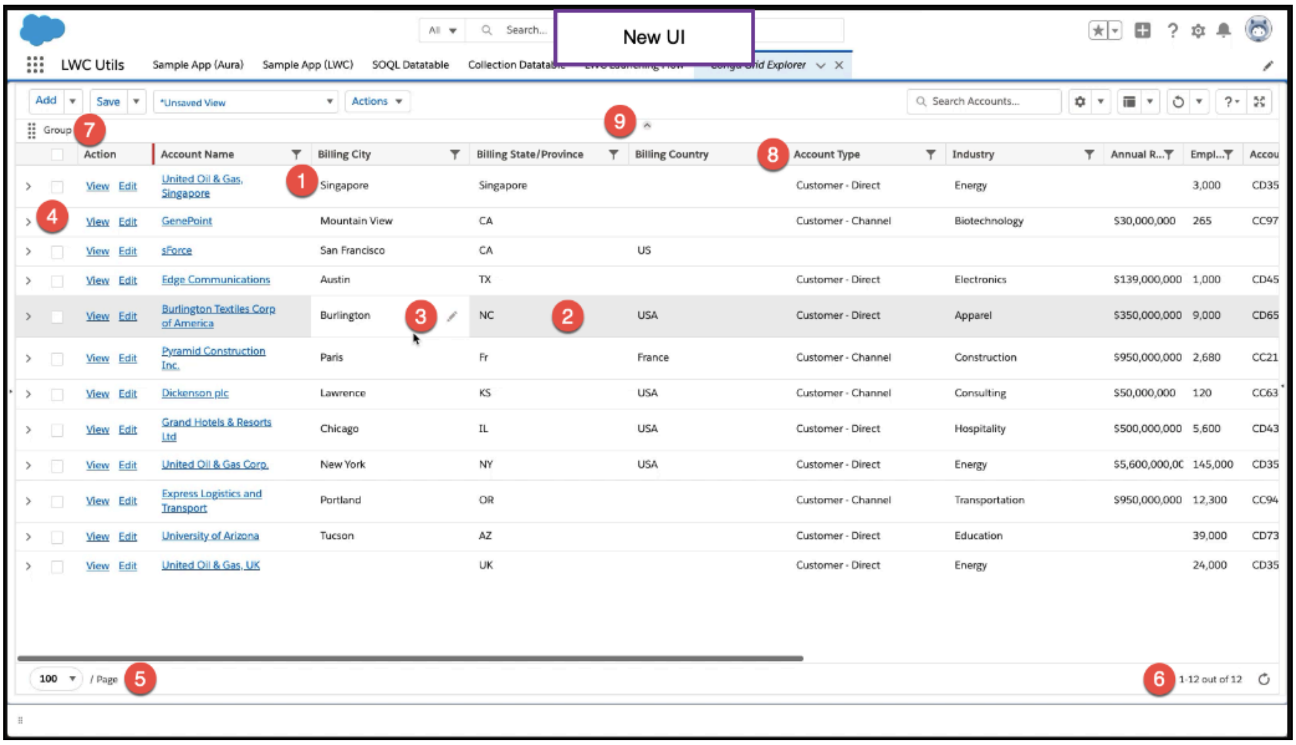
Conga Grid User Interface Updates

Conga Grid was updated with several user interface updates to give you a more streamlined experience. The updates include the following:

General Updates



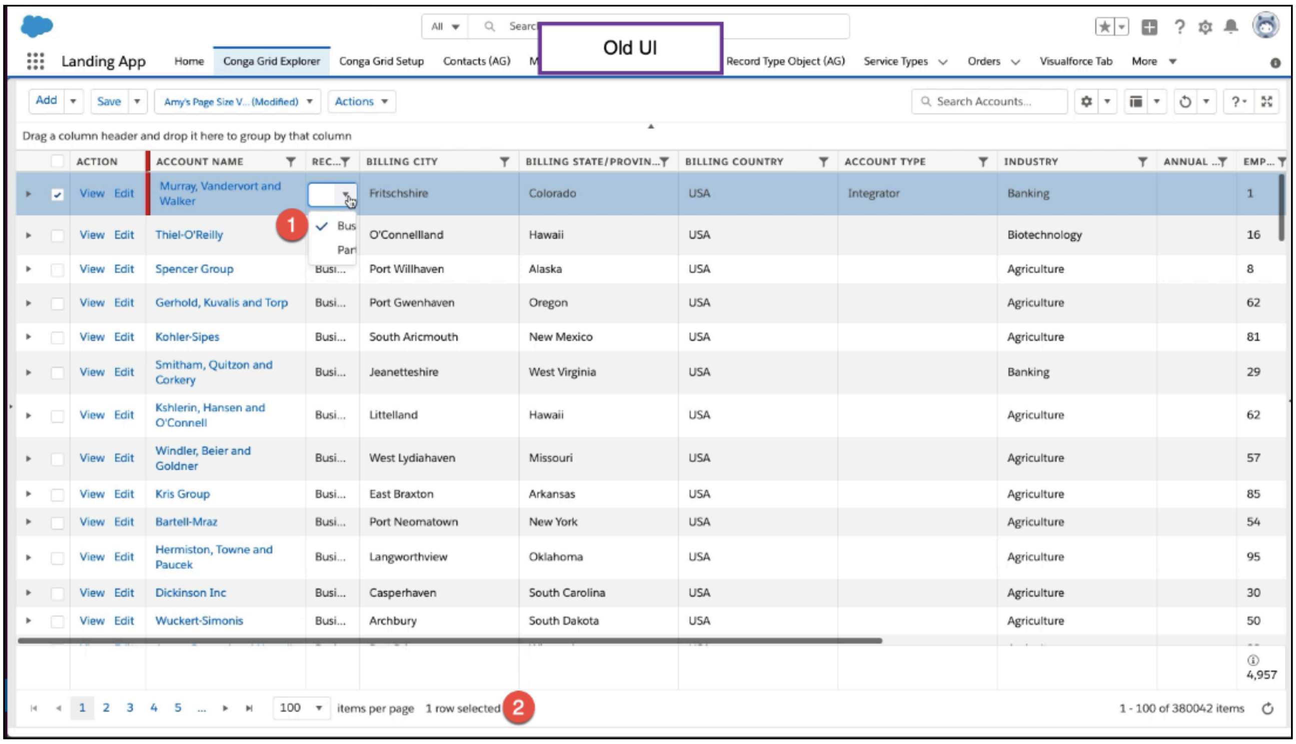
	ACTION	ACCOUNT NAME	RECORD TYPE	BILLING CITY	BILLING STATE/PROVIN...	BILLING COUNTRY	ACCOUNT TYPE	INDUSTRY	AN
□	View Edit	Murray, Vandervort and Walker	Business	Fritschshire	Colorado	USA	Integrator	Banking	
	View Edit	Thiel-O'Reilly	Business	O'Connellland	Hawaii	USA		Biotechnology	
□	View Edit	Spencer Group	Business	Port Willhaven	Alaska	USA		Agriculture	
□	View Edit	Gerhold, Kuvalis and Torp	Business	Port Gwenhaven	Oregon	USA		Agriculture	
□	View Edit	Kohler-Sipes	Business	South Aricmouth	New Mexico	USA		Agriculture	
□	View Edit	Smitham, Quitzon and Corkery	Business	Jeanetteshire	West Virginia	USA		Banking	
□	View Edit	Kshlerin, Hansen and O'Connell	Business	Littelland	Hawaii	USA		Agriculture	
□	View Edit	Windler, Beier and Goldner	Business	West Lydiahaven	Missouri	USA		Agriculture	
□	View Edit	Kris Group	Business	East Braxton	Arkansas	USA		Agriculture	
□	View Edit	Bartell-Mraz	Business	Port Neomatown	New York	USA		Agriculture	
□	View Edit	Hermiston, Towne and Paucek	Business	Langworthview	Oklahoma	USA		Agriculture	
□	View Edit	Dickinson Inc	Business	Casperhaven	South Carolina	USA		Agriculture	
□	View Edit	Wuckert-Simonis	Business	Archbury	South Dakota	USA		Agriculture	



	Action	Account Name	Billing City	Billing State/Province	Billing Country	Account Type	Industry	Annual R...	Empl...	Accou
□	View Edit	United Oil & Gas, Singapore	Singapore	Singapore		Customer - Direct	Energy		3,000	CD35
	View Edit	GenePoint	Mountain View	CA		Customer - Channel	Biotechnology	$30,000,000	265	CC97
□	View Edit	sForce	San Francisco	CA	US					
□	View Edit	Edge Communications	Austin	TX		Customer - Direct	Electronics	$139,000,000	1,000	CD45
□	View Edit	Burlington Textiles Corp of America	Burlington	NC	USA	Customer - Direct	Apparel	$350,000,000	9,000	CD65
□	View Edit	Pyramid Construction Inc.	Paris	Fr	France	Customer - Channel	Construction	$950,000,000	2,680	CC21
□	View Edit	Dickenson plc	Lawrence	KS	USA	Customer - Channel	Consulting	$50,000,000	120	CC63
□	View Edit	Grand Hotels & Resorts Ltd	Chicago	IL	USA	Customer - Direct	Hospitality	$500,000,000	5,600	CD43
□	View Edit	United Oil & Gas Corp.	New York	NY	USA	Customer - Direct	Energy	$5,600,000,0C	145,000	CD35
□	View Edit	Express Logistics and Transport	Portland	OR		Customer - Channel	Transportation	$950,000,000	12,300	CC94
□	View Edit	University of Arizona	Tucson	AZ		Customer - Direct	Education		39,000	CD73
□	View Edit	United Oil & Gas, UK		UK		Customer - Direct	Energy		24,000	CD35

1. The font size was changed from 14 to 13.
2. The rows no longer have alternating colors. The row is now only highlighted by your cursor.

3. There is now a pencil or a lock indicating if a cell is available to edit.
4. The carrot symbol to open the detail pane was changed to a chevron.
5. The page size and page numbers display format changed.
6. The text was shortened to only indicate how many items are displayed on the current page.
7. The text was shortened indicating where to group fields.
8. The column headers are now camel case instead of all uppercase.
9. The collapse toolbar icon changed from a carrot symbol to a chevron.

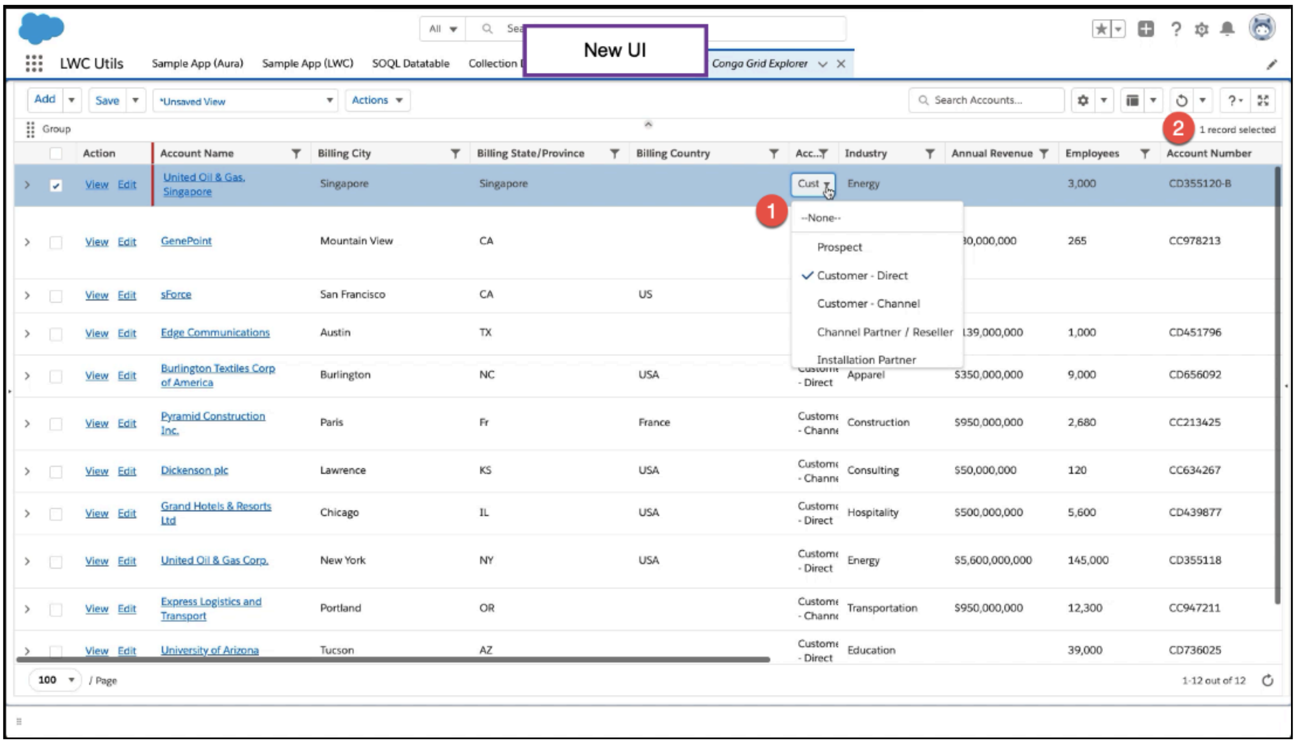


Drag a column header and drop it here to group by that column

	ACTION	ACCOUNT NAME	REC...	BILLING CITY	BILLING STATE/PROVIN...	BILLING COUNTRY	ACCOUNT TYPE	INDUSTRY	ANNUAL ...	EMP...
☑	View Edit	Murray, Vandervort and Walker		Fritschshire	Colorado	USA	Integrator	Banking		1
□	View Edit	Thiel-O'Reilly		O'Connellland	Hawaii	USA		Biotechnology		16
□	View Edit	Spencer Group	Busi...	Port Willhaven	Alaska	USA		Agriculture		8
□	View Edit	Gerhold, Kuvalis and Torp	Busi...	Port Gwenhaven	Oregon	USA		Agriculture		62
□	View Edit	Kohler-Sipes	Busi...	South Aricmouth	New Mexico	USA		Agriculture		81
□	View Edit	Smitham, Quitzon and Corkery	Busi...	Jeanetteshire	West Virginia	USA		Banking		29
□	View Edit	Kshlerin, Hansen and O'Connell	Busi...	Littelland	Hawaii	USA		Agriculture		62
□	View Edit	Windler, Beier and Goldner	Busi...	West Lydiahaven	Missouri	USA		Agriculture		57
□	View Edit	Kris Group	Busi...	East Braxton	Arkansas	USA		Agriculture		85
□	View Edit	Bartell-Mraz	Busi...	Port Neomatown	New York	USA		Agriculture		54
□	View Edit	Hermiston, Towne and Paucek	Busi...	Langworthview	Oklahoma	USA		Agriculture		95
□	View Edit	Dickinson Inc	Busi...	Casperhaven	South Carolina	USA		Agriculture		30
□	View Edit	Wuckert-Simonis	Busi...	Archbury	South Dakota	USA		Agriculture		50
										4,957

1 2 3 4 5 ... 100 items per page 1 row selected

1 - 100 of 380042 items



1 record selected

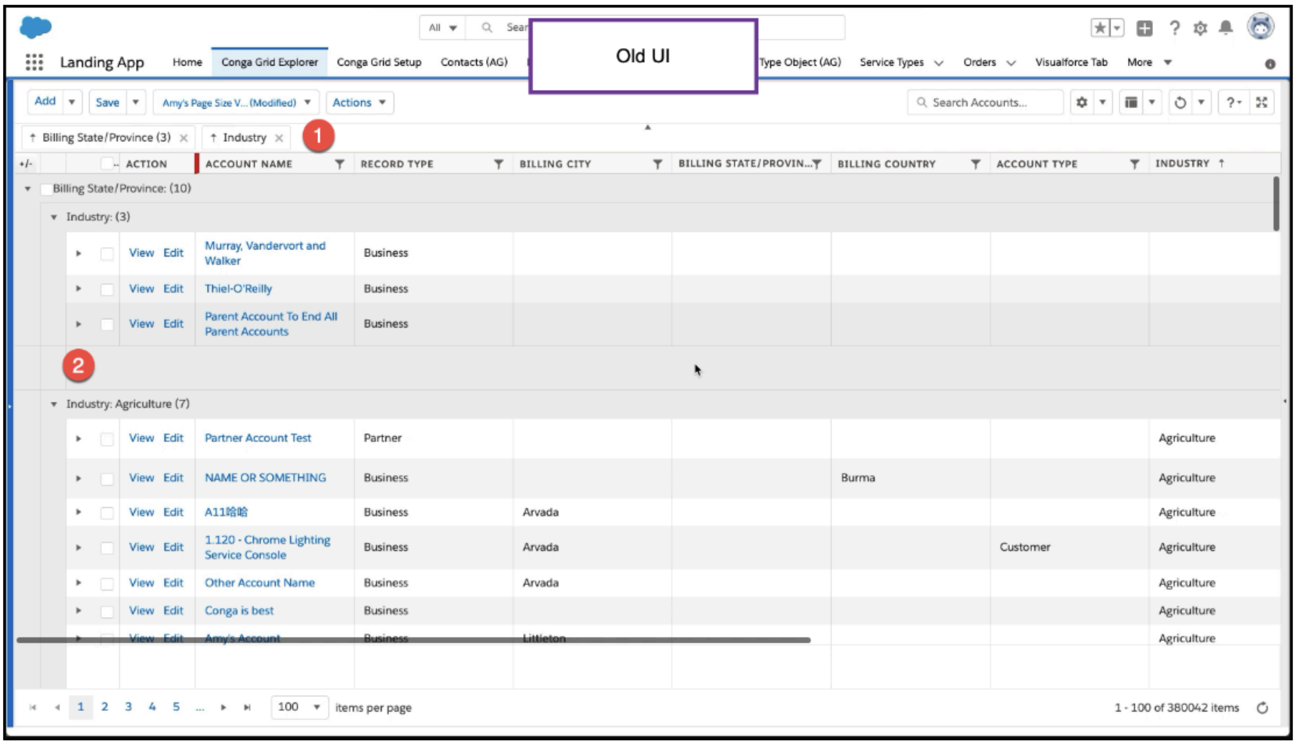
	Action	Account Name	Billing City	Billing State/Province	Billing Country	Acc...	Industry	Annual Revenue	Employees	Account Number
☑	View Edit	United Oil & Gas, Singapore	Singapore	Singapore		Cust	Energy		3,000	CD355120-B
□	View Edit	GenePoint	Mountain View	CA				30,000,000	265	CC978213
□	View Edit	sForce	San Francisco	CA	US					
□	View Edit	Edge Communications	Austin	TX				139,000,000	1,000	CD451796
□	View Edit	Burlington Textiles Corp of America	Burlington	NC	USA	Custome - Direct	Apparel	$350,000,000	9,000	CD656092
□	View Edit	Pyramid Construction Inc.	Paris	Fr	France	Custome - Channe	Construction	$950,000,000	2,680	CC213425
□	View Edit	Dickenson plc	Lawrence	KS	USA	Custome - Channe	Consulting	$50,000,000	120	CC634267
□	View Edit	Grand Hotels & Resorts Ltd	Chicago	IL	USA	Custome - Direct	Hospitality	$500,000,000	5,600	CD439877
□	View Edit	United Oil & Gas Corp.	New York	NY	USA	Custome - Direct	Energy	$5,600,000,000	145,000	CD355118
□	View Edit	Express Logistics and Transport	Portland	OR		Custome - Channe	Transportation	$950,000,000	12,300	CC947211
□	View Edit	University of Arizona	Tucson	AZ		Custome - Direct	Education		39,000	CD736025

100 / Page

1-12 out of 12

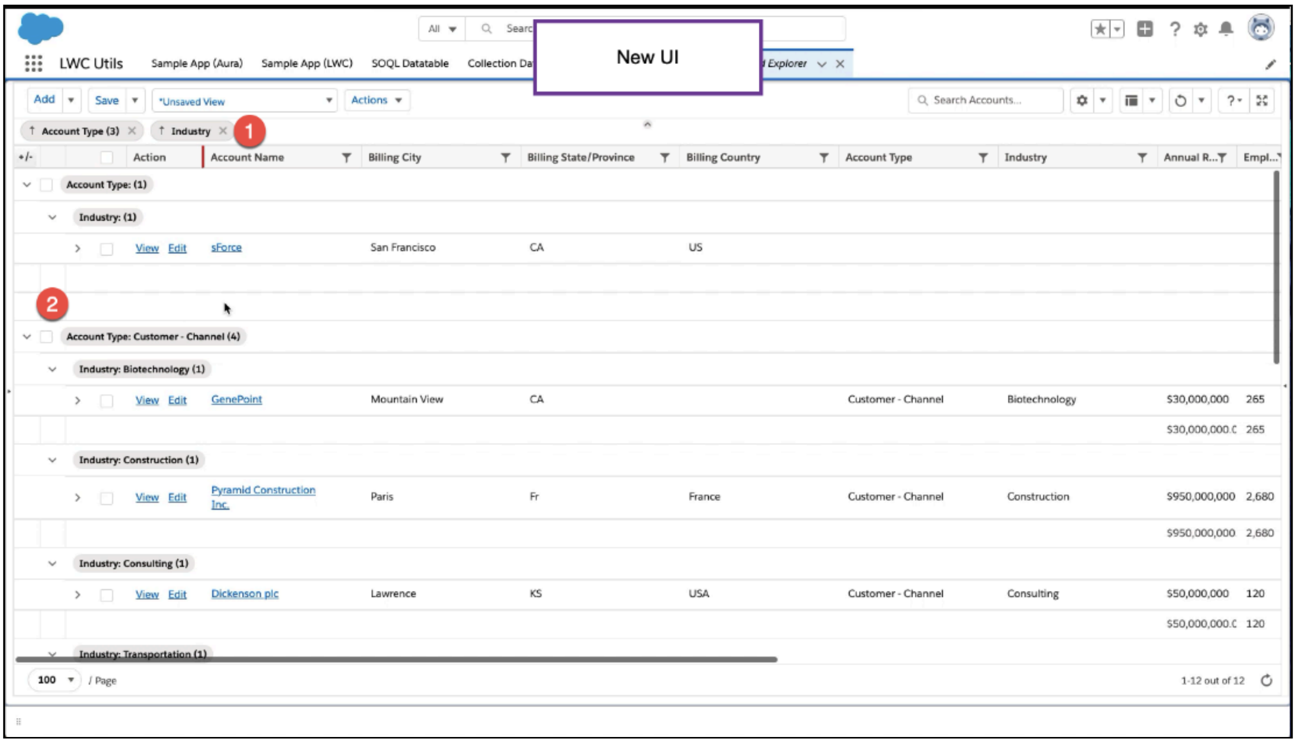
1. The menu size no longer conforms to column size and now resizes to display full contents.
2. The indicator that shows the number of records you have selected was relocated from the bottom of the page to the top within the grouping area.

Grouping Updates



+/-	ACTION	ACCOUNT NAME	RECORD TYPE	BILLING CITY	BILLING STATE/PROVIN...	BILLING COUNTRY	ACCOUNT TYPE	INDUSTRY ↑
Billing State/Province: (10)								
Industry: (3)								
	View Edit	Murray, Vandervort and Walker	Business					
	View Edit	Thiel-O'Reilly	Business					
	View Edit	Parent Account To End All Parent Accounts	Business					
Industry: Agriculture (7)								
	View Edit	Partner Account Test	Partner					Agriculture
	View Edit	NAME OR SOMETHING	Business			Burma		Agriculture
	View Edit	A11哈哈	Business	Arvada				Agriculture
	View Edit	1.120 - Chrome Lighting Service Console	Business	Arvada			Customer	Agriculture
	View Edit	Other Account Name	Business	Arvada				Agriculture
	View Edit	Conga is best	Business					Agriculture
	View Edit	Amy's Account	Business	Littleton				Agriculture

1 2 3 4 5 ... 100 items per page 1 - 100 of 380042 items

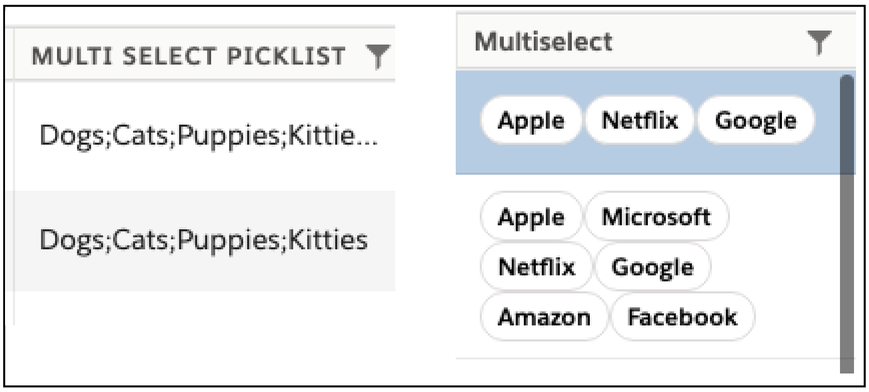


+/-	Action	Account Name	Billing City	Billing State/Province	Billing Country	Account Type	Industry	Annual R...	Empl...
Account Type: (1)									
Industry: (1)									
	View Edit	sForce	San Francisco	CA	US				
Account Type: Customer - Channel (4)									
Industry: Biotechnology (1)									
	View Edit	GenePoint	Mountain View	CA		Customer - Channel	Biotechnology	$30,000,000	265
								$30,000,000.C	265
Industry: Construction (1)									
	View Edit	Pyramid Construction Inc.	Paris	Fr	France	Customer - Channel	Construction	$950,000,000	2,680
								$950,000,000	2,680
Industry: Consulting (1)									
	View Edit	Dickenson plc	Lawrence	KS	USA	Customer - Channel	Consulting	$50,000,000	120
								$50,000,000.C	120
Industry: Transportation (1)									

100 / Page 1-12 out of 12

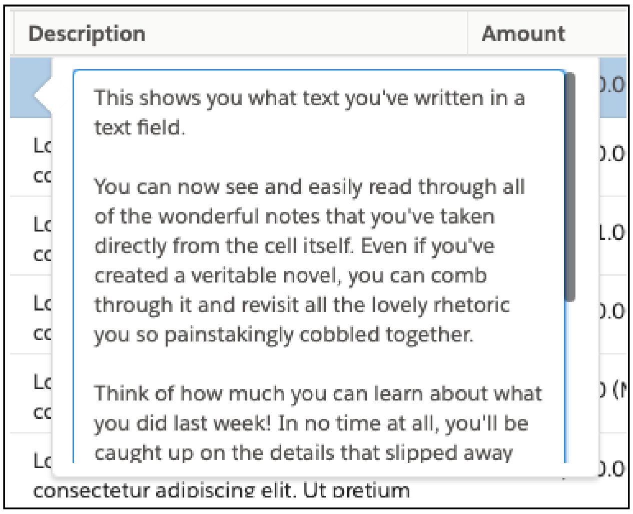
1. The Grouping tiles display format was updated.
2. The Grouping margins were changed from grey to white.

Multiselect Updates



The multiselect item display format was updated.

Text Area Popover



This shows you what text you've written in a text field.

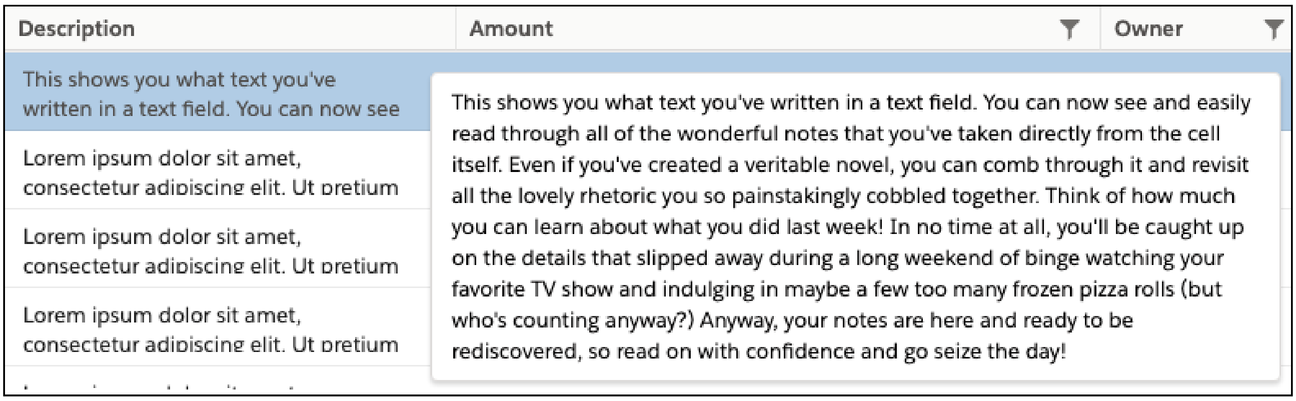
You can now see and easily read through all of the wonderful notes that you've taken directly from the cell itself. Even if you've created a veritable novel, you can comb through it and revisit all the lovely rhetoric you so painstakingly cobbled together.

Think of how much you can learn about what you did last week! In no time at all, you'll be caught up on the details that slipped away

consectetur adipiscing elit. Ut pretium

Clicking into a text area field now opens a popover to allow users to see and edit the entire field more easily.

Field Preview



Description	Amount	Owner
This shows you what text you've written in a text field. You can now see		
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pretium		
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pretium		
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pretium		

This shows you what text you've written in a text field. You can now see and easily read through all of the wonderful notes that you've taken directly from the cell itself. Even if you've created a veritable novel, you can comb through it and revisit all the lovely rhetoric you so painstakingly cobbled together. Think of how much you can learn about what you did last week! In no time at all, you'll be caught up on the details that slipped away during a long weekend of binge watching your favorite TV show and indulging in maybe a few too many frozen pizza rolls (but who's counting anyway?) Anyway, your notes are here and ready to be rediscovered, so read on with confidence and go seize the day!

A preview tooltip is now available for text area fields in Grid Views that have editing disabled to allow users to read the full contents of the cell.

Using Conga Grid for the First Time

To start Conga Grid℠, go to the App Menu in Salesforce® and click Conga Grid. The quickest way to view your Salesforce records is to click the Conga Grid Explorer tab. Depending on how Conga Grid was configured, you can also invoke it using one of these options:

Conga Grid button in the List Views or Search Results screen

Embedded Conga Grid of related lists on a page layout

Conga Grid tab for specific objects

In the Conga Grid Explorer, select an object from the left column. The corresponding records display in a new grid.

When you move the cursor over a record that has a link (for example, an Account record), a popup screen shows the details of the associated record. You can turn this feature on or off in the Conga Grid Setup.

To take a guided tour, click the drop-down arrow next to Configure and click Help. The tour shows you how to use common features in Conga Grid.

Choose Fields to View

One of the first steps in working with Conga Grid℠ is to choose which fields you want to display. See the following sections:

- Organize Fields

- Search Fields
- View Field API Names and Help Text

Organize Fields

In the Conga Grid Explorer, click an object in the object list and click Configure.
The Configure menu on the right side of the application lists the available fields for the current object. Select the specific fields you want to view.

As you add and remove fields in the Configure menu, the grid automatically refreshes to show your changes. This behavior is controlled by the Auto Update check box at the bottom of the Configure menu. Clear the check box if you do not want the grid to refresh while you are selecting fields.

Here are some tips for organizing fields (also called columns):

- To remove a field, clear the check box for the field name. Or click and drag the field off the grid.
- To rearrange fields, click and drag them to their desired order.
- To rename a field, right-click on the field name and select Column Settings. In the Column Settings screen, enter a column Label. For example, you can create shorter field names to maximize the number of columns displayed in the grid.
- To enter a Default Value for a field, enter the value in the Column Settings screen. This is useful when you need to create many new records that have the same value in a field. When you click New object to create new records, that field will populate with the default value that you specified.
- To make a field Read Only or Required, click the appropriate check box in the Column Settings screen.
- To keep a selected column frozen on the left side of the grid while scrolling horizontally, right-click on the field name and click Freeze Column.
- Once you have arranged fields the way you like them, you can save the current view. See Save Views for instructions.

Search Fields

Use the Search Fields box in the Configure menu to search for specific fields.

For example, suppose you are viewing a grid of Opportunity Products. Typing price in the Search Fields box shows all Opportunity Product field names that contain that string. If a field containing price is a related column (child), only the parent field is displayed.

Click the arrow icon above the check boxes twice to group all of the currently selected fields at the top of the available fields list. Click the arrow icon once to group the currently selected fields at the bottom of the list. Click the icon three times to revert the fields to their original order.

View API Field Names and Help Text

When you move the cursor over a field name in the Configure menu, a blue popup screen appears with the API Name and Help Text for that field (as set in Salesforce).

Add Records Inline

This feature requires some additional configuration steps to add the Add Inline option to the New object button. See Create an Add Inline Option for instructions.

1. Click the drop-down arrow next to Object and click Add Inline to add an empty record.
2. Fill in the new record information.
3. Repeat these steps to add multiple records.
4. Click Save to save your edits.

If multiple records have similar information, use the Clone action instead of Add Inline to create new records.

Edit the Conga Grid

Here are some tips for editing records in Grid:

- For text fields, click inside of a grid cell to edit its value.
- For date fields, click in the grid cell and enter a new date. Or, click the calendar icon and select a date from the calendar that appears.
- Parent fields have options that let you select existing values. An example is when you are viewing a grid of Contacts and you click in an Account field. You can select from a list of existing accounts or click the magnifier icon to search for accounts. You receive a notification that you are editing a parent object. An example is when you are viewing a grid of Opportunities and you edit a Contact record. A blue triangle icon appears next to the field, reminding you that it has updated the parent object data too. Any other Opportunity associated with that Contact also updates.
- Clicking the Edit link in the Action field displays a new Salesforce page for that record, where you can also edit fields as needed.
- You cannot edit access levels for Opportunity Team Member.

- When you edit a field, a small red triangle appears in the field to indicate that it has not yet been saved.
- To revert your edits, click the drop-down arrow next to Save and click Revert Data.
- To save your edits, click Save. If you are the owner of the current view, clicking Save also saves the current view. To save edits without saving the current view, click the drop-down arrow next to Save and click Save Data.

Keyboard Shortcuts

Press CTRL+ / to display a popup list with keyboard shortcuts that you can use. Or, click Configure in the Grid toolbar and select Keyboard Shortcuts.

Here is a list of the shortcuts that are available:

- Press TAB to advance to the next field.
- Press SHIFT+TAB to move to the previous field.
- Press ENTER to advance to the next row.
- Press SHIFT+ENTER to move up one row.
- Press CTRL+[or CTRL+] to traverse between parent and child grids.
- Press CTRL+SHIFT+S to display the Save View screen.
- Press CTRL+\ to expand and collapse the Configure menu.
- Press CTRL+I to insert a new empty row.
- Press CTRL+S to save your edits.
- Press CTRL+D to fill up or fill down.
- Press CTRL+E to enable or disable Edit mode.
- Press CTRL+V to paste data from an external table.Configure Edit Options

Edit mode is enabled by default. You can disable Edit mode by pressing CTRL+E or by using the Configure menu. Follow these steps:

1. Click the Configure ▦ button. The Configure menu appears.
2. Expand the Toolbar drop-down arrow.
3. Click Disable Edit. The grid is now in read-only mode. The Edit button appears in the toolbar. You must click this to resume editing.
4. Optional: Clear the Edit check box in the Configure menu. The editing buttons from the toolbar are removed.

Filter and Group Records

Filtering Options

You can filter values in a specific field or even multiple fields at the same time. Filtering applies to the entire record set for a given object, not just records that are displayed on the current page.

Click the Filter icon to apply a filter, then click Filter in the filtering screen. The Filter icon changes to orange:

Conga Grid displays a Basic Filter or Smart Filter screen, depending on the data type of the field. When you click the Filter icon on a numeric field, a Basic Filter screen appears.

When you click the Filter icon on a date field, a different type of Basic Filter screen appears. When you click the Filter icon on a non-numeric field, a filtering window with Basic Filter and Smart Filter tabs appears.

See the Basic Filter and Smart Filter sections below for further instructions.

To clear a filter once it has been applied, click the Filter icon and click Clear in the filtering screen.

To delete all filters (even those in the Field Chooser), click the drop-down arrow next to Reset and click Clear Filters. This lets you delete filters without losing the fields themselves.

Basic Filter

The Basic Filter option is based on specific parameters that you can filter with, using AND or OR logical operators.

1. Click the Filter icon .
2. Click Basic Filter.
3. Enter a search term in the first box, then select a filtering option from the drop-down list. The choices are:
   - Contains - returns all records where the field contains the value entered
   - Does not contain - returns all records except those where the field contains the value entered
   - Is equal to - returns all records where the field contains exact matches for the value entered
   - Is not equal to - returns all records except those where the field contains exact matches for the value entered

  - Starts with - returns all records where the field value starts with the characters entered
  - Ends with - returns all records where the field value ends with the characters entered
  - Includes (',' delimited) - returns all records where the field contains any of the values in the comma delimited list
  - Excludes (',' delimited) - returns all records except those where the field contains any of the values in the comma delimited list
  - Is empty (A search term is unnecessary with this option.)
  - Is not empty (A search term is unnecessary with this option.)

4. (Optional) To enter a second filter criterion, select And or Or from the list. Then enter a search term in the first box, and select a filtering option from the drop-down list.
5. For date/time fields, you can click the Switch to relative date icon.
6. Then select a relative date option from the list and click Filter.

Smart Filter

Use the Smart Filter option to find records that contain a specific value that you enter.

1. Click the Filter icon .
2. Click Smart Filter.
3. Select individual check boxes to constrain the filtering to specific values. Or click Select All to filter within all available values.
4. (Optional) Type a search term in the Search box.
5. From the drop-down list, select to show items that Include selected values or Exclude selected values.
6. Select the Select All Search Results check box to select all records that match the search term.
7. Select the Add current selection to filter check box to apply the selected records to the filter.
8. Click Filter.

Filter Logic

Click the drop-down arrow next to Configure and click Filter Logic. The Filter Logic screen shows a list of fields that are currently filtered, and the order in which they are filtered.

You can modify the Boolean Logic as needed by changing the order of parentheses or logical operators.

The Scope field allows access to filters used for items that have group ownership. The default values are All <objects>, (for example, All Accounts), My <objects>, and My team's <objects>. Some objects such as Cases offer more options.

When finished, click Apply Logic.

Clicking Filter Logic also provides an easy way to find missing filtered fields.

Grouping Options

You can click and drag column titles to the position you want.

You can also drag and drop column headers to the Header field, which is the empty space above the grid:

This creates a new view that is organized by the field’s contents.

It only groups records that are displayed on the current page, not the entire record set. You can increase the Items per page at the bottom of the grid, if needed.

You can add any columns from the Field Chooser to the Header field. You can also create nested views and filter each column using its drop-down button.

Grouping by numeric fields displays footers that sum the numeric values.

Sort Records in Conga Grid

To sort records by a single column, click inside the field header. This sorts records by close date, in ascending order. An arrow appears in the field header.

Click inside the field header again to sort by descending order. Click again to reset the default order.

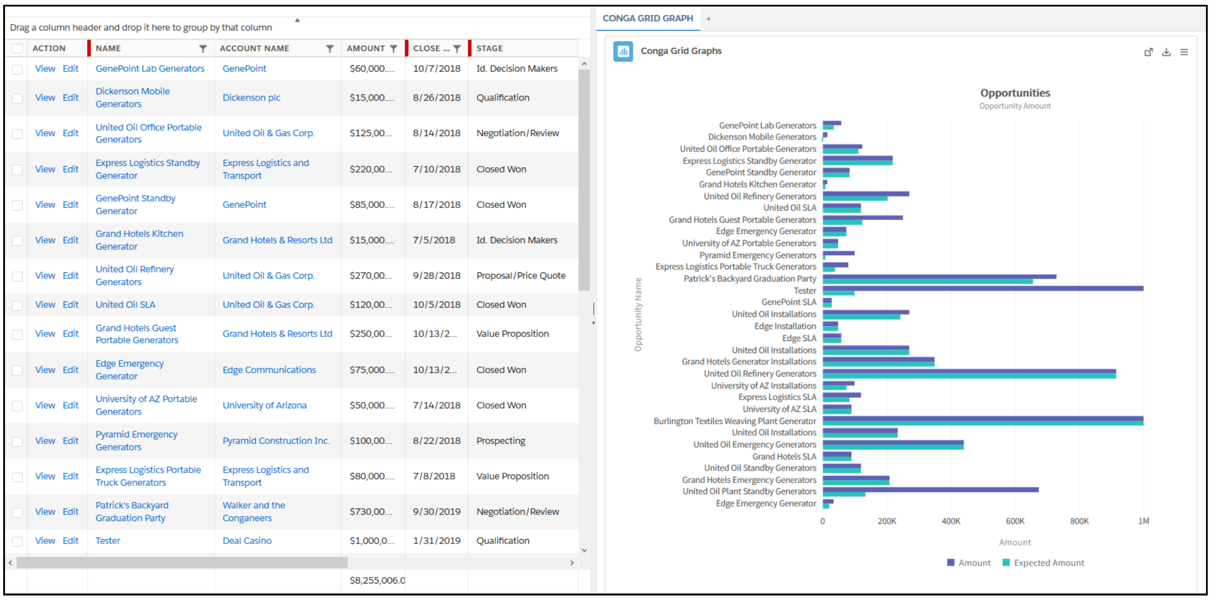
Once you have sorted a single field, you can continue to sort by additional fields. Right-click in another field and click **Add Ascending Sort** or **Add Descending Sort** to sort the records in that field.

> Left-clicking in any field header removes all nested sorting that you have built in other fields, while sorting only that field.

Conga Grid Graphs

Use Conga Grid Graphs to visually analyze data in every Grid with custom graphs.

> ⚠ Conga Grid Graphs requires Conga Grid version 2.58 or higher.



Drag a column header and drop it here to group by that column

□	ACTION	NAME	ACCOUNT NAME	AMOUNT	CLOSE ...	STAGE
□	View Edit	GenePoint Lab Generators	GenePoint	$60,000....	10/7/2018	Id. Decision Makers
□	View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....	8/26/2018	Qualification
□	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,00...	8/14/2018	Negotiation/Review
□	View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,00...	7/10/2018	Closed Won
□	View Edit	GenePoint Standby Generator	GenePoint	$85,000....	8/17/2018	Closed Won
□	View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....	7/5/2018	Id. Decision Makers
□	View Edit	United Oil Refinery Generators	United Oil & Gas Corp.	$270,00...	9/28/2018	Proposal/Price Quote
□	View Edit	United Oil SLA	United Oil & Gas Corp.	$120,00...	10/5/2018	Closed Won
□	View Edit	Grand Hotels Guest Portable Generators	Grand Hotels & Resorts Ltd	$250,00...	10/13/2...	Value Proposition
□	View Edit	Edge Emergency Generator	Edge Communications	$75,000....	10/13/2...	Closed Won
□	View Edit	University of AZ Portable Generators	University of Arizona	$50,000....	7/14/2018	Closed Won
□	View Edit	Pyramid Emergency Generators	Pyramid Construction Inc.	$100,00...	8/22/2018	Prospecting
□	View Edit	Express Logistics Portable Truck Generators	Express Logistics and Transport	$80,000....	7/8/2018	Value Proposition
□	View Edit	Patrick's Backyard Graduation Party	Walker and the Conganeers	$730,00...	9/30/2019	Negotiation/Review
□	View Edit	Tester	Deal Casino	$1,000,0...	1/31/2019	Qualification
				$8,255,006.0		

CONGA GRID GRAPH
+
Conga Grid Graphs
Opportunities
Opportunity Amount
Opportunity Name
GenePoint Lab Generators
Dickenson Mobile Generators
United Oil Office Portable Generators
Express Logistics Standby Generator
GenePoint Standby Generator
Grand Hotels Kitchen Generator
United Oil Refinery Generators
United Oil SLA
Grand Hotels Guest Portable Generators
Edge Emergency Generator
University of AZ Portable Generators
Pyramid Emergency Generators
Express Logistics Portable Truck Generators
Patrick's Backyard Graduation Party
Tester
GenePoint SLA
United Oil Installations
Edge Installation
Edge SLA
United Oil Installations
Grand Hotels Generator Installations
United Oil Refinery Generators
University of AZ Installations
Express Logistics SLA
University of AZ SLA
Burlington Textiles Weaving Plant Generator
United Oil Installations
United Oil Emergency Generators
Grand Hotels SLA
United Oil Standby Generators
Grand Hotels Emergency Generators
United Oil Plant Standby Generators
Edge Emergency Generator
0
200K
400K
600K
800K
1M
Amount
Amount
Expected Amount

Users can easily add a Conga Grid Graph in the Reading Pane section of any Grid. For more information on how to setup and use Conga Grid Graphs, see Setting Up Conga Grid Graphs.

Graph Types

Conga Grid Graphs offers the following graph types:

- Bar Chart
  - Bar (horizontal)
  - Column (vertical)
  - Stacked Bar
  - Stacked Column
- Pie Chart
  - Donut
  - Pie
- Line Chart
  - Line (left to right)
  - Area
- Other Charts
  - Funnel
  - Gauge
  - Scatter Plot

For more detailed information on each graph type, see Conga Grid Graph Types.

Conga Grid Graph Features

Conga Grid Graphs allows users to select any field as the Category and one or multiple fields as the Series.

A Caption and Sub Caption are available for each individual graph to provide users with labels and additional context. Conga Grid Graphs also provides functionality to customize the labels and positioning for the X Axis, Y Axis, and Legend of each graph.

Users can also customize the Theme of their graph using Light, Pastel, and Dark themes.

Exporting a Grid Graph

Conga Grid also features the ability to easily export Conga Grid Graphs in the following files:

- PNG
- PDF
- SVG

> ⚠ For more information on how to download a graph, see Download a Conga Grid Graph.

Setting Up Conga Grid Graphs

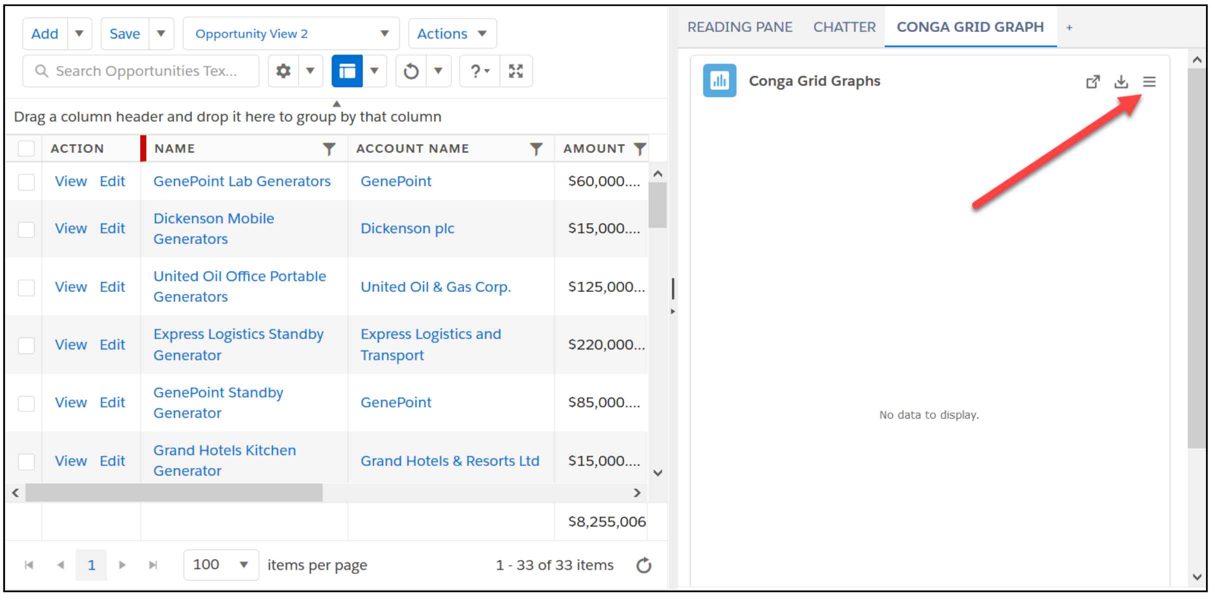
Perform the following steps to set up Conga Grid Graphs:

1. Upgrade to Conga Grid version 2.58 or the current version available on the Salesforce AppExchange.
2. Enable Grid Graphs in Conga Grid Setup.
   a. Navigate to the Conga Grid Setup tab.
   b. Click **Global Defaults** or a specific object to allow users to use Conga Grid Graphs on.
   c. Click **Feature Security**.
   d. Select the **Enable** radio button for the Conga Grid Graph feature and choose whether to apply to All Users, specific Profiles, or specific Users.
3. Display the Reading Pane in the selected Grid View to begin building a Conga Grid Graph.

   > ✓ **Important**
   >
   > The Reading Pane is where Conga Grids displays the graph.

Easily add a Conga Grid Graph to your Grids to provide users with a visual analysis

1. Navigate to a Grid.
2. Click the gear wheel icon to open the Grid Configuration options.
3. Click the arrow next to Reading Pane to display the available features.
4. Check the check-box field next to Conga Grid Graph Tab.
5. Open up the Reading Pane in the Grid and click the Conga Grid Graph tab.
6. Click the Conga Grid Graph Settings icon.



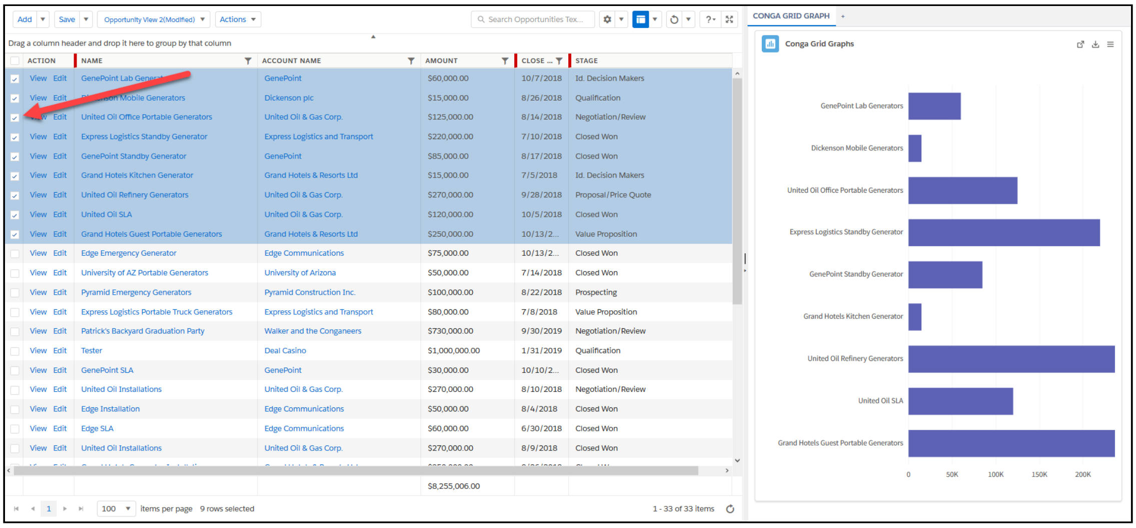
Drag a column header and drop it here to group by that column

	ACTION	NAME	ACCOUNT NAME	AMOUNT
□	View Edit	GenePoint Lab Generators	GenePoint	$60,000....
□	View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....
□	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,000...
□	View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,000...
□	View Edit	GenePoint Standby Generator	GenePoint	$85,000....
□	View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....
				$8,255,006

7. Select a Graph Type under the Settings tab. For more information on the different Graph Types, see Conga Grid Graph Types.
8. Click the Data tab and select a field for the Category and select a field(s) for the Series.
   Note: Users can select multiple fields depending on the graph type. For more information on field requirements for each graph type, see Conga Grid Graph Types.
9. (Optional) Click the Format tab to add additional formatting to the Conga Grid Graph.
   a. Caption: adds entered text as a caption on the graph
   b. Sub Caption: adds entered text as a sub caption on the graph
   c. Caption Alignment: determines the caption alignment
   d. X Axis Name: adds entered text as the X Axis Label
   e. Y Axis Name: adds entered text as the Y Axis Label
   f. Show Legend: displays a legend on the graph
   g. Legend Caption: displays entered text as a caption for the legend
   h. Legend Position: determines position of the legend
10. (Optional) Click the Theme tab to select a theme for the Conga Grid Graph.

   a. Dark Mode: enables Dark Mode for the Conga Grid Graph user interface
   b. Click one of the Light, Pastel or Dark themes to apply the theme to the Conga Grid Graph.

11. Click the Settings icon in the Conga Grid Graph window once all configurations are selected.
12. Select records to display in a graph.
    - Select specific records using the check box fields to left of each record.

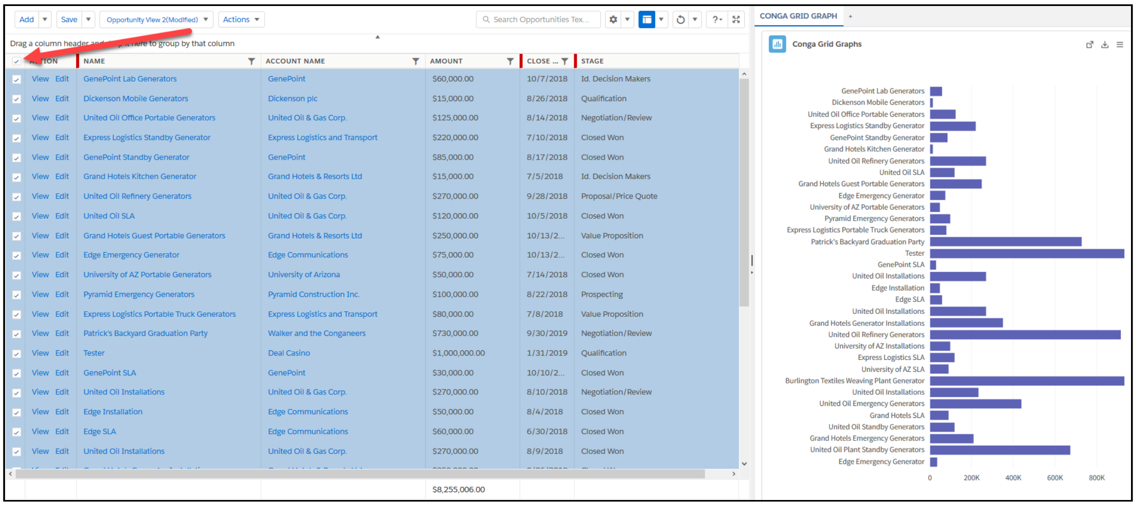

Add
Save
Opportunity View 2(Modified)
Actions
Search Opportunities Tex...
CONGA GRID GRAPH
+
Conga Grid Graphs
GenePoint Lab Generators
Dickenson Mobile Generators
United Oil Office Portable Generators
Express Logistics Standby Generator
GenePoint Standby Generator
Grand Hotels Kitchen Generator
United Oil Refinery Generators
United Oil SLA
Grand Hotels Guest Portable Generators
0
50K
100K
150K
200K

Drag a column header and drop it here to group by that column

□	ACTION	NAME	ACCOUNT NAME	AMOUNT	CLOSE ...	STAGE
☑	View Edit	GenePoint Lab Generators	GenePoint	$60,000.00	10/7/2018	Id. Decision Makers
☑	View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000.00	8/26/2018	Qualification
☑	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,000.00	8/14/2018	Negotiation/Review
☑	View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,000.00	7/10/2018	Closed Won
☑	View Edit	GenePoint Standby Generator	GenePoint	$85,000.00	8/17/2018	Closed Won
☑	View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000.00	7/5/2018	Id. Decision Makers
☑	View Edit	United Oil Refinery Generators	United Oil & Gas Corp.	$270,000.00	9/28/2018	Proposal/Price Quote
☑	View Edit	United Oil SLA	United Oil & Gas Corp.	$120,000.00	10/5/2018	Closed Won
☑	View Edit	Grand Hotels Guest Portable Generators	Grand Hotels & Resorts Ltd	$250,000.00	10/13/2...	Value Proposition
□	View Edit	Edge Emergency Generator	Edge Communications	$75,000.00	10/13/2...	Closed Won
□	View Edit	University of AZ Portable Generators	University of Arizona	$50,000.00	7/14/2018	Closed Won
□	View Edit	Pyramid Emergency Generators	Pyramid Construction Inc.	$100,000.00	8/22/2018	Prospecting
□	View Edit	Express Logistics Portable Truck Generators	Express Logistics and Transport	$80,000.00	7/8/2018	Value Proposition
□	View Edit	Patrick's Backyard Graduation Party	Walker and the Conganeers	$730,000.00	9/30/2019	Negotiation/Review
□	View Edit	Tester	Deal Casino	$1,000,000.00	1/31/2019	Qualification
□	View Edit	GenePoint SLA	GenePoint	$30,000.00	10/10/2...	Closed Won
□	View Edit	United Oil Installations	United Oil & Gas Corp.	$270,000.00	8/10/2018	Negotiation/Review
□	View Edit	Edge Installation	Edge Communications	$50,000.00	8/4/2018	Closed Won
□	View Edit	Edge SLA	Edge Communications	$60,000.00	6/30/2018	Closed Won
□	View Edit	United Oil Installations	United Oil & Gas Corp.	$270,000.00	8/9/2018	Closed Won
				$8,255,006.00		

1 | 100 items per page | 9 rows selected | 1 - 33 of 33 items

    - Select all records in the Grid using the All Records checkbox.


Add
Save
Opportunity View 2(Modified)
Actions
Search Opportunities Tex...
CONGA GRID GRAPH
+
Conga Grid Graphs
GenePoint Lab Generators
Dickenson Mobile Generators
United Oil Office Portable Generators
Express Logistics Standby Generator
GenePoint Standby Generator
Grand Hotels Kitchen Generator
United Oil Refinery Generators
United Oil SLA
Grand Hotels Guest Portable Generators
Edge Emergency Generator
University of AZ Portable Generators
Pyramid Emergency Generators
Express Logistics Portable Truck Generators
Patrick's Backyard Graduation Party
Tester
GenePoint SLA
United Oil Installations
Edge Installation
Edge SLA
United Oil Installations
Grand Hotels Generator Installations
United Oil Refinery Generators
University of AZ Installations
Express Logistics SLA
University of AZ SLA
Burlington Textiles Weaving Plant Generator
United Oil Installations
United Oil Emergency Generators
Grand Hotels SLA
United Oil Standby Generators
Grand Hotels Emergency Generators
United Oil Plant Standby Generators
Edge Emergency Generator
0
200K
400K
600K
800K

Drag a column header and drop it here to group by that column

☑	ACTION	NAME	ACCOUNT NAME	AMOUNT	CLOSE ...	STAGE
☑	View Edit	GenePoint Lab Generators	GenePoint	$60,000.00	10/7/2018	Id. Decision Makers
☑	View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000.00	8/26/2018	Qualification
☑	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,000.00	8/14/2018	Negotiation/Review
☑	View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,000.00	7/10/2018	Closed Won
☑	View Edit	GenePoint Standby Generator	GenePoint	$85,000.00	8/17/2018	Closed Won
☑	View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000.00	7/5/2018	Id. Decision Makers
☑	View Edit	United Oil Refinery Generators	United Oil & Gas Corp.	$270,000.00	9/28/2018	Proposal/Price Quote
☑	View Edit	United Oil SLA	United Oil & Gas Corp.	$120,000.00	10/5/2018	Closed Won
☑	View Edit	Grand Hotels Guest Portable Generators	Grand Hotels & Resorts Ltd	$250,000.00	10/13/2...	Value Proposition
☑	View Edit	Edge Emergency Generator	Edge Communications	$75,000.00	10/13/2...	Closed Won
☑	View Edit	University of AZ Portable Generators	University of Arizona	$50,000.00	7/14/2018	Closed Won
☑	View Edit	Pyramid Emergency Generators	Pyramid Construction Inc.	$100,000.00	8/22/2018	Prospecting
☑	View Edit	Express Logistics Portable Truck Generators	Express Logistics and Transport	$80,000.00	7/8/2018	Value Proposition
☑	View Edit	Patrick's Backyard Graduation Party	Walker and the Conganeers	$730,000.00	9/30/2019	Negotiation/Review
☑	View Edit	Tester	Deal Casino	$1,000,000.00	1/31/2019	Qualification
☑	View Edit	GenePoint SLA	GenePoint	$30,000.00	10/10/2...	Closed Won
☑	View Edit	United Oil Installations	United Oil & Gas Corp.	$270,000.00	8/10/2018	Negotiation/Review
☑	View Edit	Edge Installation	Edge Communications	$50,000.00	8/4/2018	Closed Won
☑	View Edit	Edge SLA	Edge Communications	$60,000.00	6/30/2018	Closed Won
☑	View Edit	United Oil Installations	United Oil & Gas Corp.	$270,000.00	8/9/2018	Closed Won
				$8,255,006.00		

13. Save the View to display the Conga Grid Graph.
    - Click Save to save the Conga Grid Graph to the existing View.
    - Click the Save dropdown arrow and then select Save View As to save a new View that displays the Conga Grid Graph.

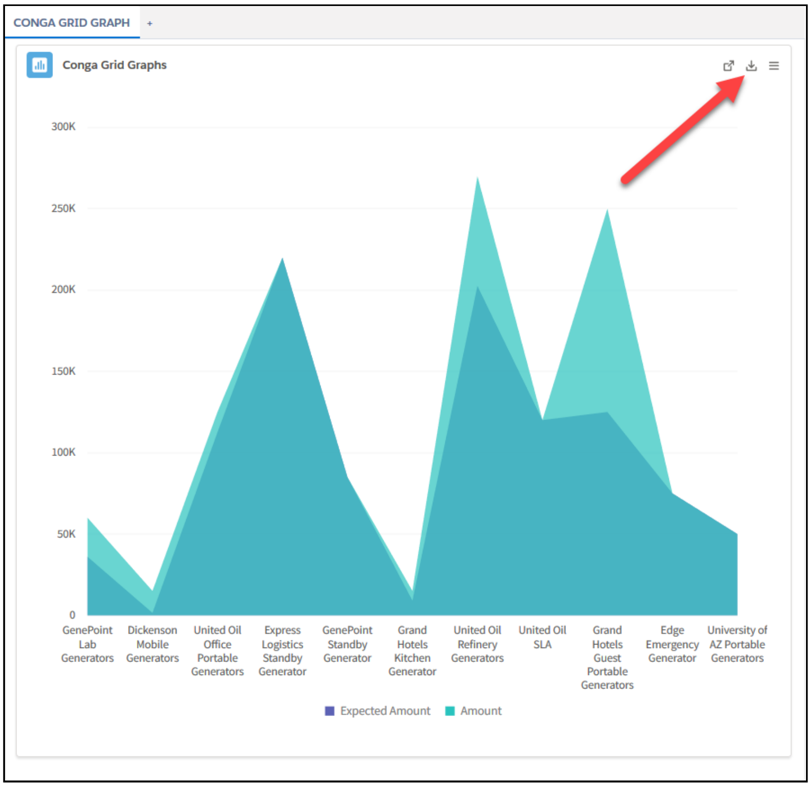
For information on downloading a Conga Grid Graph, see Download a Conga Grid Graph.

Download a Conga Grid Graph

Download a Conga Grid Graph as a PDF, PNG, or SVG file to easily share with stakeholders or embed it within documents.

To download a Conga Grid Graph

1. Navigate to a View with a Conga Grid Graph.
2. Click the **Download icon** at the top right corner of the graph.


CONGA GRID GRAPH
+
Conga Grid Graphs
300K
250K
200K
150K
100K
50K
0
GenePoint Lab Generators
Dickenson Mobile Generators
United Oil Office Portable Generators
Express Logistics Standby Generator
GenePoint Standby Generator
Grand Hotels Kitchen Generator
United Oil Refinery Generators
United Oil SLA
Grand Hotels Guest Portable Generators
Edge Emergency Generator
University of AZ Portable Generators
Expected Amount
Amount

3. Select a download file type from the listed options.
   - Download as PDF
   - Download as PNG
   - Download as SVG

The Conga Grid Graph then downloads as the selected file.

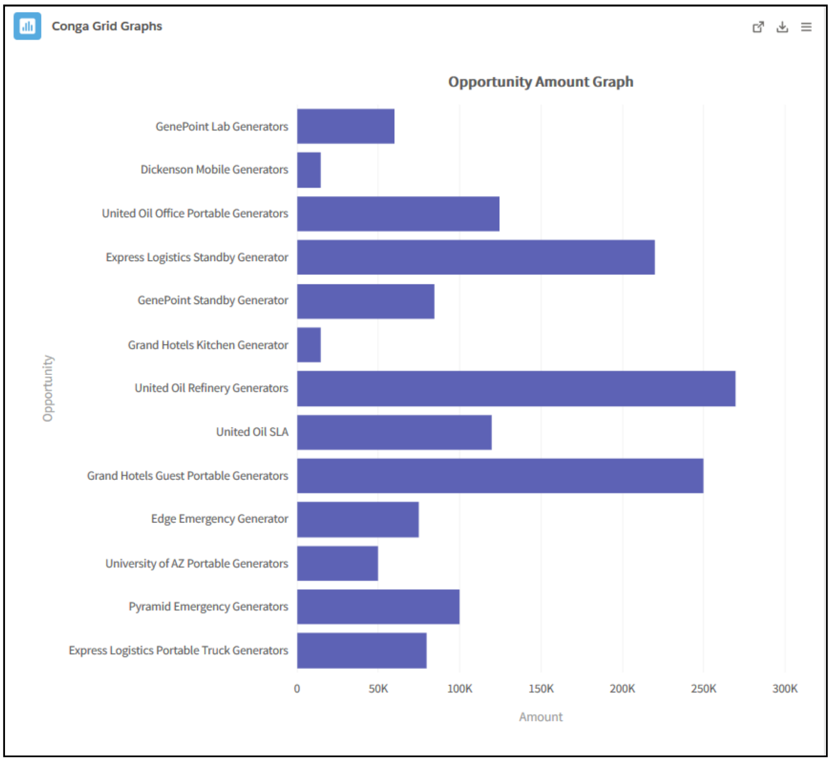
Conga Grid Graph Types

Bar Graphs

Important

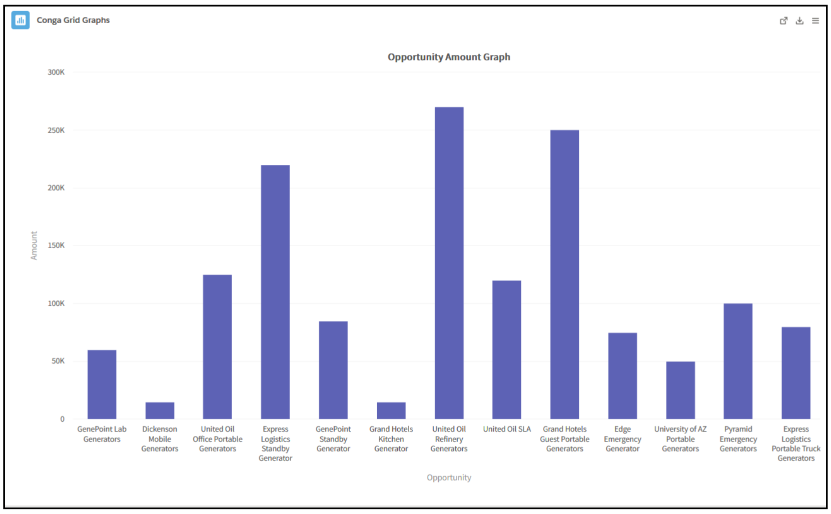
Bar Graphs in Conga Grid allow users to select one field as a Category and one or multiple fields as the Series.

- **Bar**: A Bar graph displays quantitative data and is useful for comparing data points. It consists of horizontal bars with lengths proportional to the data values they represent. One axis of a bar chart graphs a Category and the other axis portrays the data values.
    - **Example**:

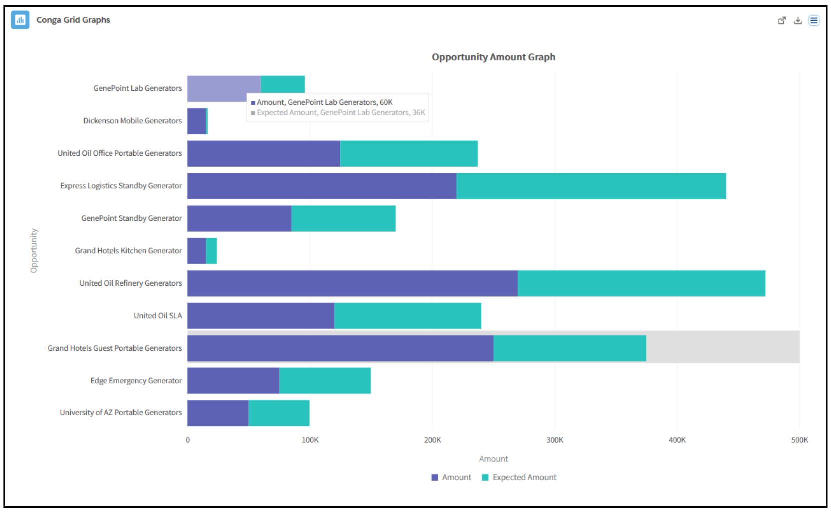

Conga Grid Graphs
Opportunity Amount Graph
GenePoint Lab Generators
Dickenson Mobile Generators
United Oil Office Portable Generators
Express Logistics Standby Generator
GenePoint Standby Generator
Grand Hotels Kitchen Generator
United Oil Refinery Generators
United Oil SLA
Grand Hotels Guest Portable Generators
Edge Emergency Generator
University of AZ Portable Generators
Pyramid Emergency Generators
Express Logistics Portable Truck Generators
Opportunity
0
50K
100K
150K
200K
250K
300K
Amount

- **Column**: A Column graph is useful for comparing data values of related categories in a vertical fashion. The value of a Category is represented by an individual vertical column. It is also useful to compare and contrast data over time.

- Example:

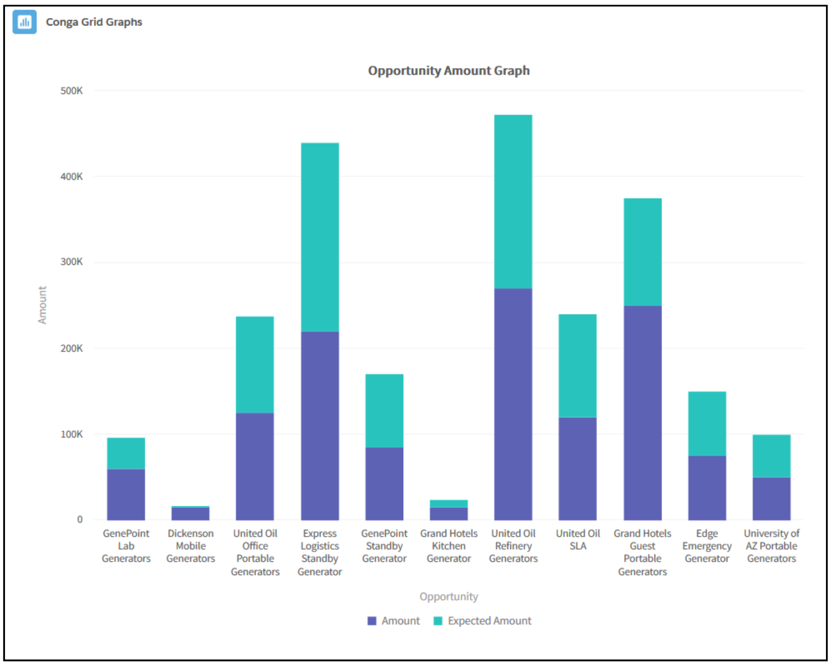


- **Stacked Bar**: Stacked Bar graphs are useful for portraying how much a Category contributed to its Category's total. This type of graph is also helpful in comparing sub-groups with similar sub-groups of other categories, which standard bar chart cannot portray as clearly.
  - Example:



- **Stacked Column**: A Stacked Column graph portrays the extent of a record's specific contribution to a total data value. It is use for breaking down and comparing multiple metrics against each record.

- Example:



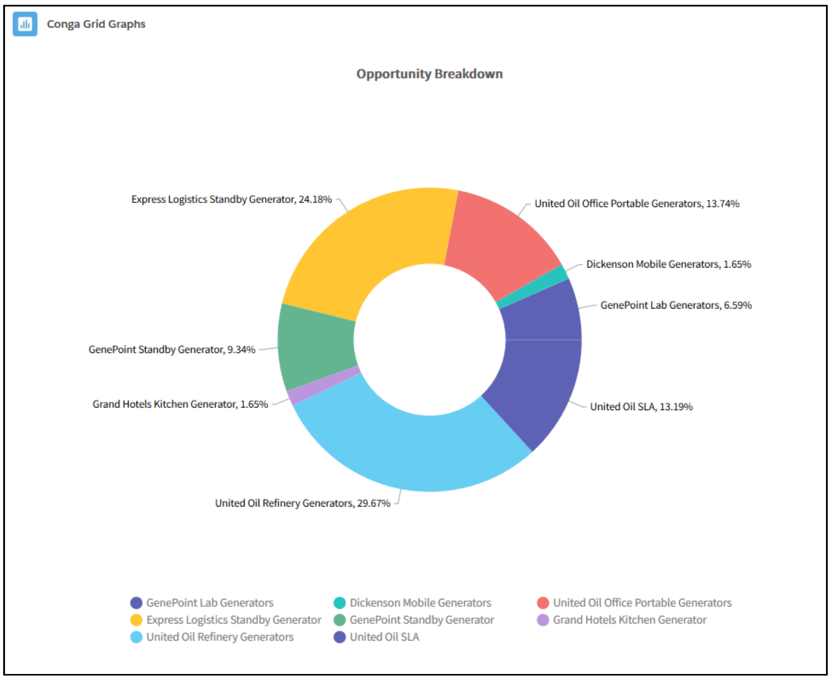
Pie Graphs

Important

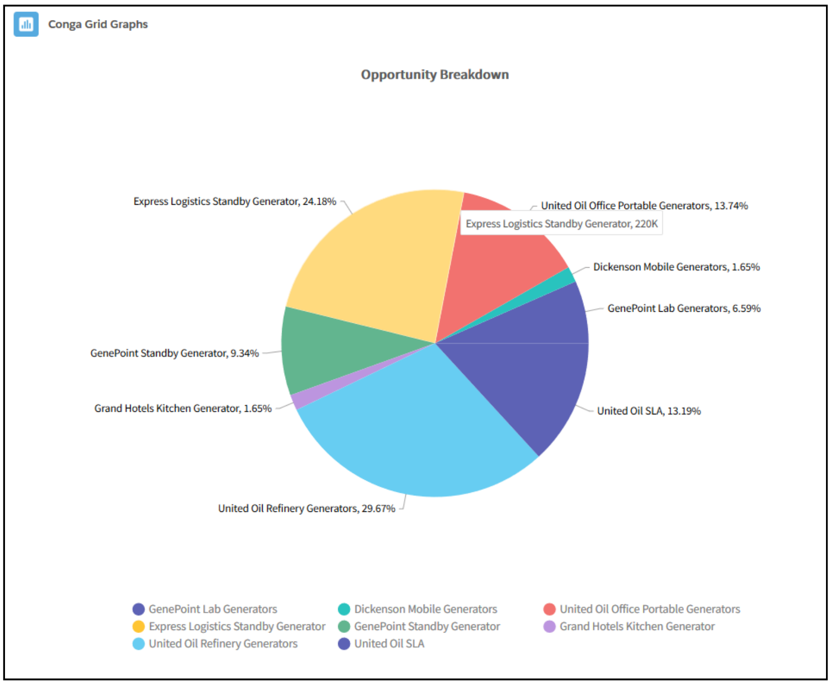
Pie Graphs in Conga Grid allow users to select one field as a Category and one field as the Series.

- **Donut**: The Donut graph contains a blank circle in the middle and is useful for emphasizing each record's contribution to an entire total.

- Example:



- **Pie**: The Pie graph is a circle divided into categories, each representing a contribution to an overall total. It is useful to display the percentage or contribution of each record.
  - Example:

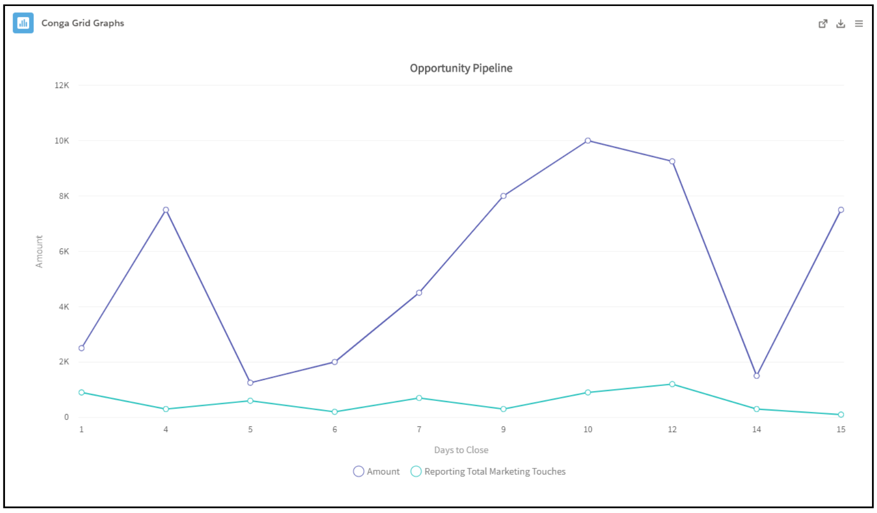


Line Graphs

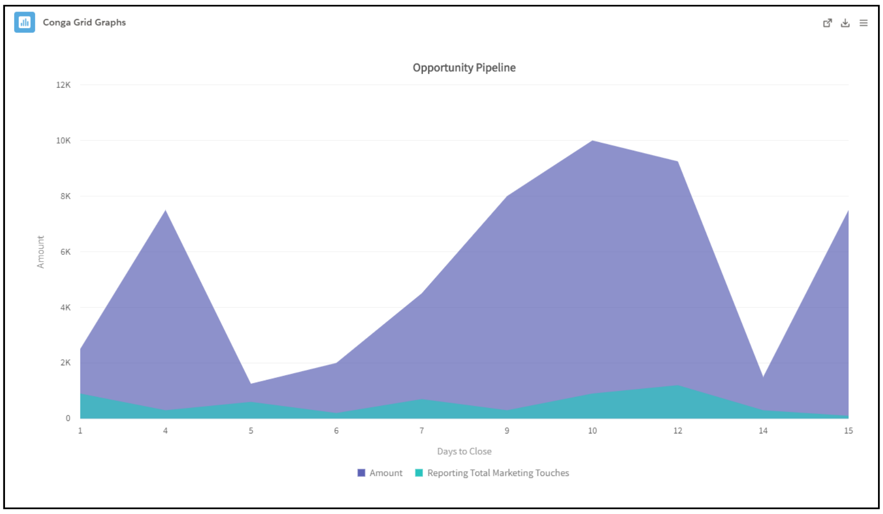
Important

Line Graphs in Conga Grid allow users to select one field as a Category and one or multiple fields as the Series.

- **Line**: The Line graph displays trends and relationships for multiple data points. It uses connected line segments to portray multiple data points. Line graphs are useful for displaying data trends over a time period.
  - **Example**:



- **Area**: The Area graph displays data values of multiple fields connected with line segments. It is useful for portraying data trends and changes over time.
  - **Example**:

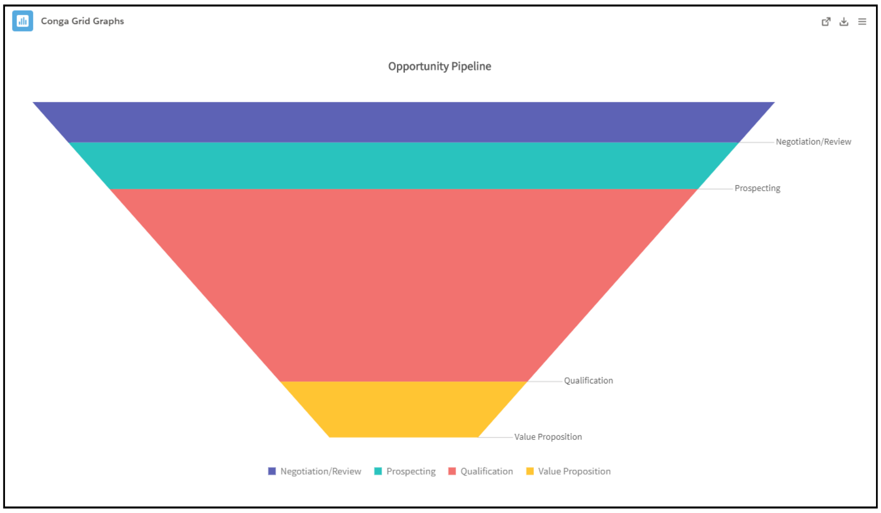


Other Graphs

Important

Funnels in Conga Grid allow users to select one field as a Category and one field as the Series.

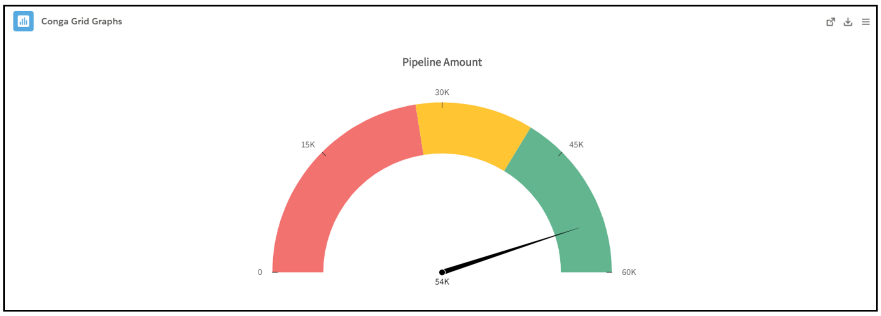
- **Funnel**: A Funnel displays data effectively throughout an entire process. It measures and portrays data during specific phases in a funnel shape with multiple segments.
  - **Example**:



Important

A Gauge in Conga Grid allows users to measure records by one field and identify up to three Segment Ranges by specific values.

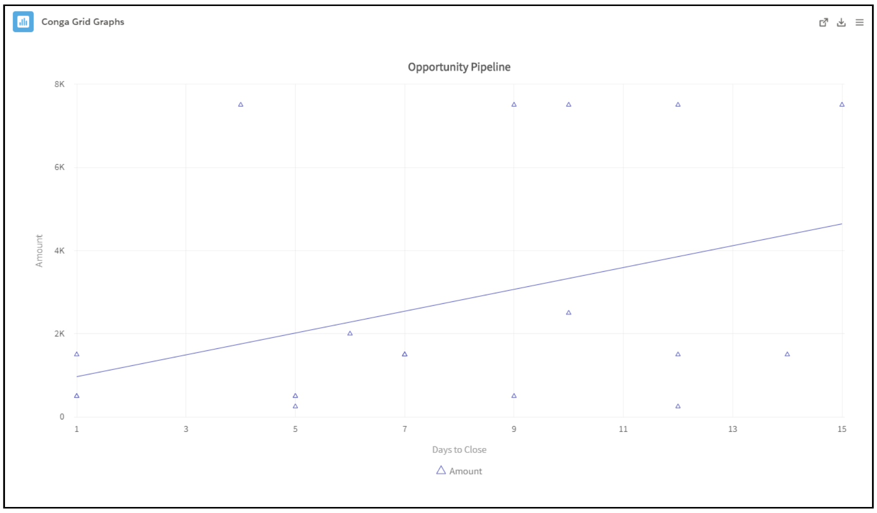
- **Gauge**: A Gauge displays data in a progressive manner on a radial scale. The dial in a Gauge indicates the progress towards a designated data point. Gauges are useful for tracking "% to Goal" metrics and other metrics typically displayed on sales dashboards.
  - **Example**:



Important

A Scatter Plot in Conga Grid allows users to measure records by one Category and a minimum of two Series.

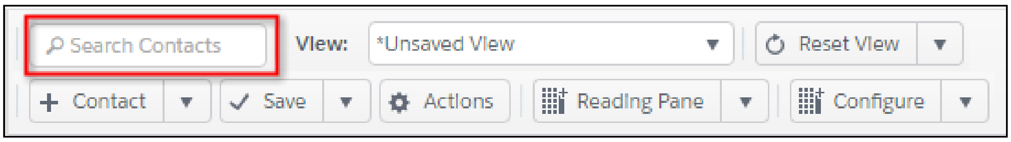
- **Scatter Plot**: The Scatter Plot displays the relationship between data points by plotting coordinates on a chart. Scatter Plots are useful for comparing trends against one another and determining how certain field values influence one another.
  - **Example**:



For more information on the different types of graphs, see Fusion Charts documentation.

Search Records

The Search box works similarly to the Salesforce® global search.



There are some exceptions:

- For a specific object with no Conga Grid℠ filters applied, the search returns only the first 2,000 records displayed in the current view. It does not return the entire data set.
- For a specific object with a filter applied, the search initially ignores the filter and finds the first 2,000 records displayed in the current view. That result set is passed into a SOQL search to further filter the Conga Grid filters against that set of records.
- For a Conga Grid page layout, the search returns the first 2,000 records displayed in the current view. By default, it passes in the foreign key of the record that you are currently viewing.

- It searches text fields associated with a given object, even if the fields are not displayed in the grid. This does not include Lookups.
- It searches both the body and subject of emails.
- It does not support boolean strings like AND/OR logic, for example: Sales and Transportation

See the following Salesforce documents for more details on search constraints, including which fields cannot be searched:

- About SOSL
- SOSL Limits

Conditional Formatting

You can format your grid based on conditions and rules.

To apply conditional formatting rules:

1. Click the drop-down arrow next to Configure and click Conditional Formatting.
2. In the Conditional Formatting Rules screen, click New Rule.
3. Enter a Rule Name.
4. Select a Field from which to apply the rule.
5. Select a Cell Color and Text Color.
6. Click a Font Style (Bold, Italics, and Underline).
7. Select the Apply to Row check box if the rule should apply to the entire row where the field exists.
8. Click Basic Rule to apply the rule using criteria such as Contains, Does Not Contain, and others. Or, click Advanced to apply the rule using JavaScript code.
9. Click Apply Rules.

Here are some other tips for using conditional formatting:

- To save a conditional format rule, save the current view.
- To delete a conditional format rule, click the drop-down arrow next to Configure and select Conditional Formatting. Select the rule and click Delete Rule.
- To remove all formatting rules, click the drop-down arrow next to Reset View and click Clear Formatting.
- You can hover over a cell with conditional formatting to show the applied rule.
- The conditional formatting legend shows at the bottom of a view next to the page size selector
- The conditional formatting legend can be controlled in Conga Grid setup via a Feature Security Parameter called "Conditional Formatting Legend"

- The legend only shows if there are conditional formatting rules applied to the view
- Clicking on the legend displays a quick overview of the different rules being applied to the view
- Hovering over a rule name will display which field the rule is applied to and the logic used in the rule
- Clicking on a rule name will toggle the formatting on or off, but will not remove it from the view completely

On the Advanced tab, you can use advanced conditional formats of a date range. For example:

ActivityDate < (new Date((Date.today()))) && ActivityDate > (new Date((Date.today()))).addDays(-14)

Reading Pane

The Reading Pane is a separate panel that displays detailed information about a selected record.

Click Reading Pane in the Conga Grid℠ toolbar to display the Reading Pane on the right side of the grid. Or, click the drop-down arrow next to Reading Pane and select a position to display it within the grid. The options are Top, Right, Bottom, or Left.

To help organize the information to view in the Reading Pane, you can add and configure different tabs by following these steps:

1. Click the small arrow on the right side of the grid. The Configure menu appears.
2. Click the drop-down arrow next to Reading Pane to show the default tabs that are available.
3. Select the check boxes of tabs that you want to display in the Reading Pane. These options are described in more detail next.

Reading Pane Tab

This tab is enabled by default. It contains a long editable text field to capture details or e-mail conversations related to the selected record.

The drop-down list within the Reading Pane tab shows fields that are set to Text or Rich Text. Select a field and enter your own text related to that field.

Chatter Tab

This tab is enabled by default. It contains Salesforce® Chatter messages related to the selected record.

Detail Grid Tab

Use this tab to view multiple fields (and their values) for the selected record. This is a convenient alternative to displaying many fields in the grid, where you would need to scroll horizontally to view all of them. Instead, select the Detail Grid Tab check box, then select the fields that you want to view in the Detail Grid tab.

Chart Tab

Use this tab to create a quick graph of a selected numeric field for all of the records on the current page. Select the Chart Tab check box, then select a numeric field to graph in the Chart tab.

Tabs Based on Visualforce Pages

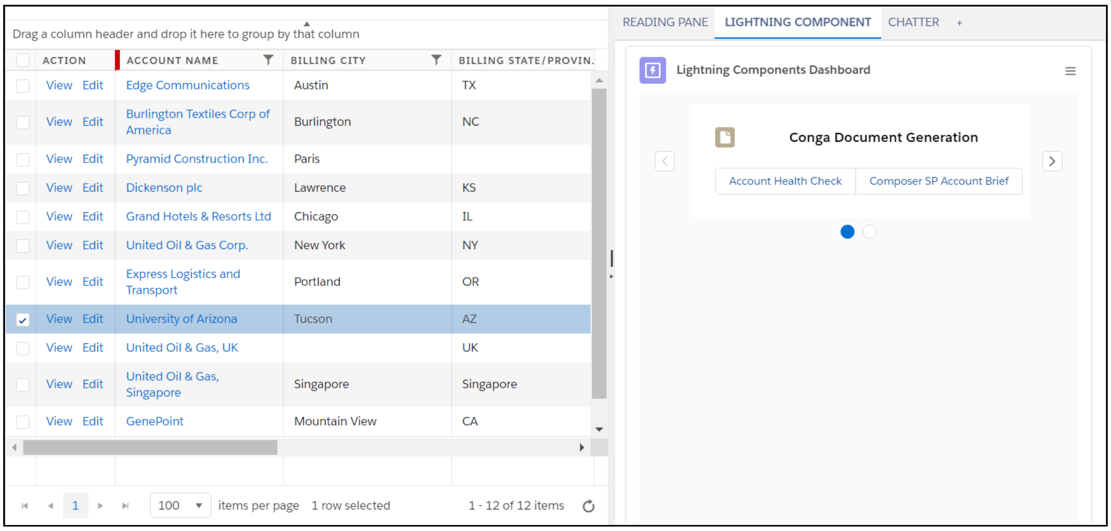
You can add tabs to the Reading Pane that are based on Visualforce pages. Click the drop-down arrow next to Visualforce Pages, then select the pages you want to display as tabs.

You can even create a Visualforce page with a Multi-Tabular Conga Grid so that you can view related details in one convenient location within the Reading Pane.

Lightning Components and Lightning Web Components in the Reading Pane

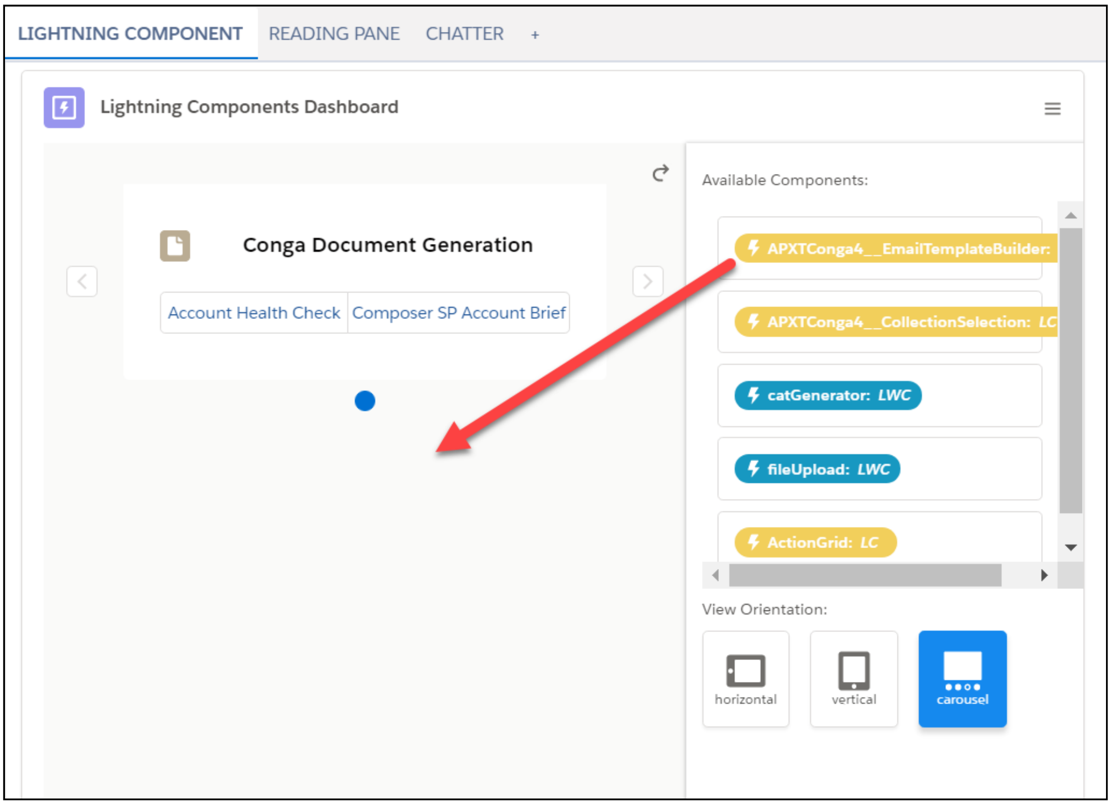
Conga Grid allows administrators to add Lightning Components and Lightning Web Components in the Reading Pane.

This functionality lets Conga Grid users view and access data for multiple records at once, and use Lightning Components in one centralized location.


Drag a column header and drop it here to group by that column
READING PANE
LIGHTNING COMPONENT
CHATTER
+
Lightning Components Dashboard
Conga Document Generation
Account Health Check
Composer SP Account Brief
100
items per page
1 row selected
1 - 12 of 12 items

	ACTION	ACCOUNT NAME	BILLING CITY	BILLING STATE/PROVIN.
□	View Edit	Edge Communications	Austin	TX
□	View Edit	Burlington Textiles Corp of America	Burlington	NC
□	View Edit	Pyramid Construction Inc.	Paris	
□	View Edit	Dickenson plc	Lawrence	KS
□	View Edit	Grand Hotels & Resorts Ltd	Chicago	IL
□	View Edit	United Oil & Gas Corp.	New York	NY
□	View Edit	Express Logistics and Transport	Portland	OR
☑	View Edit	University of Arizona	Tucson	AZ
□	View Edit	United Oil & Gas, UK		UK
□	View Edit	United Oil & Gas, Singapore	Singapore	Singapore
□	View Edit	GenePoint	Mountain View	CA

Administrators can add multiple customs or managed Lightning Components and Lightning Web Components by dragging and dropping them into the Lightning Component section of the Reading Pane.



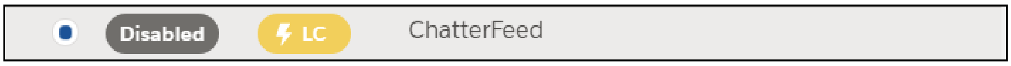
Additionally, administrators can customize the display orientation of Lightning Components and Lightning Web Components, as well as preview how they will look in the Reading Pane before saving any changes. Grid also provides administrators the option to set additional Component attributes with custom JSON code.

To setup Lightning Components and Lightning Web Components in Conga Grid, see Setting Up Lightning Components and Lightning Web Components in Conga Grid.

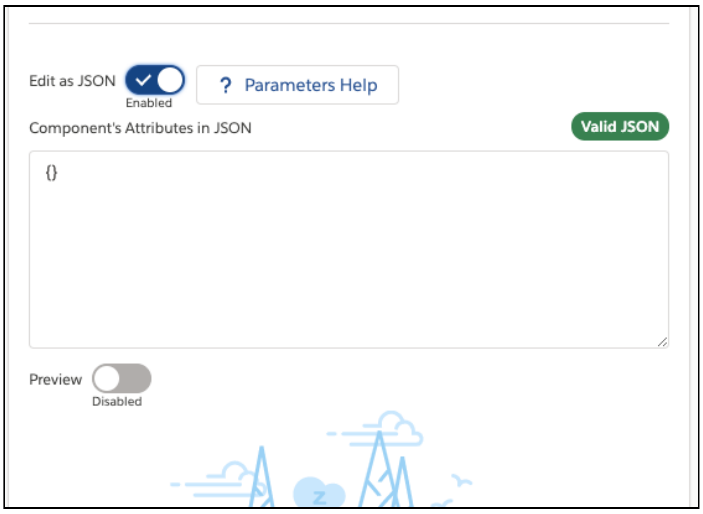
Setting Up Lightning Components and Lightning Web Components in Conga Grid

Enable and Setup Lightning Components and Lightning Web Components to use them in the Reading Pane of your grids.

1. Enable the Lightning Component Feature in Conga Grid Setup.
    a. Navigate to the Conga Grid Setup tab.
    b. Select Global Defaults or a specific object to enable the Lightning Component Feature on.
    c. Click the Feature Security tab.
    d. Locate the Lightning Component Feature and select Enable to enable the feature.
    e. Click Save.
2. Enable specific Lightning Components and Lightning Web Components in Conga Grid Setup.
    a. Navigate to the Conga Grid Setup tab.
    b. Select Global Defaults or a specific object to enable Lightning Components and Lightning Web Components on.
    c. Click the radio button next a specific Component.



    d. Change the Component Enabled toggle from Disabled to Enabled. This enables the individual Component.
    e. (Optional) Enable the Edit as JSON feature to set additional Component attributes in JSON.



Edit as JSON ✓ Enabled

Component's Attributes in JSON

```
{}
```

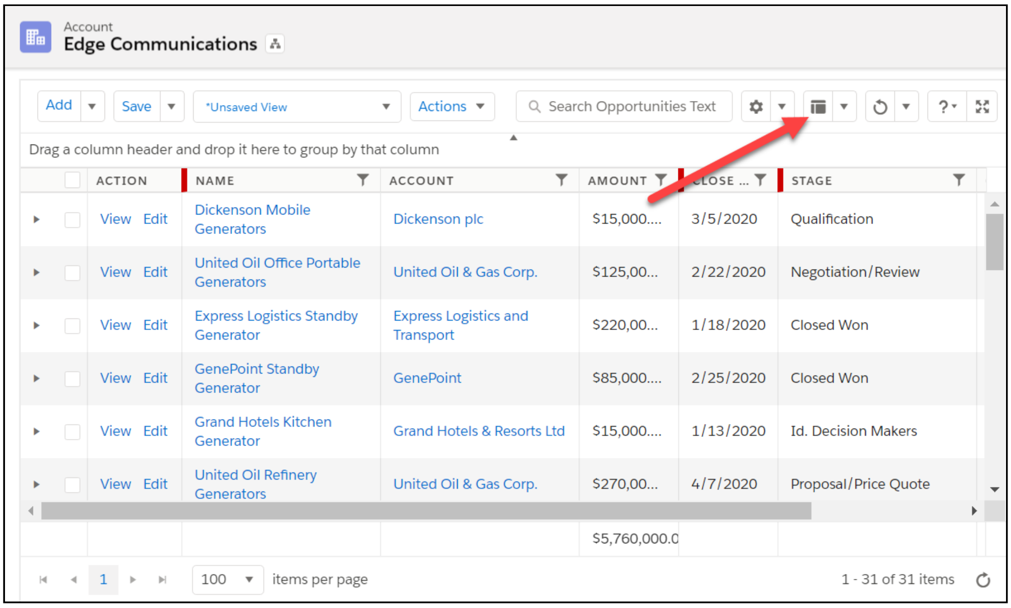
Preview Disabled

> ⚠ If enabled, administrators and developers can add their custom JSON code in the Component's Attributes in JSON box.

f. (Optional) Enable the Preview feature to display a preview of the Component below.

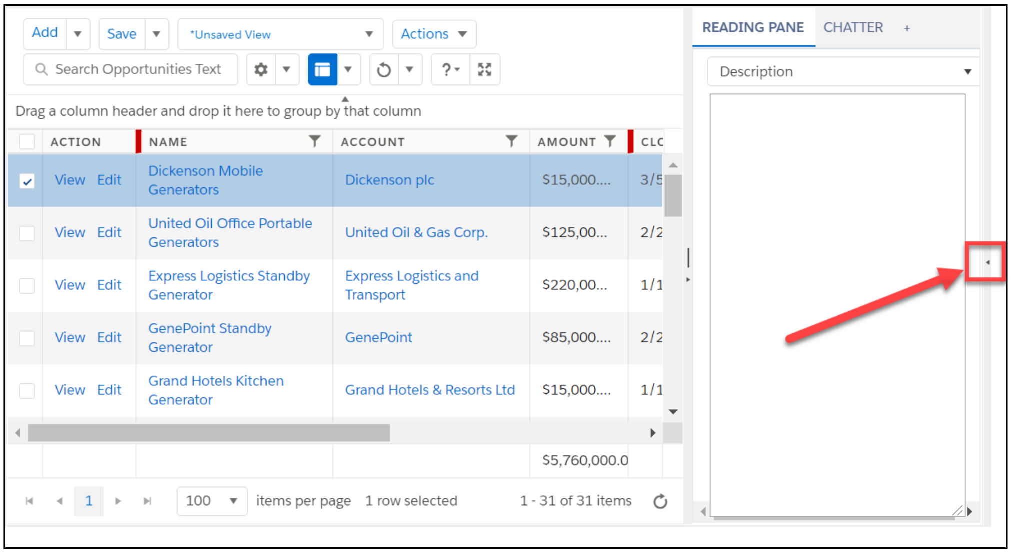
g. Click Save.

3. Configure Lightning Components and Lightning Web Components for your Conga Grid.
   a. Navigate to a specific Conga Grid.
   b. Click the Select Reading Pane Options to display the Reading Pane.


Account
Edge Communications
Add
Save
*Unsaved View
Actions
Search Opportunities Text
?
Drag a column header and drop it here to group by that column

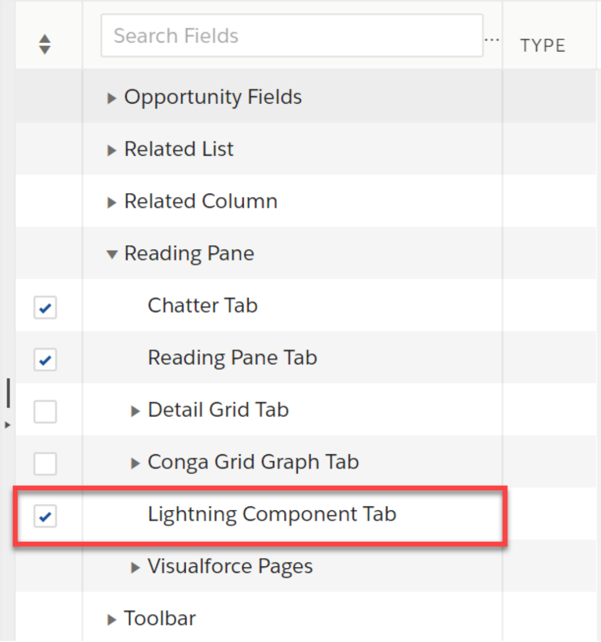
ACTION	NAME	ACCOUNT	AMOUNT	CLOSE ...	STAGE
View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....	3/5/2020	Qualification
View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,00...	2/22/2020	Negotiation/Review
View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,00...	1/18/2020	Closed Won
View Edit	GenePoint Standby Generator	GenePoint	$85,000....	2/25/2020	Closed Won
View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....	1/13/2020	Id. Decision Makers
View Edit	United Oil Refinery Generators	United Oil & Gas Corp.	$270,00...	4/7/2020	Proposal/Price Quote
			$5,760,000.0		

c. Click the expandable arrow to access Reading Pane Configuration options.


Add
Save
*Unsaved View
Actions
Search Opportunities Text
?
Drag a column header and drop it here to group by that column
READING PANE
CHATTER
+
Description

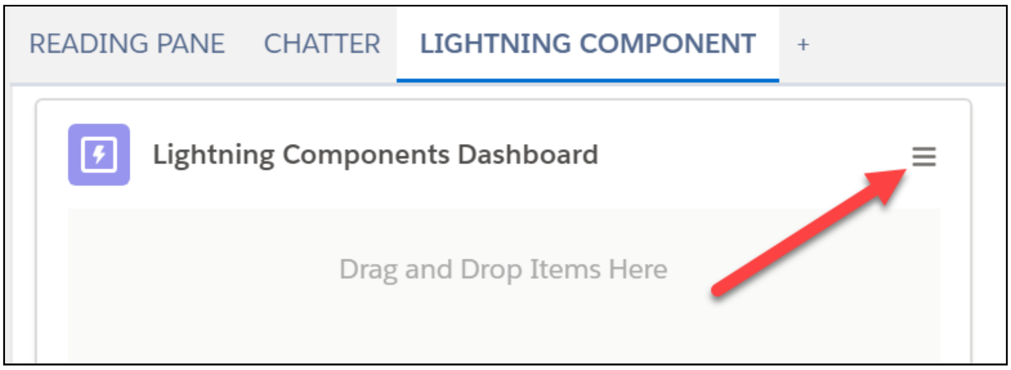
ACTION	NAME	ACCOUNT	AMOUNT	CLC
☑ View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....	3/5
View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,00...	2/2
View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,00...	1/1
View Edit	GenePoint Standby Generator	GenePoint	$85,000....	2/2
View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....	1/1
			$5,760,000.0	

d. Expand the Reading Pane menu and select Lightning Component Tab to add the Lightning Component tab to the Reading Pane.

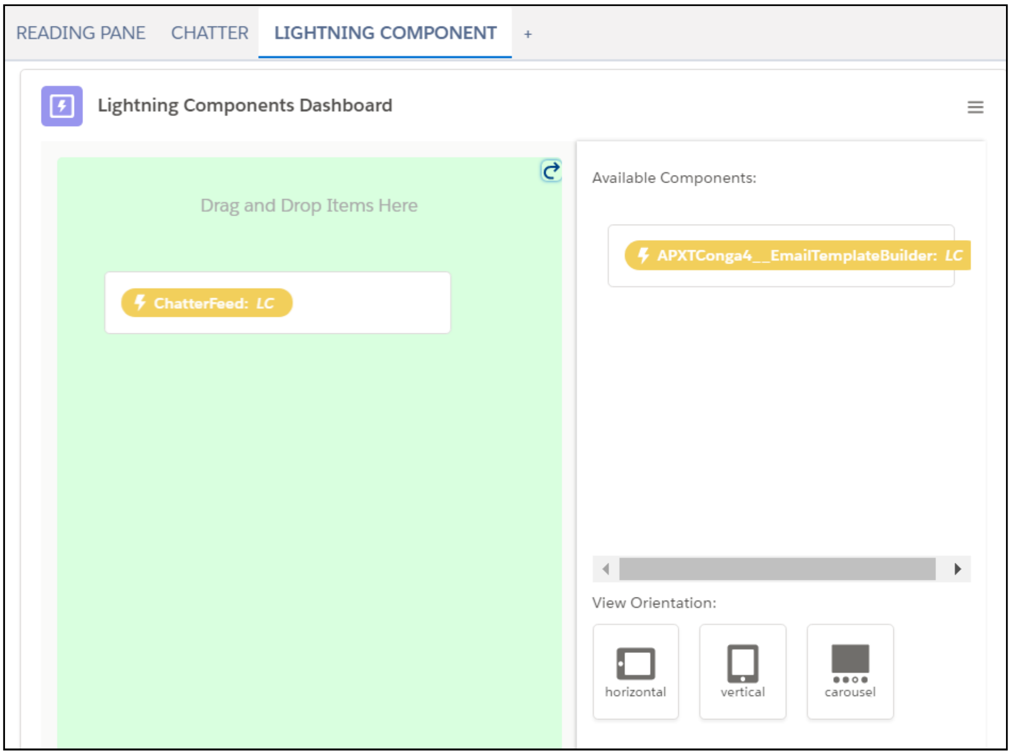


4. Add and customize your Components in the Reading Pane.

   a. Select the newly added Lightning Component tab and click the Settings menu icon.



b. Drag and drop the Component from the Available Components section to the Lightning Components Dashboard section.



c. Choose a View Orientation to determine how the Component displays in the Reading Pane.

- Vertical
- Carousel

5. In the Conga Grid View, click Save, or one of options in the Save dropdown menu, to save the View.

Using Composer and Orchestrate Lightning Components in Conga Grid

Use Conga Composer and Conga Orchestrate Lightning Components in Conga Grid's Reading Pane.

Important

Before You Begin: Enable the Lightning Component Feature in Conga Grid Setup. For specific instructions, see Setting Up Lightning Components and Lightning Web Components in Conga Grid.

Conga Composer Lightning Component

You must have preconfigured Composer Solutions and Solution Collections to use this Lightning Component in Conga Grid.

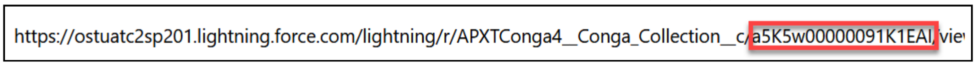
Use the **Composer Lightning Component** (*Conga Composer > Composer Lightning Component*) in Conga Grid to merge documents and send emails for multiple records at once without leaving the grid.

1. Enable the Composer Lightning Component.
    a. Navigate to the Conga Grid Setup tab.
    b. Select Global Defaults or a specific object to enable the Composer Lightning Component on.
    c. Locate the APXTConga4__ComposerMerge Lightning Component and click the radio button next to it.
    d. Change the Component Enabled toggle from Disabled to Enabled.
2. Enable the Edit as JSON feature.
    a. Change the Edit as JSON toggle from Disabled to Enabled.
    b. Copy and paste the JSON code below into the Component's Attributes in JSON box.

    ```
    {
      "selectedCollectionId" : "[insert Solution Collection Id here]"
    }
    ```

    > **Important**
    >
    > The Solution Collection (*Conga Composer > Conga Solution Collections*)record (referenced in the JSON code) must be configured for the same object as Conga Grid. For example, a Solution Collection configured for the standard Contract object cannot be used on an Account grid.

    - To locate a Solution Collection Id:
        1. Navigate to Conga Grid Explorer.
        2. Search for Conga Collection and select Conga Collection.
        3. Click View next to a Solution Collection record.
        4. Locate the record Id in the Salesforce URL and copy it.

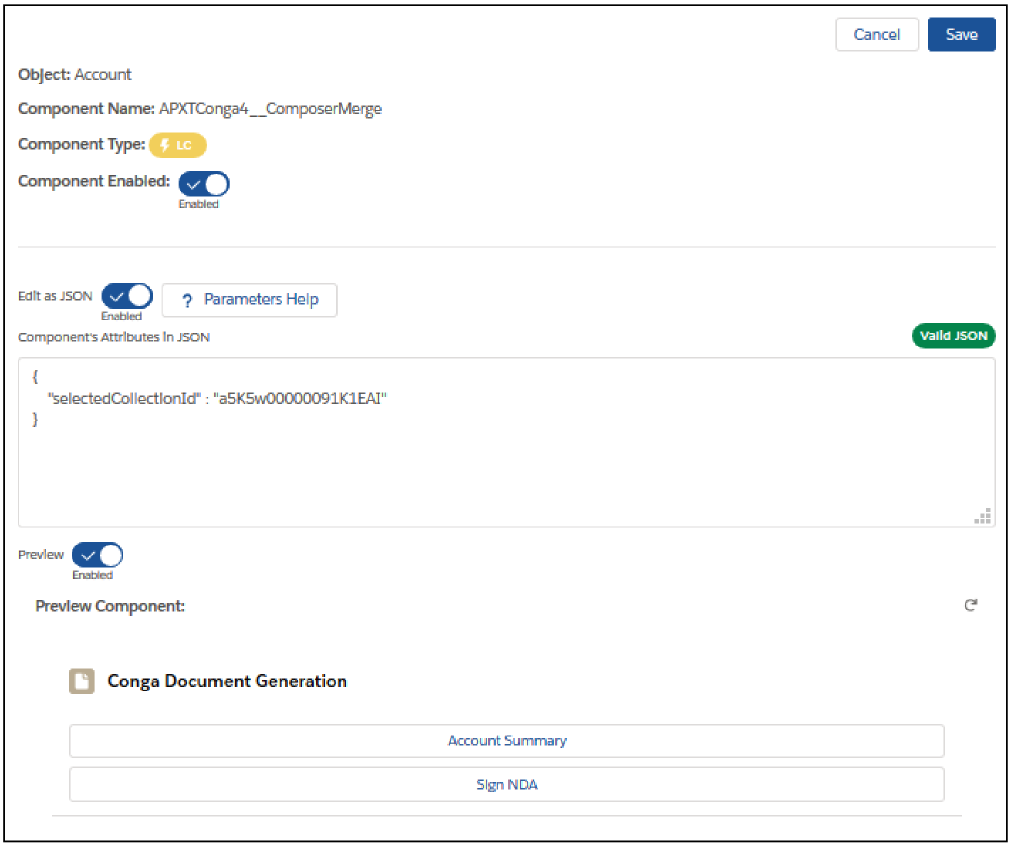


https://ostuatc2sp201.lightning.force.com/lightning/r/APXTConga4__Conga_Collection__c/a5K5w00000091K1EAI/vie

    c. Remove the brackets and text in between the brackets and replace it with a specific Solution Collection Record Id.
        - Example:

    ```
    {
      "selectedCollectionId" : "a5K5w00000091K1EAI"
    }
    ```

d. (Optional) Change the Preview toggle from Disabled to Enable to preview the Composer Lightning Component.



Object: Account

Component Name: APXTConga4__ComposerMerge

Component Type: LC

Component Enabled: Enabled

Edit as JSON Enabled

Component's Attributes in JSON

```
{
  "selectedCollectionId" : "a5K5w00000091K1EAI"
}
```

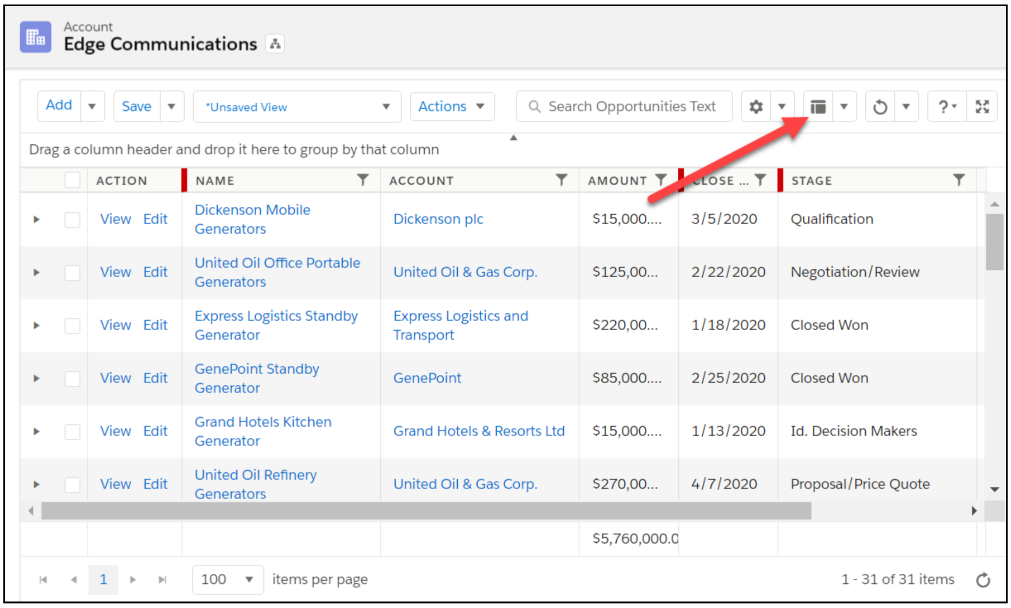
Preview Enabled

Preview Component:

Conga Document Generation

e. Click Save.

3. Add the Composer Lightning Component to a Conga Grid View.
   a. Navigate to a specific Conga Grid and select a View.
   b. Click the Select Reading Pane Options to display the Reading Pane

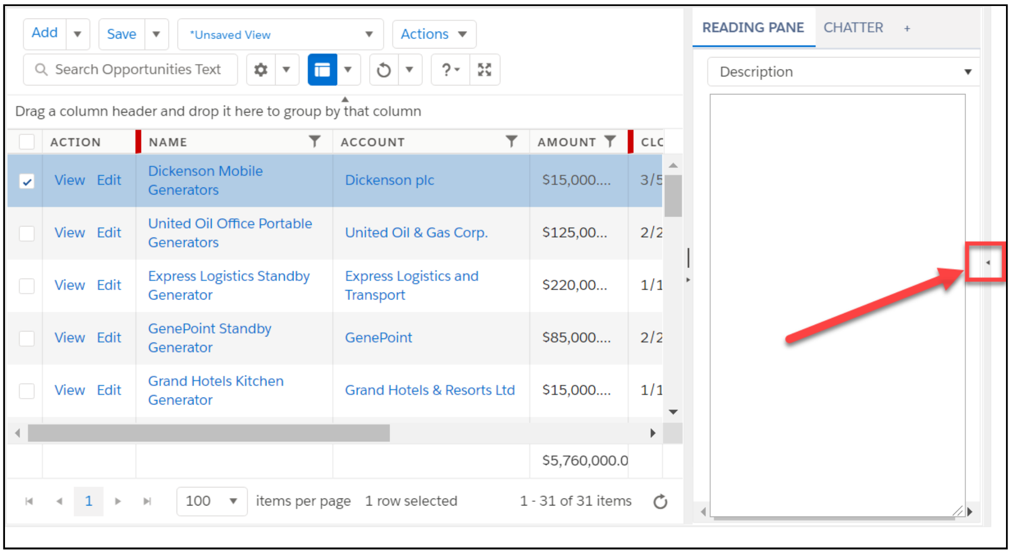

Add
Save
*Unsaved View
Actions
Search Opportunities Text
Drag a column header and drop it here to group by that column

Account
Edge Communications

ACTION	NAME	ACCOUNT	AMOUNT	CLOSE ...	STAGE
View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....	3/5/2020	Qualification
View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,00...	2/22/2020	Negotiation/Review
View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,00...	1/18/2020	Closed Won
View Edit	GenePoint Standby Generator	GenePoint	$85,000....	2/25/2020	Closed Won
View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....	1/13/2020	Id. Decision Makers
View Edit	United Oil Refinery Generators	United Oil & Gas Corp.	$270,00...	4/7/2020	Proposal/Price Quote
			$5,760,000.0		

1 | 100 items per page | 1 - 31 of 31 items

c. Click the expandable arrow to access Reading Pane Configuration options.

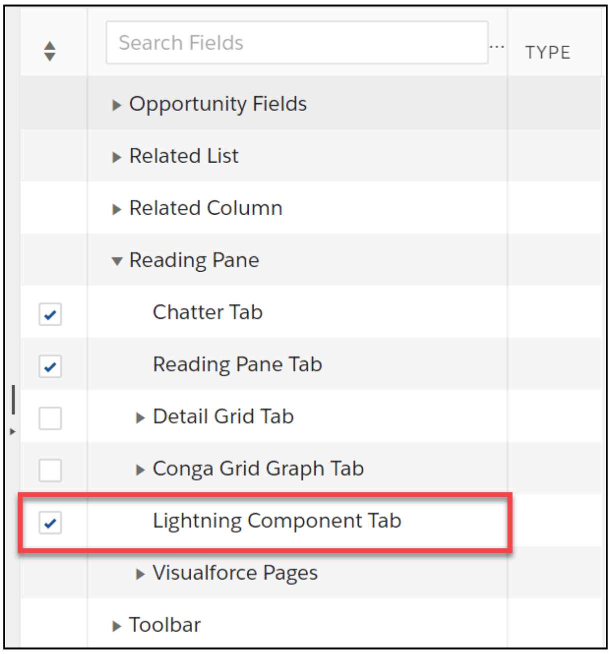


Drag a column header and drop it here to group by that column

	ACTION	NAME	ACCOUNT	AMOUNT	CLC
☑	View Edit	Dickenson Mobile Generators	Dickenson plc	$15,000....	3/5
□	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$125,00...	2/2
□	View Edit	Express Logistics Standby Generator	Express Logistics and Transport	$220,00...	1/1
□	View Edit	GenePoint Standby Generator	GenePoint	$85,000....	2/2
□	View Edit	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	$15,000....	1/1
				$5,760,000.0	

1 | 100 items per page | 1 row selected | 1 - 31 of 31 items

d. Expand the Reading Pane menu and select Lightning Component Tab to add the Lightning Component tab to the Reading Pane.



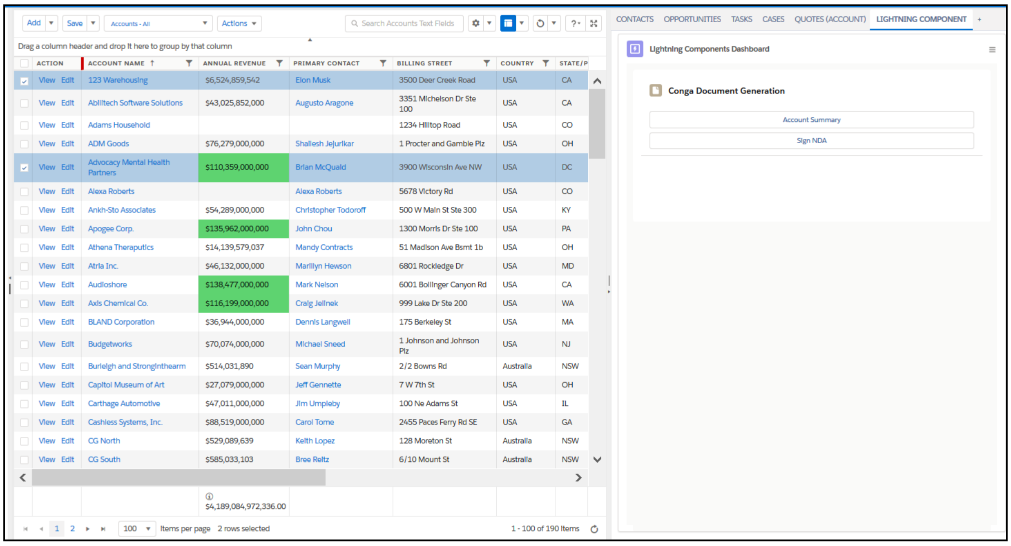
- Opportunity Fields
- Related List
- Related Column
- Reading Pane
  - ☑ Chatter Tab
  - ☑ Reading Pane Tab
  - □ Detail Grid Tab
  - □ Conga Grid Graph Tab
  - ☑ Lightning Component Tab
  - Visualforce Pages
- Toolbar

e. Drag and drop the APXTCONGA4__ComposerMerge: LC Lightning Component from the Available Components section to the Lightning Components Dashboard section.

f. (Optional) Choose a View Orientation to determine how the Component displays in the Reading Pane.

g. In the Conga Grid View, click Save, or one of the options in the Save dropdown menu, to save the View.

- Example of Composer Lightning Component in Conga Grid



Drag a column header and drop it here to group by that column

□	ACTION	ACCOUNT NAME ↑	ANNUAL REVENUE	PRIMARY CONTACT	BILLING STREET	COUNTRY	STATE/P
☑	View Edit	123 Warehousing	$6,524,859,542	Elon Musk	3500 Deer Creek Road	USA	CA
□	View Edit	Abilitech Software Solutions	$43,025,852,000	Augusto Aragone	3351 Michelson Dr Ste 100	USA	CA
□	View Edit	Adams Household			1234 Hilltop Road	USA	CO
□	View Edit	ADM Goods	$76,279,000,000	Shailesh Jejurikar	1 Procter and Gamble Plz	USA	OH
☑	View Edit	Advocacy Mental Health Partners	$110,359,000,000	Brian McQuaid	3900 Wisconsin Ave NW	USA	DC
□	View Edit	Alexa Roberts		Alexa Roberts	5678 Victory Rd	USA	CO
□	View Edit	Ankh-Sto Associates	$54,289,000,000	Christopher Todoroff	500 W Main St Ste 300	USA	KY
□	View Edit	Apogee Corp.	$135,962,000,000	John Chou	1300 Morris Dr Ste 100	USA	PA
□	View Edit	Athena Theraputics	$14,139,579,037	Mandy Contracts	51 Madison Ave Bsmt 1b	USA	OH
□	View Edit	Atria Inc.	$46,132,000,000	Marillyn Hewson	6801 Rockledge Dr	USA	MD
□	View Edit	Audioshore	$138,477,000,000	Mark Nelson	6001 Bollinger Canyon Rd	USA	CA
□	View Edit	Axis Chemical Co.	$116,199,000,000	Craig Jelinek	999 Lake Dr Ste 200	USA	WA
□	View Edit	BLAND Corporation	$36,944,000,000	Dennis Langwell	175 Berkeley St	USA	MA
□	View Edit	Budgetworks	$70,074,000,000	Michael Sneed	1 Johnson and Johnson Plz	USA	NJ
□	View Edit	Burleigh and Stronginthearm	$514,031,890	Sean Murphy	2/2 Bowns Rd	Australia	NSW
□	View Edit	Capitol Museum of Art	$27,079,000,000	Jeff Gennette	7 W 7th St	USA	OH
□	View Edit	Carthage Automotive	$47,011,000,000	Jim Umpleby	100 Ne Adams St	USA	IL
□	View Edit	Cashless Systems, Inc.	$88,519,000,000	Carol Tome	2455 Paces Ferry Rd SE	USA	GA
□	View Edit	CG North	$529,089,639	Keith Lopez	128 Moreton St	Australia	NSW
□	View Edit	CG South	$585,033,103	Bree Reitz	6/10 Mount St	Australia	NSW
			ⓘ $4,189,084,972,336.00				

1 2 | 100 Items per page | 2 rows selected | 1 - 100 of 190 Items

Conga Orchestrate Checklist Lightning Web Component

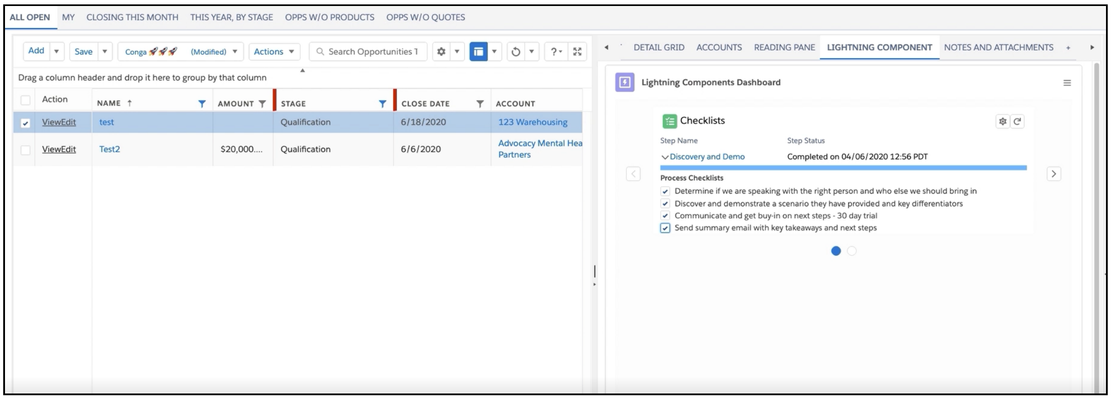
> You must have preconfigured Checklist Steps in your Process Definition to use this Lightning Web Component in Conga Grid.

Use the Conga Orchestrate Checklist Lightning Web Component (*Conga Orchestrate > Process Configuration > Process Steps > Managing the Status of a Step With a Checklist*) in Conga Grid to manage the status of processes for multiple records all on the same page.

1. Enable the Orchestrate Checklist Lightning Web Component.
   a. Navigate to the Conga Grid Setup tab.
   b. Select Global Defaults or a specific object to enable the Composer Lightning Component on.
   c. Locate the FSTR__checklistsComponent Lightning Component and click the radio button next to it.
   d. Change the Component Enabled toggle from Disabled to Enabled.
   e. Click Save.
2. Add the Orchestrate Checklist Lightning Web Component to a Conga Grid View.
   a. Navigate to a specific Conga Grid and select a View.
   b. Click the Select Reading Pane Options to display the Reading Pane.
   c. Click the expandable arrow to access Reading Pane Configuration options.
   d. Expand the Reading Pane menu and select Lightning Component Tab to add the Lightning Component tab to the Reading Pane.
   e. Drag and drop the FSTR__checklistsComponent: LWC Lightning Web Component from the Available Components section to the Lightning Components Dashboard section.

  f. (Optional) Choose a View Orientation to determine how the Component displays in the Reading Pane.
  g. In the Conga Grid View, click Save, or one of the options in the Save dropdown menu, to save the View.

- Example of Orchestrate Checklist Lightning Web Component in Conga Grid:



Drag a column header and drop it here to group by that column

☐	Action	NAME ↑	AMOUNT	STAGE	CLOSE DATE	ACCOUNT
☑	ViewEdit	test		Qualification	6/18/2020	123 Warehousing
☐	ViewEdit	Test2	$20,000....	Qualification	6/6/2020	Advocacy Mental Hea Partners

Checklists

Step Name	Step Status
Discovery and Demo	Completed on 04/06/2020 12:56 PDT

Process Checklists

☑ Determine if we are speaking with the right person and who else we should bring in
☑ Discover and demonstrate a scenario they have provided and key differentiators
☑ Communicate and get buy-in on next steps - 30 day trial
☑ Send summary email with key takeaways and next steps

Conga Orchestrate Approvals Lightning Component

> ⚠ You must have preconfigured Approvals Steps in your Process Definition to use this Lightning Component in Conga Grid.

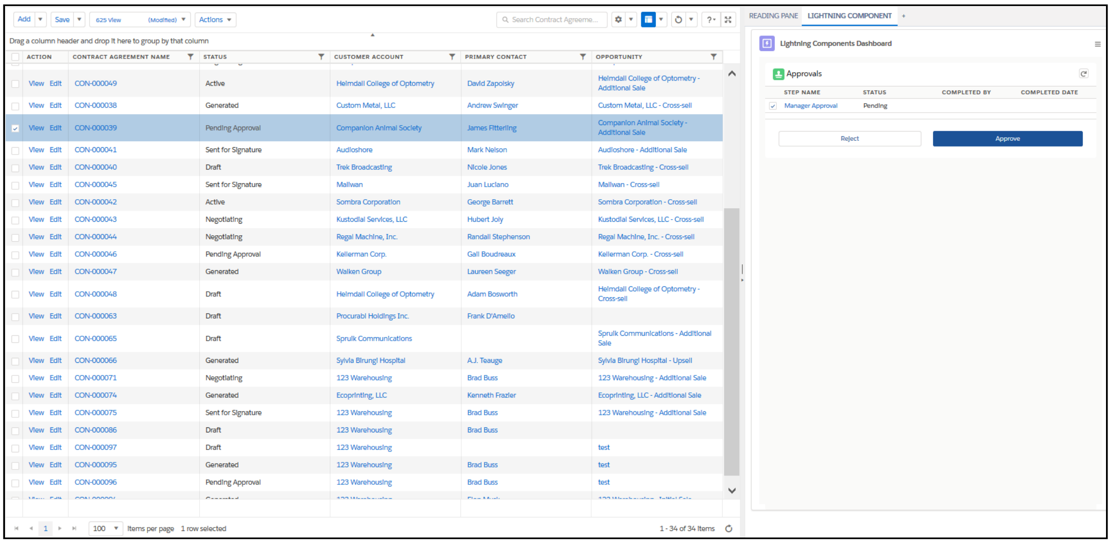
Use the Conga Orchestrate Approvals Lightning Component (*Conga Orchestrate > Installation and Setup > Advanced Setup > Use a Lighting Component to Manage Approvals for a Process Object*) in Conga Grid to manage multiple Process Object records' Approvals all on the same page.

> ⚠ The FSTR__ApprovalList Lightning Component is only available for use on Process Objects. Use the FSTR__ApprovalItem Lightning Component to use the Approvals Lightning Component on Step Objects.

1. Enable the Orchestrate Approvals Web Component.
    a. Navigate to the Conga Grid Setup tab.
    b. Select Global Defaults or a specific object to enable the Composer Lightning Component on.
    c. Locate the FSTR__ApprovalList Lightning Component and click the radio button next to it.
    d. Change the Component Enabled toggle from Disabled to Enabled.
    e. Click Save.
2. Add the Orchestrate Approvals Lightning Component to a Conga Grid View.
    a. Navigate to a specific Conga Grid and select a View.

b. Click the Select Reading Pane Options to display the Reading Pane.
c. Click the expandable arrow to access Reading Pane Configuration options.
d. Expand the Reading Pane menu and select Lightning Component Tab to add the Lightning Component tab to the Reading Pane.
e. Drag and drop the FSTR__ApprovalList: LC Lightning Component from the Available Components section to the Lightning Components Dashboard section.
f. (Optional) Choose a View Orientation to determine how the Component displays in the Reading Pane.
g. In the Conga Grid View, click Save, or one of the options in the Save dropdown menu, to save the View.

- Example of Conga Orchestrate Approvals Lightning Component in Conga Grid:



Drag a column header and drop it here to group by that column

☐	ACTION	CONTRACT AGREEMENT NAME	STATUS	CUSTOMER ACCOUNT	PRIMARY CONTACT	OPPORTUNITY
☐	View Edit	CON-000049	Active	Heimdall College of Optometry	David Zapolsky	Heimdall College of Optometry - Additional Sale
☐	View Edit	CON-000038	Generated	Custom Metal, LLC	Andrew Swinger	Custom Metal, LLC - Cross-sell
☑	View Edit	CON-000039	Pending Approval	Companion Animal Society	James Fitterling	Companion Animal Society - Additional Sale
☐	View Edit	CON-000041	Sent for Signature	Audioshore	Mark Nelson	Audioshore - Additional Sale
☐	View Edit	CON-000040	Draft	Trek Broadcasting	Nicole Jones	Trek Broadcasting - Cross-sell
☐	View Edit	CON-000045	Sent for Signature	Mailwan	Juan Luciano	Mailwan - Cross-sell
☐	View Edit	CON-000042	Active	Sombra Corporation	George Barrett	Sombra Corporation - Cross-sell
☐	View Edit	CON-000043	Negotiating	Kustodial Services, LLC	Hubert Joly	Kustodial Services, LLC - Cross-sell
☐	View Edit	CON-000044	Negotiating	Regal Machine, Inc.	Randall Stephenson	Regal Machine, Inc. - Cross-sell
☐	View Edit	CON-000046	Pending Approval	Kellerman Corp.	Gail Boudreaux	Kellerman Corp. - Cross-sell
☐	View Edit	CON-000047	Generated	Walken Group	Laureen Seeger	Walken Group - Cross-sell
☐	View Edit	CON-000048	Draft	Heimdall College of Optometry	Adam Bosworth	Heimdall College of Optometry - Cross-sell
☐	View Edit	CON-000063	Draft	Procurabl Holdings Inc.	Frank D'Amelio	
☐	View Edit	CON-000065	Draft	Spruik Communications		Spruik Communications - Additional Sale
☐	View Edit	CON-000066	Generated	Sylvia Birungi Hospital	A.J. Teauge	Sylvia Birungi Hospital - Upsell
☐	View Edit	CON-000071	Negotiating	123 Warehousing	Brad Buss	123 Warehousing - Additional Sale
☐	View Edit	CON-000074	Generated	Ecoprinting, LLC	Kenneth Frazier	Ecoprinting, LLC - Additional Sale
☐	View Edit	CON-000075	Sent for Signature	123 Warehousing	Brad Buss	123 Warehousing - Additional Sale
☐	View Edit	CON-000086	Draft	123 Warehousing	Brad Buss	
☐	View Edit	CON-000097	Draft	123 Warehousing		test
☐	View Edit	CON-000095	Generated	123 Warehousing	Brad Buss	test
☐	View Edit	CON-000096	Pending Approval	123 Warehousing	Brad Buss	test

1 100 Items per page 1 row selected 1 - 34 of 34 Items

	STEP NAME	STATUS	COMPLETED BY	COMPLETED DATE
☑	Manager Approval	Pending		

Add Formula Fields

The following steps illustrate an example of experimenting with a new pricing model. This example creates a new field called 5% price increase that adds five percent to the Amount field of an Opportunity object.

1. In Conga Grid℠ Explorer, select an object from the list. For this example, click **Opportunity**.
2. Click the drop-down arrow next to **Configure** and click **Formulas**.
3. In the Saved Formulas window, click **New**.
4. Enter a **Formula Label**, which is the name of the new field header.
5. Click **Format As** and select an option:
    - No Formatting
    - Currency
    - Date

- DateTime
- Decimal
- Number
- Time

6. Click **Functions** and select a mathematical function or logical operator to apply. The 5% price increase example uses **Number.**
7. **Multiply**. (See Formula Functions for definitions of each option.) Click the ellipses to enter or select a value.
8. Click **Fields** and select a field on which to apply the formula function.
9. Click the ellipses area and enter a value.
10. Click **Apply**. The new field appears at the end of the grid. You can drag the formula field next to the original field for comparison.

Formula Functions

The following options are available from the Functions drop-down list in the Saved Formulas window:

Date

Function	Javascript Notation	Definition
Add X Days	(new Date(...)).addDays(...)	Add a specified number of days to a date/time field.
Add X Hours	(new Date(...)).addHours(...)	Add a specified number of hours to a date/time field.
Add X Minutes	(new Date(...)).addMinutes(...)	Add a specified number of minutes to a date/time field.
Add X Months	(new Date(...)).addMonths(...)	Add a specified number of months to a date/time field.
Add X Weeks	(new Date(...)).addWeeks(...)	Add a specified number of weeks to a date/time field.
Add X Years	(new Date(...)).addYears(...)	Add a specified number of years to a date/time field.

Function	Javascript Notation	Definition
Day of Month (1-31)	... .getDate()	Returns the day of month for the specified date.
Day of Week (0-6)	... .getDay()	Returns the zero-based day of the week (0=Sunday, 6=Saturday).
Hour of Day (0-23)	... .getHours()	Returns the zero-based hour for the specified date.
Is After	... .isAfter(...)	Returns true if the first date occurs after the second date. If a second date is not specified, Now is used.
Is Before	... .isBefore(...)	Returns true if the first date occurs before the second date. If a second date is not specified, Now is used.
Is Between	... .between(...,...)	Returns true if the specified date is between or equal to the specified start and end dates.
Is Blank	... == null	Checks if a reference is to an empty grid cell. It returns true or false.
Is Equal To	... === ...	Checks if a numeric argument is equal in value to a second numeric argument. If so, it returns true.
Is In Next X	....inNext(...numberOf...,"...DaysWeeksMonthsOrYears...")	Returns true if the date occurs within the next specified number of days, weeks, months, or years of the current date.
Is In Previous X	....inLast(...numberOf...,"...DaysWeeksMonthsOrYears...")	Returns true if the date occurs within the prior specified number of days, weeks, months, or years of the current date.
Is Not Blank	... != null	Checks if a reference is to a populated grid cell. It returns true or false.

Function	Javascript Notation	Definition
Is Not Equal To	... !== ...	Checks if a numeric argument is not equal in value to a second numeric argument. If so, it returns true.
Is Today	(...).between(Date.today(), Date.parse("tomorrow"))	Returns true if date value is todays date.
Is Weekday	... .isWeekday()	Returns true if the date is a weekday (Monday-Friday).
Minutes (0-59)	... .getMinutes()	Returns the zero-based number of minutes past the hour, for the specified date.
Month (0-11)	... .getMonth()	Returns the zero-based month (0=January, 11=December) of the specified date.
Now	(new Date())	Returns the current date and time.
Today	(Date.today())	Returns the current date.
Value	(new Date(...year...,...month..., ... day...,...hour...,...minutes...))	Creates a new date with the specified year, month, day, hours and minutes.
Year	... .getFullYear()	Returns the year of the specified date.

For additional information see: https://github.com/datejs/Datejs

Format

Function	Javascript Notation	Definition
As Currency	kendo.format("{0:c}", ...)	Formats a numeric value as currency with a dollar sign, for example: $1234.56.
As Date	kendo.format("{0:d}", ...)	Formats a numeric value as a date in the form m/d/yyyy.

Function	Javascript Notation	Definition
As DateTime	kendo.format("{0:g}", ...)	Formats a numeric value as a date and time in the form M/d/yyyy h:mm tt.
As Decimal	kendo.format("{0:n}", ...)	Formats a numeric value as a floating-point number, for example: 1234.56.
As Number	kendo.format("{0:n0}", ...)	Formats a numeric value as an integer, for example: 1234.
As Time	kendo.format("{0:hh:mm tt}", ...)	Formats a numeric value as time in the form hh:mm tt.

Logic

Function	Javascript Notation	Definition
And	... && ...	Checks whether all arguments are true. If so, it returns true.
Group Selected	(...)	Logically groups a statement for order of operations.
If	(...if...) ? (...then...) : (...else...)	Checks whether a condition is met. It returns one value if true and another value if false.
Is Blank	... == null	Checks if a reference is to an empty grid cell. It returns true or false.
Is Equal To	... === ...	Checks if a numeric argument is equal in value to a second numeric argument. If so, it returns true.
Is Greater Than	... > ...	Checks if a numeric argument is greater than a second numeric argument. If so, it returns true.
Is Less Than	... < ...	Checks if a numeric argument is less than a second numeric argument. If so, it returns true.

Function	Javascript Notation	Definition
Is Not Blank	... != null	Checks if a reference is to a populated grid cell. It returns true or false.
Is Not Equal To	... !== ...	Checks if a numeric argument is not equal in value to a second numeric argument. If so, it returns true.
Not	!(...)	Checks whether a condition is not met. If so, it returns true.
Or	... || ...	Checks whether any of the arguments are true. If so, it returns true. If all arguments are false, it returns false.

Number

Function	Javascript Notation	Definition
Multiply	... * ...	Multiplies two numeric values.
Add	... + ...	Adds two numeric values.
Subtract	... - ...	Subtracts two numeric values.
Divide	... / ...	Divides two numeric values.
Absolute Value	Math.abs(...)	Returns the absolute value of a number (the number without its sign).
Is Blank	... == null	Checks if a reference is to an empty grid cell. It returns true or false.
Is Equal To	... === ...	Checks if a numeric argument is equal in value to a second numeric argument. If so, it returns true.
Is Greater Than	... > ...	Checks if a numeric argument is greater than a second numeric argument. If so, it returns true.

Function	Javascript Notation	Definition
Is Less Than	... < ...	Checks if a numeric argument is less than a second numeric argument. If so, it returns true.
Is Not Blank	... != null	Checks if a reference is to a grid cell that is populated. It returns true or false.
Is Not Equal To	... !== ...	Checks if a numeric argument is not equal in value to a second numeric argument. If so, it returns true.
Max	Math.max(...1...,...2...,...)	Returns the maximum value in a set of values.
Min	Math.min(...1...,...2...,...)	Returns the minimum value in a set of values.
Random (0-1)	Math.random()	Returns a random number greater than or equal to 0 and less than 1, evenly distributed.
Round	Math.round(...)	Rounds a number to a specified number of digits.
Round Down	Math.floor(...)	Rounds a number down, to the nearest integer or to the nearest multiple of significance.
Round Up	Math.ceil(...)	Rounds a number up, to the nearest integer or to the nearest multiple of significance.
Exponent	Math.pow(...base...,...exp...)	Returns the result of a number raised to a power.
Square Root	Math.sqrt(...)	Returns the square root of a number.

Text

Function	Javascript Notation	Definition
Append	... + ...	Appends two text strings.
Contains	(... .indexOf(...) > 0)	Returns true if a string contains a specified value.
Index Of	...indexOf(...substring...)	Returns the index if a string appears in the value, or -1 if the value does not contain the string.
Is Blank	... == null	Checks if a reference is to an empty grid cell. It returns true or false.
Is Equal To	... === ...	Checks if a text string is equal in value to a second text string. If so, it returns true.
Is Not Blank	... != null	Checks if a reference is to a grid cell that is populated. It returns true or false.
Is Not Equal To	... !== ...	Checks if a text string is not equal in value to a second text string. If so, it returns true.
Replace	... .replace(...find...,...replace...)	Replaces part of a text string with a different text string.
Substring	... .substr(...startIndex...,... length...)	Returns a substring of the value starting at a specified index and with a specified length.
Value	"..."	A specified text value.

Advanced

Function	Javascript Notation
For Each	Object.each(...list..., function(item){ // Use item });
Function (required)	function(){ // comments return ...; }()

If	if (...test...) { ...then... } else { ...else... }
Regular expression	/...regex.../.test(...string...)
Variable	var ...x... = ...y...;

Date Formula Fields

We provided several formulas below that you can copy and paste into the JavaScript field of the Formula Fields window. The following example shows the Work Day Difference formula for determining the number of working days since the dates in the Created Date field:

Result:

The following are some examples of different formulas that you can use. These examples assume a typical work day of Monday through Friday, 9:00 a.m. to 5:00 p.m.

- Change the work days as needed using the *workDays* array.
- Change the working hours as needed using the *startHour* and *endHour* variables.
- Refer to the kendo number formatting and date formatting documentation, for details on the *kendo.format* specification.

Work minutes difference

This formula returns the number of work minutes between two specified dates.

```
(function(){

    var startHour = 9;
    var endHour = 17;
    var workDays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];
    var minutes = 0;
    var dateField = new Date(Date1__c);  // This is the first date field in Salesforce
    var todaysDate = new Date(Date2__c);  // This is the second date field in Salesforce

     // To specify today instead of Date2, use new Date(). For example:
     // var todaysDate = new Date();

    while(true) {
      //compare Date1 to today
      if(dateField.compareTo(todaysDate) < 0) {

        // (...if...) ? (...then...) : (...else...)
        // If the date is within the workDays array and within work hours then add 1, else 0

        minutes += (workDays.indexOf(dateField.getDayName()) > -1
              && dateField.getHours() >= startHour
              && dateField.getHours() < endHour
              ? 1 : 0);

      dateField.addMinutes(1); //increment the counter until it reaches today's date and time
      }

      else{
        break;
      }
    }
    return minutes;
  }())
```

Work hour difference

This formula returns the number of work hours between two specified dates.

```
kendo.format("{0:n}",

(function(){

    var startHour = 9;
    var endHour = 17;
    var workDays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];
    var minutes = 0;
    var dateField = new Date(Date1__c);  // This is the first date field in Salesforce
    var todaysDate = new Date(Date2__c);  // This is the second date field in Salesforce

     // To specify today instead of Date2, use new Date(). For example:
     // var todaysDate = new Date();
    while(true) {
      //compare Date1 with today
      if(dateField.compareTo(todaysDate) < 0) {
        // (...if...) ? (...then...) : (...else...)
        // If the date is within the workDays array and within work hours then add 1, else 0

        minutes += (workDays.indexOf(dateField.getDayName()) > -1
              && dateField.getHours() >= startHour
              && dateField.getHours() < endHour
              ? 1 : 0);

      dateField.addMinutes(1); //increment the counter till we hit today's date/time
      }

      else{
        break;
      }
    }
    return minutes / 60 ;
  }())
)
```

Work day difference

This formula returns the number of work days between two specified dates.

```
kendo.format("{0:n}",

(function(){

    var startHour = 9;
    var endHour = 17;
    var workDays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];
    var minutes = 0;
    var dateField = new Date(Date1__c);  // This is the first date field in Salesforce
    var todaysDate = new Date(Date2__c); // This is the second date field in Salesforce

     // To specify today instead of Date2, use new Date(). For example:
     // var todaysDate = new Date();

    while(true) {
      //compare Date1 with today
      if(dateField.compareTo(todaysDate) < 0) {

        // (...if...) ? (...then...) : (...else...)
        // If the date is within the workDays array and within work hours then add 1, else 0

        minutes += (workDays.indexOf(dateField.getDayName()) > -1
              && dateField.getHours() >= startHour
              && dateField.getHours() < endHour
```

```
                    ? 1 : 0);

            dateField.addMinutes(1); //increment the counter till we hit today's date/time
            }

            else{
               break;
            }
        }
        return minutes / 60 / (endHour - startHour);
    }())
)
```

+1 day from Day 1

This formula adds one day to a specified date.

```
kendo.format("{0:d}", new Date(Date1__c).addDays(1))
```

-1 day from Day 1

This formula subtracts one day from a specified date.

```
kendo.format("{0:d}", new Date(Date1__c).addDays(-1))
```

+1 hour from Day 1

This formula adds one hour to a specified date.

```
kendo.format("{0:g}", new Date(Date1__c).addHours(1))
```

+1 month from Day 1

This formula adds one month to a specified date.

```
kendo.format("{0:d}", new Date(Date1__c).addMonths(1))
```

Hours difference

This formula computes the number of hours between two specified dates.

```
kendo.format("{0:n}", ((Date1__c - Date2__c) / (1000*60*60)) )
```

Days difference

This formula computes the number of days between two specified dates.

```
kendo.format("{0:n}", ((Date1__c - Date2__c) / (1000*60*60*24)))
```

Days difference UTC

This formula returns the number of days between two specified dates. Use this function to compare dates where one is in a date/time format (such as today's date) and the other is just a date field (with no time). Unless you want the decimal component of the partial day, you need to put both dates in the 12:00:00 a.m. format. The formula does this by subtracting the hours component, if it exists.

```
kendo.format("{0:n}",
(
function () {

   var startDate = new Date(Date2__c);
   var endDate = new Date(Date3__c);
   var millisecondsPerDay = 24 * 60 * 60 * 1000;

   // Make sure that both dates are at midnight. Salesforce date SF fields have a Date Offset built in.
   // So we have to subtract from midnight and then move the date up to midnight of the next day.

   if(startDate.getHours() > 0) {
    //if the time is already @ midnight then don't adjust
    startDate = startDate.addHours( -1 * startDate.getHours())
    startDate = new Date(startDate.getFullYear(),startDate.getMonth(), startDate.getDay());
   }

   if(endDate.getHours() > 0) {
     //if the time is already @ midnight then don't adjust
    endDate = endDate.addHours( (-1 *  endDate.getHours()));
   }

   return ((endDate  - startDate ) / millisecondsPerDay);
}()
)
)
```

Opportunity Product Details

Suppose that you are quoting a product with the following values:

If you change the Quantity to 2, Salesforce® will adjust the List Price down to $5.00 so that the Total Price stays at $10. (See the Salesforce API documentation for details.) This may not be intuitive for most users. If you define Quantity and List Price in Conga Grid℠, the Total Price updates accordingly. In this example, increasing the quantity to 2 will update the Total Price to $20.

If you prefer the default Salesforce behavior, then only define the Quantity. The List Price will adjust down in order to keep the current Total Price.

Conga Grid and Cirrus Insight

The Cirrus Insight app brings Salesforce into your inbox. Conga Grid conversely brings your inbox into Salesforce. With both applications installed, you can do things like:

- Use Conga Grid as a dashboard to show follow-up or overdue tasks, then view emails related to the associated records.
- View emails associated with open activities or opportunities.
- Click a Conga Grid tab within an email message to view all records associated with that contact.

The first example shows a custom tab that displays a Conga Grid of open tasks. With conditional formatting, you can quickly see which tasks are more than three days overdue (colored red), less than two days overdue (colored yellow), and due today (green). In this example, clicking one of the records displays an email message (in the Reading Pane) related to an upcoming task. For this to work properly, you must have already imported email conversations into Salesforce.

The next example shows a custom tab that displays a Conga Grid of Opportunities. Selecting a single record displays a Multi-Tabular Conga Grid that includes a Tasks tab. This displays a list of tasks and their associated email conversations.

Conga Grid Feature Settings

To access Grid Feature Settings:

- Go to **Grid Setup**
- Click on **Global Defaults**
- Click on **Feature Security**

The following feature settings are available:

- Action Column Toggles the Action Column on and off (view and edit links).
- Add Contacts to Emma Group Action to add one or multiple Contacts to Emma Groups (integration).

- Add Record (Inline) Gives the option to add new records inline on a grid. Add Record (New Window) Gives the option to add a record in a new window rather than inline.
- Advanced Copy/Paste - Allows the user to designate which object a lookup should be associated with.
- Batch Add - AG License Action to assign Grid Licenses to one or multiple users at a time.
- Batch Add - Campaign Members Action to quickly add one or multiple Contacts or Leads to Campaigns.
- Batch Add – Chatter Action to send out Chatter messages to one or multiple records at a time.
- Batch Add – Note Action to add Notes to one or multiple records at a time.
- Batch Add - Permission Sets Action to assign permission sets to one or multiple users at a time.
- Batch Add – Product Action to add a Product to one or multiple Opportunities, Orders, or Quotes at a time
- Batch Add - Request Action Action to request field updates on one or multiple records at a time.
- Batch Add – Task Action to create tasks for one or multiple records at a time.
- Batch Clone Action to create Clones of one or multiple records at a time.
- Batch Delete Action to Delete multiple records at a time.
- Campaign Member Loader (Legacy app) Allows users with the legacy Campaign Member Loader app to run it directly from the grid (integration).
- Checkbox Column Toggles the selection checkboxes for a grid.
- Column Settings Toggles the Column Settings for a grid which one can find by right clicking on a column header. Column Settings include setting Label, Default Value, Read Only, and Required attributes for a column on a grid.
- Conditional Formatting Toggles the ability to use Conditional Formatting for a grid.
- Conditional Formatting Legend shows a legend at the bottom of the Grid for each conditional formatting rule within the view.
- Configure Toggles the ability to use the field chooser and the options under the Configure button. These options are: Fields, Conditional Formatting, Formulas, Filter Logic, Manage Views, Keyboard Shortcuts, and Help.
- Detail Grid Toggles the ability to use the Detail Grid within the reading pane. This is a tab that allows the user to view data vertically rather than horizontally.
- DrawLoop Retired setting that formally allowed integration with Drawloop.
- Drill Down Toggles the ability to select one or multiple records and drill down to child records in a new window.
- Drill Up Toggles the ability to select one or multiple records and drill up to parent record information in a new window.

- Edit Mode Toggles the ability to edit information inline on a grid.
- Editable Parental Fields Toggles the ability to edit parent record information inline on a grid.
- Export to CSV controls the Export to CSV action.
- Export to CSV with Record IDs controls the Export to CSV with Record IDs action.
- Export to Excel Toggles the ability to select one or multiple records and export the information to excel.
- Export to Excel with Record IDs controls the Export to Excel with Record IDs.
- Fill Down Toggles the ability to make a change to a field, select one or more records, and apply the edited value to all selected records.
- Filter Logic Toggles the ability to see what the current filters are on a grid in the Filter Logic window.
- Find/Replace Toggles the ability to find and replace values for one or multiple records at a time.
- Follow Records Toggles the ability to follow one or multiple records at a time.
- Font Size allows users to set the font size for any given view.
- Formula Fields Toggles the ability for the creation of formula fields on a grid.
- Formula Mass Update Toggles the ability to update one or multiple records using the values from a formula field created on a grid.
- Fullscreen Mode Toggles the ability to put a grid into full screen mode.
- Geopointe Toggles the ability to select one or more records and view them in a Geopointe map (integration).
- Grid Required Field Prefill controls whether or not required fields get prefilled with a default value.
- Hover Detail Tips (Lookup) Toggles the ability to hover over a lookup and see details about the linked record in a mini layout.
- Lightning Component allows users to add lightning components to the reading pane.
- ListBrowse Toggles the ability to select multiple records and view them as a collection of records that can be paged through. The ListBrowse feature is deprecated and no longer supported by Conga.
- Manage Views Toggles the ability to open the manage view window from a grid.
- MapAnything Toggles the ability to select one or more records and view them in a MapAnything map (integration).
- Mass Update Large Batch (disabled by default) Toggles the ability to update more than 1000 records at a time by allowing the user to update all pages rather than just one.
- Mass Update Single Page Toggles the ability to update the values of multiple records at a time.
- Merge Toggles the ability to merge duplicate records together using DupeCheck (integration).

- Paste to Batch Insert Toggles the ability to paste the information in from an Excel Spreadsheet to insert new records.
- Paste to Batch Update (disabled by default) Toggles the ability to paste the information in from an Excel Spreadsheet to overwrite existing records.
- Reading Pane Toggles the ability to use the reading pane in a grid to display related information.
- Rename Columns Toggles the ability to rename a column on a grid.
- Reorder Columns gives users an additional interface for reordering columns within the Grid.
- Save Views Toggles the ability to save views, save data, and revert data that has been changed on a grid.
- Secure Mass Update (Legacy app) Allows users with the legacy Secure Mass Updater app to run it directly from the grid (integration).
- Send Email Toggles the ability to send emails directly from a grid.
- Show Footer causes the footer to automatically hide unless the user is scrolling through the view.
- Summarize Column Toggles the summary information seen at the bottom of columns with numerical values.
- Toolbar Toggles the availability of the toolbar on a grid. View in New Window Toggles the ability to view a record in a new window.
- View In New Window opens selected records in new windows.
- Wrap Text allows users to wrap text for text fields under the column settings.

Conga Grid Technical Requirements

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

Salesforce® Edition:

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

Internet Browsers (latest stable version of):

- Firefox®
- Chrome®
- Internet Explorer® (version 11 or the latest version supported by Salesforce)
- Safari® (Mac only)
- Microsoft Edge

Add a Related List to a Grid

1. Click **Configure** or with the Reading Pane open click **+** (the plus sign)
2. Expand the Related List item
3. Click the check box for the object to add the related list
   Any related list that you could add on a page layout can be added through the related list section.
4. Added related lists will show up as a tab in the reading pane
5. Click the dropdown next to **Save**
6. Click **Save View**
   Add fields to a Related List tab
7. Open the related list you want to modify and click **Configure**
8. Click the check box next to the fields you want to add
9. Click the dropdown next to **Save**
10. Click **Save View**

Copy and Paste Records

Conga Grid currently supports copy-and-paste operations from Google Docs®, Microsoft Excel® (desktop only), and Apple Numbers®.

There are two ways to paste records into Conga Grid℠:

Paste to Batch Insert: This is the default operation. Conga Grid adds new rows with the records that you paste into the grid. This ensures that you do not override existing data.

Paste to Batch Update: When you paste data into Conga Grid, it updates and overrides existing records. You receive a warning message confirming that you want to do this.

You can set the default option using Feature Security:

1. Click **Conga Grid Setup**.
2. Click **Global Defaults** in the object list.
3. Click the **Feature Security** tab.
4. Scroll down to the **Paste to Batch Insert** and **Paste to Batch Update** features.

5. Choose to **Enable**, **Disable**, or **Inherit** (from the user's Salesforce permissions) each option.
   How to Copy and Paste
6. Select the first editable field in that row.
7. Copy the data you want to insert.
8. Right-click paste or use the CTRL + V function to paste the copied data into the Grid.
9. Click **Save** to save the new or updated records.

> ⓘ **Tip**
>
> Ensure the column headers in the content you are pasting match exactly with the column headers in your Grid.

Tips for Copy and Paste

If data does not paste correctly into Conga Grid, formatting issues might be to blame. For example, Excel tries to automatically detect data types and apply formatting. Dates might convert to preferred cultural settings, or numeric totals might have currency symbol prefixes. If data does not paste as expected, we recommend changing the formatting in Excel, Google Docs, or Apple Numbers to match the displayed data in Conga Grid. Cultural settings on your system might not match those of your browser or the Salesforce user. When data is pasted into the grid an extra row is created that allows you to paste additional data or create new records.

Grid Product Data Limits

The below table summarized the limits for Conga Grid.

Conga Grid	Limits
Rows per View	1000
Rows per Paste	100
Rows per Standard Mass Update	1000

Navigating the New Conga Grid

New Conga Grid Layout

The Conga Spring '19 release comes with a new user interface for Conga Grid. In particular, this release contains updated buttons and groupings intended to make the Grid more simplistic and intuitive. The standard Grid user interface configuration is depicted below:

Add Function

- **Add** - Behavior varies depending on Grid Security settings.
    - If both Add In New Window and Add Inline are enabled, a new record is added inline.
    - If Add In New Window is enabled but Add Inline is disabled, a new record is added in a new window.
    - If Add In New Window is disabled but Add Inline is enabled, a new record is added inline.
- **New Window** - Allows users to add new records in a new window.
- **Inline** - Allows users to add new records inline.

Save Function

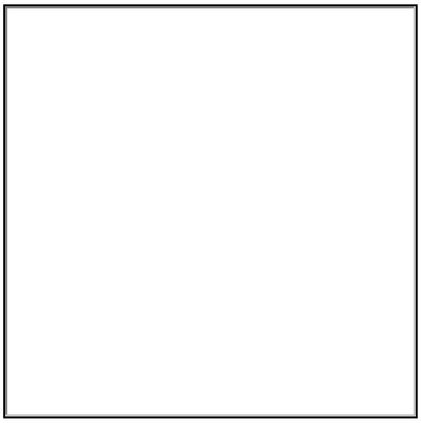
Save - Behavior varies depending on what changes have been made

- If only changes to the view have been made, the view is saved.
- If only changes to the data have been made, the data is saved.
- If changes to the data and the view are made, the data is saved and the user is prompted to save the view.

Save View - Saves any changes that have been made to the current view.

Save View As - Allows users to save additional details of the current view or create a copy of the current view.

Save Data - Saves changes made to records within the view.



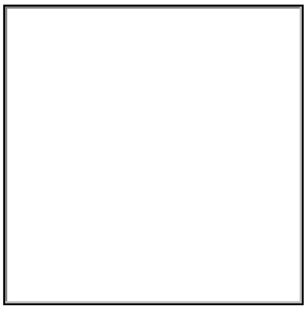
Recently Viewed - Shows the last five views a user has accessed per object for a Grid. This section will only show the current view if the user has no recent views.

List All Views - Displays the full list of views for the current object that is available to the user to select and load.

Manage Views - Displays a grid of views for the current object that is available to the user and presents multiple actions to perform on selected views.

New View - Creates a new view for the current object.

Actions Menu



Contains the standard actions for Grid. For information about the different actions available and their functionality, see Conga Grid Feature Settings

Custom Menu

Displays any custom actions that an organization may have built to be displayed in the toolbar. This will not exist on every Grid and contents will vary by organization.

Configure Function

Gear Icon - Opens the field menu on the right hand side of the Grid.

Fields - Opens the field menu on the right hand side of the Grid.

Conditional Formatting - Allows users to create and set conditional formatting rules for the data in the view.

Formulas - Allows users to create formulas for the data in the view.

Filter Logic - Displays a list of all filters set for a view and allows for more complex boolean logic to be set.

Clear Filters - Clears all filters set for a view.

Clear Formatting - Clears all conditional formatting rules set for a view.

Reading Pane

Reading Pane Icon - Opens the reading pane on the right hand side of the Grid.

Right - Positions the reading pane on the right hand side of the Grid.

Left - Positions the reading pane on the left hand side of the Grid.

Bottom - Positions the reading pane on the bottom of the Grid.

Top - Positions the reading pane on the top of the Grid.

Reset Function

Reset Icon - Resets the current view.

Reset to Default - Resets the Grid to the Default view for the current object.

Revert Data - Reverts any changes that were made to the data in a view but have not yet been saved.

Help Function

Help - Opens the Conga Support Site.

Keyboard Shortcuts - Displays a list of keyboard shortcuts that can be performed within a view.

New Conga Grid UI Explained

There are a number of noticeable changes in Conga Grid with the Conga Spring '19 release, providing users with a user interface that has a new look and feel. Do not worry! We are here to help you get oriented and back on your way.

What's New & Different?

A New Toolbar

Updates were made to the buttons and items within each grouping, resulting in a cleaner and more intuitive toolbar. The updates make it easier for both seasoned and new Grid users to find their way around. See the information below for more details on the toolbar updates:

1. Everything involved with configuring a view resides under the Configure gear icon.
   - The Configure section now contains Clear Filters and Clear Formatting.
2. Everything that takes action on your data lives under the actions dropdown.
3. Everything involved with managing and selecting views is under the view dropdown.
   - The new Recently Viewed section allows you to see your most recent views and easily pick back up where you left off.
   - The new List All Views window allows you to easily search for and load your views.
   - The new View & Manage Views now live under the view dropdown instead of Reset View and Configure respectively.
4. The Reset View icon dropdown now contains Revert Data.
5. A new Custom Action dropdown was added to hold Custom Action buttons.
6. Buttons can no longer be reordered, but buttons can still be taken on and off the toolbar.
7. A new help dropdown contains a help link to the Conga support site and the keyboard shortcut option.

An Overall User Interface Refresh

Grid has a new look that gives it a more natural and updated feel in both Salesforce Classic and Salesforce Lightning. Details regarding the User Interface Refresh include:

1. New Fonts
2. New Windows

3. New Buttons
4. New Icons
5. New Alerts

Common Questions

Will I need to rebuild all my Grids and actions?

No - the new changes are automatically applied to your existing Grids and actions. No work needs to be done on your part, and every view, tab, and action is available for use right away.

Will my Grids still have the same functionality?

Yes - Changes like these can be jarring, but everything that you've come to know and love about Conga Grid's functionality still remains.

Where did my Views go?

The views are still there! It is likely you need to access more views to add them to your recently viewed list. To see your available views, click the View dropdown and select List All Views. This brings up a list of all of your available views to search from and load.

Can the new UI be disabled?

No - it cannot be disabled. Where you could previously switch between the interfaces, we now only allow one. This is in an effort to have a similar experience across the board for all of our customers.

Why did you make these changes?

A few reasons:

1. Navigating the toolbar was confusing at times. We wanted to make it easier for users both new and old to find their way around.
2. The Conga Grid classic UI did not match the Conga brand. We wanted to give the product a refresh that lets people know it's a part of the Conga family.
3. Similarly, we wanted Conga Grid to feel familiar right out of the gate. Many of the changes made are to help our users feel at home by aligning with what is seen in standard Salesforce Lighting List Views.

What Else?

Please use the resources below to get familiar with the new Grid changes:

Conga Grid UI Changes - Learn more about the specific changes to the Grid UI.

- Video (Coming Soon!)
- Webinar (Coming Soon!)

Opportunity Partners Behavior

There are two separate behaviors expected when using Opportunity Partners in Conga Grid.

Salesforce Classic

In Salesforce Classic, users have the ability to view Partner records and create Partners in a new window by clicking the new record button. Due to Salesforce restrictions on the object, no inline capabilities are supported.

Salesforce Lightning

In Salesforce Lightning, users have the ability to view Partner records, but cannot add them in a new window or inline. These options are disabled because the object is not currently supported in Salesforce Lightning.

Modifying the Toolbar

To access these options, take the following steps:

1. Click the **Configure Gear**.
2. From the menu that appears on the right-hand side, open the **Toolbar** grouping:

The following options are available:

Option	Description
Actions	Toggles the Actions button on the toolbar.
Custom Actions	Allows users to toggle whatever custom action buttons they have available for the current object.You will not see Custom Actions as an option under Toolbar if no Custom Action buttons are available. These buttons can also be toggled at the Global and Object level through Conga Grid Setup with Conga Grid Feature Settings.
Reading Pane	Toggles the Reading Pane button on the toolbar.
Reset View	Toggles the Reset View button on the toolbar.
Save	Toggles the Add and Save buttons on the toolbar.

Option	Description
Search	Toggles the search bar on the toolbar.
Setup	Toggles the configure button on the toolbar.
Views	Toggles the view dropdown on the toolbar.

Table 1. Toolbar Options

Using the Approval Center

The Conga Grid Approval Center is a dashboard integrated with Conga Intelligent Workflow Approvals. Here you can easily approve, reject, and manage your pending approvals. For information on how to access the Approval Center, see Installing and Setting up the Approval Center. For information on how to use the Approval Center, see Approval Center.

Installing and Setting up the Approval Center

The following products and packages are required for installation and setup of the Approval Center:

- Conga Intelligent Workflow Approvals
- Conga Grid Version 3.0
- Conga Approvals 11.3

To install and setup the Approval Center in Conga Grid, take the following steps:

1. Install Conga Intelligent Workflow Approvals. For more information, see Approvals.
2. Install the latest package of Conga Grid. For more information, see Install or Upgrade Conga Grid.
3. Install the Conga Approvals extension package.
4. Follow the instructions in Manage Licenses and Assign Permission Sets to assign permission sets.
5. In Salesforce, go to **Setup**.
6. in the Quick Find search bar, enter "Custom Settings" and click the **Custom Settings** link.
7. In Custom Settings, click **Approvals System Properties**.
8. Click **Manage**.
9. Click **Edit** next to System Properties.

10. Go to the Approval Center field and enter your object names (example: Apttus__APTS_Agreement__c). This is a comma separated list of API names to include in the Approvals Center. The first item in the list is used as the default tab if one or more requests exist for that Object when the user navigates to the dashboard. Only objects configured to display Approvals will display correctly in the Approval Center.
11. Click **Save**.
12. In the Quick Find search bar, enter "App Manager" and click the **App Manager** link.
13. In App Manager, go to **Conga Approvals**. Under the right hand arrow icon, click **Edit** from the dropdown list.
14. In the Available Tabs section, click on the name of the tab you want to display in Approvals Center, and click the **Add** arrow to add it to the Selected Tabs section.
15. Click **Save**.
16. From the Salesforce App Launcher, go to **Conga Approvals**.
17. Click on the **More** tab > **Approval Center** to view and use the Approval Center.

Review the Building Conga Grid Solutions section that includes configuration topics such as how to create a view. For information on how to use the Approval Center, see Approval Center.

Build Conga Grid from existing Salesforce data

Information on how to build Conga Grids from existing Salesforce data, such as List Views and reports.

Build a Conga Grid from existing Salesforce data

The steps in this article are meant for new Conga Grid℠ and Salesforce® users who are not familiar with configuring Conga Grid. If you have experience using Conga Grid, you can create a new grid or customize an existing grid using the methods described in Conga Grid Configuration.

When asked how you want to set up Conga Grid, choose one of the options described below.

- Use an Existing Salesforce App
- Use Existing List Views
- Use an Existing Salesforce Report
- Scan My Data

Convert Reports to Conga Grid Views (Beta)

Use the New Grid From Report (Beta) tool to convert Salesforce reports into Grid views.

This is a beta feature that is still in development. Please be aware that some issues may exist and that we cannot currently offer technical support.

This is a standalone tool that requires you to manually select various options. Consider using the Import from Existing Salesforce Quick Start option for a more automated solution.

To use the Import from Existing Salesforce Quick Start option for a more automated solution:

1. Click Grid Setup.
2. In the object list, click Global Defaults.
3. Under the Global Defaults
4. Global Setup tab, click Convert Report (Beta).
5. The New Grid from Report screen appears.
6. From the Report drop-down list, select a report that you want to create a Grid view from.
7. Optional: Select the Create VF Page check box to create a Visualforce page from the view.

8. The Label and Name automatically populate based on the report that you selected.
9. Optional: Enter a Description for the new view.
10. Optional: Select the Add Page to AG Users Permission Set checkbox to automatically add the new Visualforce page to the Grid Users permission set. This option is only available if you chose to create a Visualforce page in Step 5.
11. Click OK. The Map Columns screen appears.
    Grid tries to determine the best match between its own field names and the Salesforce report field names. In the example above, Case Owner is the Salesforce field, and OwnerId is the Grid field that was the closest match.
12. To choose a different Grid field, click the button to the right of the field name. Then select a field in the Map Column screen that appears.
13. Click OK in the Map Columns screen. A Visualforce Page Created screen appears.
14. Choose an option:
    - Click View VF Page to go to the Visualforce Pages page in Salesforce Setup, where you can manage the Visualforce page that was created. This option is only available if you selected to create a Visualforce page in Step 5.
    - Click Tabs to go to the Custom Tabs page in Salesforce Setup, where you can create a custom tab that contains the new view.
    - Close the Visualforce Page Created screen to return to ActionGrid Setup.
15. Click Grid Explorer.
16. From the object list, select the object associated with the new view.
17. Select the new view from the Views drop-down list in the Grid toolbar.

If Grid selects the CreatedByID field as the closest match for the Salesforce Created by ID field, click the button to the right of the field name and select CreateByUser instead.

Use an Existing Salesforce App

This option converts list views from a selected Salesforce app into separate Conga Grid views. It also creates a new app with Conga Grid tabs that match objects in the original app.

Before You Begin

Ensure that you belong to the Conga Grid Visualforce Pages permission set.

If not, add yourself to the permission set:

1. Click **Conga Grid Explorer**.
2. Select **User** from the object list.
3. Select the record with your user name.

4. Click **Actions** and click **Batch Add > Permission Sets**. The Batch Add Permission Set screen appears.
5. From the **Permission Set** drop-down list, select **Conga Grid Visualforce Pages**.
6. Click **OK**. The Results screen appears.
7. Click **Close**.

Create an App

To create an App:

1. Click the **Conga Grid Quick Start** tab.
2. Click Create Custom Conga Grid.
3. Click Create from Salesforce App.
4. Select an app from the left side of the screen.
5. Click Create New App. When processing is complete, an App Created page appears.
6. Click the **App Menu** in Salesforce and select the new app. It is labeled the same as the original app with "(AG)" appended; for example, Marketing (AG). The new app includes the same tabs as the original app with "(AG)" appended; for example,

Opportunity (AG).

This process also created Conga Grid views from the list views of the selected app.

To access the Conga Grid views:

1. In the Salesforce **App Menu**, click **Conga Grid**.
2. Click **Conga Grid Explorer**.
3. In the object list, select an object associated with the original Salesforce app (for example, Opportunity).
4. Select a view from the **View** drop-down list in the Conga Grid toolbar.

ⓘ **Tip**

To see a list of the views that were created for the selected object, click **Configure** and click **Manage Views**. In the Manage Views screen, click the **Created Date** field header twice to sort the created date in descending order; for example:

You can rename the views, share them, delete them, or set their privacy as needed. See Manage Views for details.

Use Existing List Views

This option converts one or more Salesforce list views to Conga Grid views, including all fields and filters that were applied.

To use existing list views

1. Click the **Conga Grid Quick Start** tab.
2. Click Create Custom Conga Grid.
3. Click Create Conga Grid View.
4. Select an object that contains the list views that you want to convert to Conga Grid views.
5. Click **Install**. When processing is complete, a success screen appears.

To access new Conga Grid views

1. In the Salesforce **App Menu**, click **Conga Grid**.
2. Click **Conga Grid Explorer**.
3. In the object list, select the object associated with the list views.
4. Select a view from the **View** drop-down list in the Conga Grid toolbar.

Tip

To see a list of the views that were created for the selected object, click Configure and click Manage Views. In the Manage Views screen, click the Created Date field header twice to sort the created date in descending order; for example:

You can rename the views, share them, delete them, or set their privacy as needed. See Manage Views for details.

Guided Tour of Conga Grid

Take a guided tour of Conga Grid℠.

Building Conga Grid Solutions

Learn in-depth essentials of working with Conga Grid℠, including configuration, setting up Multi-Tabular related lists, managing views, and using actions.

Conga Grid Configuration

Learn about different configuration options in Conga Grid℠.

- Options for Starting Conga Grid
- Single Conga Grid Tab
- Single Related List
- Conga Grid Buttons
- Create Buttons for List Views and Search Results
- Create an Add Inline Option
- Enable Conga Grid in the Salesforce Console
- Configuration Tips
- Configuring Conga Grid for Lightning
- Convert Salesforce Apps to Conga Grid (Beta)
- Convert Salesforce List Views to Conga Grid Views (Beta)
- Edit Parental Data
- Lightning Component Tab Labels

Options for Starting Conga Grid

The **Tabs & Related Lists** tab contains wizards that guide you through steps to create Conga Grid℠ tabs and related lists for selected objects. To access this tab, click **Conga Grid Setup** > **Conga Grid Settings** in the object list.

See the following topics for instructions on using these wizards.

Tab Creation

Multi-Tabular Conga Grid

Single Conga Grid Tab

Page Layouts of Related Lists

Multi-Tabular Related List (Automatic)

Multi-Tabular Related List (Manual)

Single Related List

You can also create buttons that display Conga Grids. Place the buttons in your list views and search result layouts or at the top of related lists. See the Conga Grid Buttons topic for details.

Single Conga Grid Tab

To create a Single Conga Grid Tab, you must first create a Visualforce page and then create a tab for that page.

Create a Visualforce Page

1. Click **Conga Grid Setup**.
2. Select an object from the object list for which you want to create a Conga Grid tab.
3. Click the **Tabs & Related Lists** tab.
4. Click **Single Conga Grid** in the Tab Creation section.
5. If any views have been created for the related list object, you can select a view from the **Default View** drop-down list. Leave it blank for no view. Or, click **Create new view for this Page** to automatically create a view that you can later redefine when you load the page.
6. Enter a **Label**, which is the name of the list as it appears on the Salesforce page layout. Keep the label short. Longer labels are more difficult to fit into their assigned fields, and they can be easily lost if you have multiple Visualforce pages.
7. Enter a **Name** that will make this list unique in Salesforce, using underscores instead of spaces.
8. Enable the Conga Grid℠ features you want by selecting their check boxes. Any cleared options can be re-enabled (and vice-versa) from the Visualforce page markup code after creating the related list.
9. Click **OK**. A Visualforce Page Created screen appears.

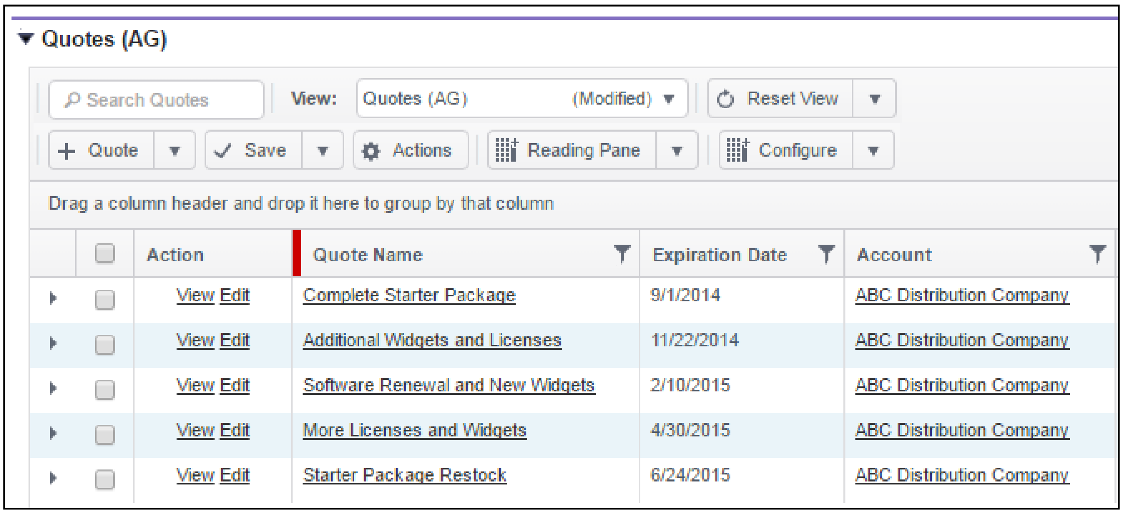
Create a Single Conga Grid Tab

1. Click **Tabs** in the Visualforce Page Created screen.
2. In the Visualforce Tabs area, click **New**.
3. From the **Visualforce Page** drop-down list, select the page that you just created.

4. Enter a **Tab Label**. The Tab Name will automatically populate based on the label.
5. Select a **Tab Style** to use for your object.
6. Leave the default option of **None** for **Splash Page Custom Link**.
7. (Optional). Enter a **Description** for the tab.
8. Click **Next** in the lower-right corner of the page.
9. Select the **Profile** and **Visibility** options for this tab.
10. Click **Next**.
11. Select which **Apps** in the Salesforce App Menu should include the tab.
12. Click **Save**.
13. Click **All Tabs** in the Salesforce® menu.
14. Click **Customize My Tabs**.
15. Select the new tab in the **Available Tabs** list, then click **Add** to move it to the **Selected Tabs** list.
16. Click **Save**.

Single Related List

Here is an example of a Quotes (AG) Conga Grid℠ related list within a Contact page layout:



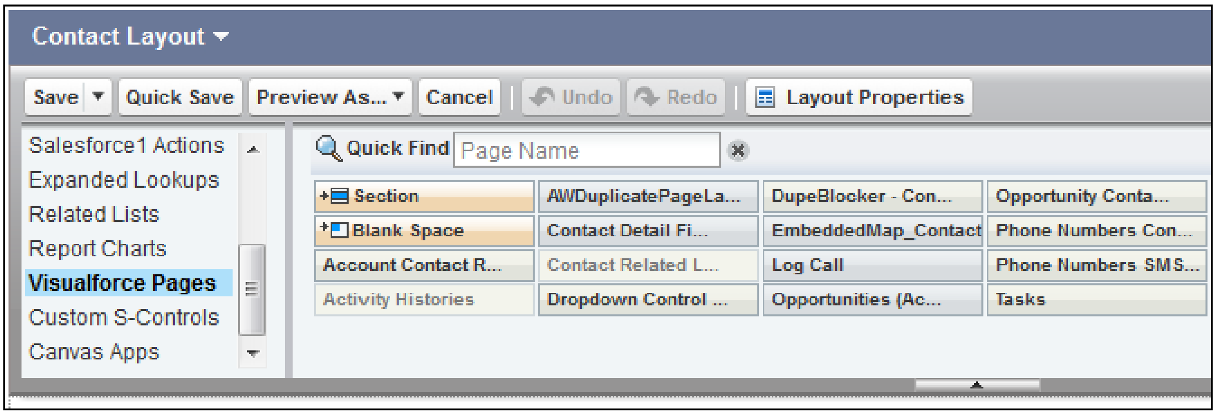
Drag a column header and drop it here to group by that column

	□	Action	Quote Name	Expiration Date	Account
▸	□	View Edit	Complete Starter Package	9/1/2014	ABC Distribution Company
▸	□	View Edit	Additional Widgets and Licenses	11/22/2014	ABC Distribution Company
▸	□	View Edit	Software Renewal and New Widgets	2/10/2015	ABC Distribution Company
▸	□	View Edit	More Licenses and Widgets	4/30/2015	ABC Distribution Company
▸	□	View Edit	Starter Package Restock	6/24/2015	ABC Distribution Company

Follow these steps:

1. Click Conga Grid Setup.
2. Select the object for which you want to make a related list.
3. Select the Tabs & Related Lists tab.
4. Click Single Conga Grid in the Page Layout section.
5. In the New Related List screen, click the Select Relationship drop-down list and select the related object you want to see as a Conga Grid.

6. If any views have been created for the related list object, you can select a view from the Default View drop-down list. Leave it blank for no view. Or, click Create new view for this Page to automatically create a view that you can later redefine when you load the page. You can also hard-code a view to be the default view. See Hard-Code a Default View using a View ID.
7. Enter a Label, which is the name of the list as it appears on the Salesforce® page layout. Keep the label short. Longer labels are more difficult to fit into their assigned fields, and they can be easily lost if you have multiple Visualforce pages.
8. Enter a Name that will make this list unique in Salesforce, using underscores instead of spaces.
9. Select the Conga Grid features you want using the check boxes. Any disabled features can be re-enabled (and vice-versa) from the Visualforce page markup after creating the related list.
10. (Optional). Select the Add Page to AG Users Permission Set check box if you want to automatically add the resulting Visualforce page to the Conga Grid Users permission set.
11. Click OK. A Visualforce Page Created screen appears.
12. If you are using any version of Salesforce except Professional Edition, click Page Layout in the Visualforce Page Created screen. If you are using Professional Edition, you must manually add related lists to page layouts using Visualforce Markup. Please see the instructions for Professional Edition below.
13. Click the Edit link next to the page layout you want your related list in.
14. On the left side of the object Layout area, scroll down the list and click Visualforce Pages.
15. Drag the Section item to a preferred location in the page layout.



16. Drag the related list Visualforce page to the new section.
17. Click Save.

ⓘ **Tip**

For best results, set the Section Height (in pixels) to 500 or higher.

Professional Edition

Salesforce Professional Edition users cannot use Conga Grid Setup to create Visualforce pages of related lists. Follow these instructions instead:

1. In Salesforce, click Setup.
2. In the App Setup section, click Develop and click Visualforce Pages.
3. Click the Edit link next to the new Visualforce page.
4. Select the Visualforce Markup tab.
5. Copy and paste the following code into the markup field. Modify the boldface values in the example to match the object to start on, along with the related lists to view for that object. The example shows an Account with a Contact related list.

```
<apex:page standardController= "Account" > <CRMC_PP:Grid ObjectName= "Contact"  FKName= "AccountId"  FKValue= "{!Account.Id}"  />
<CRMC_PP:DrillUp /> <CRMC_PP:FollowRecords /> </apex:page>
```

6. Click Save. Next, you will add the new Visualforce page to the Page Layout.
7. In Salesforce, click Setup.
8. In the App Setup section, click Customize.
9. Click the parent object and click Page Layouts.
10. Continue with Steps 12-14 above.

Conga Grid Buttons

The method described here is not supported in Salesforce® Professional Edition. For that edition, administrators need to manually add buttons by following the instructions in Create Buttons for List Views and Search Results.

Warning

- Conga Grid buttons created before version 1.89.X are not supported in Salesforce Communities. To use Conga Grid buttons in Communities, follow the steps below to create version 1.89.X or above is installed.
- Conga Grid buttons are not supported in Salesforce Lightning.

1. Click **Conga Grid Setup**.
2. Select the object for which you want to create a button.
3. Click the **Buttons** tab.

4. Click **Create Button**.
5. Choose where you want the button to appear:
    - Click **List View Search Layout** to add the button to the list view. The List View page for the selected object appears.
    - Or, click **Search Results Search Layout** to add the button to the search results. The Search Results page for the selected object appears.
6. In the Available Buttons list, click **Conga Grid** and click **Add** to move it to the Selected Buttons list.
7. Click **Save**.
8. Go to the selected object in Salesforce and click **Go!** next to your view.

The Conga Grid button will appear in the menu of buttons in your list view.

Create Buttons for List Views and Search Results

You can create button for List Views and Search Results.

To create buttons:

1. In Salesforce®, click **Setup**.
2. In the App Setup section, Click **Customize**.
3. Click the drop-down arrow for the object you want to customize.
4. Click **Buttons**, **Links**, and **Actions**.
5. Click **New Button** or **Link**.
6. Populate the fields as follows:
    - Label: **Conga Grid**
    - Name: **crmc_Grid**
    - Display Type: **List Button**
    - Display Checkboxes (for Multi-Record Selection): **Enabled**
7. Click **Save**.
8. In Salesforce Setup, click the drop-down arrow again for the object you want to customize.
9. Click **Search Layouts**.
10. Click **Edit** next to **object List View** or **Search Results**, depending on where you want to manually add the button.
11. Follow the steps in Conga Grid Buttons.

If you are adding a button for a related list (instead of having an embedded Conga Grid℠), populate the *ForeignKeyName* and *ForeignKeyValue* fields using the example below. This example shows Contacts for an Account page layout.

```
(function (e,t){var n=document.getElementsByTagName("HEAD").item(0), r=document.createElement("script");
r.type="text/javascript", n.appendChild(r), r.onload=function(){GridButton.showInGrid(e,t,'AccountId','{!
Account.Id}')},
r.src="/resource/crmc_pp__gridbutton?ts="(new Date).getTime()}("Contact", {!
GETRECORDIDS( $ObjectType.Contact)}));
```

Here is the generic code:

```
(function (e,t){var n=document.getElementsByTagName("HEAD").item(0), r=document.createElement("script");
r.type="text/javascript", n.appendChild(r), r.onload=function(){GridButton.showInGrid(e,t,'<ForeignKeyName',
'{!<ForeignKeyValue>}')},r.src="/resource/crmc_pp__gridbutton?ts="+(new Date).getTime()}("<ChildObject>",
{!GETRECORDIDS( $ObjectType.<ChildObject>}));
```

Create an Add Inline Option

To create an Add Inline Option, take the following: these steps:

1. Click **Conga Grid Setup**.
2. Select an object from the object list.
3. Click the object Features tab.
4. Scroll to the features **Add Record (Inline)** and **Add Record (New Window)**.

Features and Limitations

- If both options are enabled, the New object button will open a new window by default. It will have a drop-down menu that lets you select between the Add Inline option and the normal page layout in a new window.
- If you have an embedded or preview-type Conga Grid℠ and both features are enabled, the default behavior is Add in Place.
- If Add Record (Inline) is enabled and Add Record (New Window) is disabled, the New object button will default to Add in Place and will not open in a new window.
- If Add Record (Inline) is disabled and Add Record (New Window) is enabled, the New object button will open the new record in a new browser window or tab, and you cannot add it in place.
- If both Add Record (Inline) and Add Record (New Window) are disabled, the New object button will be greyed out and you will not be able to create new records using the New object button.
- As with other Conga Grid actions and features, you can apply these settings using Global defaults, by profiles, or for individual users.

Enable Conga Grid in the Salesforce Console

Follow these steps:

1. In Salesforce®, click **Setup**.
2. In the Build section, click **Create** and click **Apps**.
3. Choose the **Console** you would like Conga Grid℠ to appear in.
4. Find the Visualforce tabs you created, and move them from the Available Items list to Selected Items.
5. Click **Save**.

Configuration Tips

See the following sections:

- Configuration Tips
- Enable Features for Related Lists and Visualforce Pages
- Name Visualforce Pages

Default Columns for Standard and Custom Objects

For standard objects, find a field set named CRMC_Action_Grid_Defaults under the object in question. You can manipulate this field set to meet your company needs.

For custom objects, create a new field set under the object in question titled CRMC_Action_Grid_Defaults. Add the default fields you want end users to see. When they load the Conga Grid℠ for that object, they will see the columns defined in that field set. They can always redefine their fields and save views to meet their personal preferences.

Enable Features for Related Lists and Visualforce Pages

Follow these steps to enable or disable features of a Conga Grid related list or Visualforce page:

1. In Salesforce®, click **Setup**.
2. Click **Develop** and click **Visualforce Pages**.
3. Click the **Edit** link next to the Visualforce page.
4. Click the **Visualforce Markup** tab.
5. In the Grid ObjectName section of code, change the various features to **true** or **false**. If the value is true, the feature will be enabled.
6. Click **Save**.

Name Visualforce Pages

We recommend using a prefix of "AG" when naming and labeling Visualforce pages for use with Conga Grid. This helps discern which pages are related to Conga Grid, and it helps them visually stand out to end-users.

Configuring Conga Grid for Lightning

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®.

To enable you to put your Conga Grid℠ Visualforce pages into Salesforce Tabs and on Lightning Page Layouts:

1. Go to your Grid Visualforce page
2. Check the **Available for Lightning Experience, Lightning Communities, and the mobile app** checkbox

You will now be able to put your Grid℠ Visualforce pages into Salesforce Tabs and on Lightning Page Layouts

Convert Salesforce Apps to Conga Grid (Beta)

Use the New Conga Grid℠ App Page (beta) tool to convert a selected Salesforce® app to a Conga Grid view. This tool provides the following options:

- Convert list views from a selected Salesforce app into separate Conga Grid views.
- Create a Visualforce page with tabs that match objects in the app.

Warning

This is a beta feature that is still in development. Please be aware that some issues may exist and that we cannot currently offer technical support.

This is a standalone tool that requires you to manually select various options. Consider using the Import from Existing Salesforce Quick Start option for a more automated solution.

Follow these steps:

1. Click **Conga Grid Setup**.
2. Select **Global Defaults** in the object list.
3. Click the **Global Setup** tab.
4. Click **New Conga Grid App Page (beta)**. The New Conga Grid App screen appears.
5. Select a Salesforce app from the **App** drop-down list.
6. Select the **Convert to MultiTab** check box if you want to create a Multi-Tabular Conga Grid Visualforce page with tabs that match objects in the app. If you do not select this check box,
7. Enter a **Label** for the new view. This will be the name of the view in Conga Grid. The Name field automatically populates based on the Label.
8. Enter an optional **Description** for the view.
9. Select the **Create Views** check box if you want to convert Salesforce list views to separate Conga Grid views. Be aware that if you have hundreds of list views, this option will create separate Conga Grid views for each one.
10. Select a privacy option for the view from the **Privacy** drop-down list.
11. Select the **Inline Edit Mode** check box to enable inline editing in the view.
12. Select the **Reading Pane** check box to enable the Reading Pane to display in the view.
13. Select the **Toolbar** check box to display the Conga Grid toolbar in the view.
14. Select the **Add Page To AG Users Permission Set** to automatically add the Visualforce page to the Conga Grid Users permission set. Here is an example with

options selected:

15. Click **OK**.

When processing is complete:

- A Visualforce Page Created screen appears if you selected the Convert to MultiTab option. You can dismiss the dialog, or click **View VF Page** to go directly to the Visualforce page in Salesforce Setup. Or, click **Page Layout** to manage the new Visualforce page layout in Salesforce Setup.
- A Success screen appears if you did not select the Convert to MultiTab option. Click Dismiss. Click the **F5** key to refresh your browser, then click the Apps menu in Salesforce to view the new app.
- If you selected the Create Views option, you can access the new Conga Grid views from the **Views** drop-down menu in the Conga Grid toolbar.

Convert Salesforce List Views to Conga Grid Views (Beta)

Use the Convert ListView (beta) tool to convert one or all Salesforce® list views into Conga Grid℠ views.

> **Warning**
>
> This is a beta feature that is still in development. Please be aware that some issues may exist and that we cannot currently offer technical support.

This is a standalone tool that requires you to manually select various options. Consider using the Import from Existing Salesforce Quick Start option for a more automated solution.

Follow these steps:

1. Click **Conga Grid Setup**.
2. Select the object associated with the list views you want to convert.
3. Click **Convert ListView (beta)**. The New Conga Grid From List View screen appears.
4. To convert all list views to Conga Grid views, click **All List Views**. The remaining options in this window are disabled. Or, select a specific **List View** from the drop-down list.
5. Select whether or not to create a Visualforce page for the selected Conga Grid view.
6. Enter a **Label, Name**, and optional **Description** for the Conga Grid view.
7. Click **OK**.

Edit Parental Data

You can view and edit parent records from the child record. For example, while on a Contact record, you can edit fields on the parent Account record.

1. Make the inline edits (you will receive a notification).

2. Prior to saving the data, the cell will have a blue mark (as opposed to the default red).

3. Click **Save Data**. The parent record field will be updated.

This functionality is enabled by default. To disable, change the feature security for the Edit Parental Data field.

Lightning Component Tab Labels

Edit a Lightning Component Tab Label

To edit or enter a label in the Lightning Component tab, take the following steps:

1. From Conga Grid, click the name of the tab you created.
2. Click on the three horizontal bar menu icon on the top right of the pane.
3. In the Label field, enter or edit the label name.
4. Click Save.

Create a Multi-Tabular Conga Grid Interface

Learn how to create custom Multi-Tabular related lists in Conga Grid℠, using step-by-step wizards or by modifying Visualforce markup code.

- Multi-Tabular Conga Grid
- Multi-Tabular Related List (Automatic)
- Multi-Tabular Related List (Manual)
- Multi-Tabular Related List (Stacked)
- Multi-Tabular Related List (Visualforce Markup)
- Modify Multi-Tabular Related Lists

Multi-Tabular Conga Grid

Use Conga Grid Setup to create a tabbed interface with any Conga Grid℠ underneath each tab. These are configured as Visualforce pages. You can embed tabs on a page layout to replace related lists. Or you can create a top-level tab. A Multi-Tabular Conga Grid can contain multiple views of the same object or multiple objects.

To create a tabbed interface with any Conga Grid underneath each tab:

1. Click **Conga Grid Setup**.
2. Select an object from the object list for which you want to make a Multi-Tabular Conga Grid.
3. Click the **Tabs & Related Lists** tab.
4. Click **Multi-Tabular Conga Grid**. The New Conga Grid MultiTab Page screen appears.
5. Enter a **Label**. The Name field automatically populates.
6. (Optional): Enter a **Description** for the tab and click **OK**.
7. From the **Select Object** drop-down list, choose an object for the first tab.
8. From the **Default View** drop-down list, select an existing Conga Grid view for the selected object. Leave it blank for no view or select **Create new view for this Page** to automatically create a view that can be redefined when you load the tab. You can also hard-code a view to be the default view. See Hard-Code a Default View using a View ID.
9. Select the check boxes of features you want to enable in the Multi-Tabular Conga Grid page layout.

10. Click **Add Another Tab** to display another New Conga Grid MultiTab Page screen. Repeat Steps 7 through 9 for each tab that you are creating. Click **Save and Finish**.
11. In the success screen, click **Tabs**. The Custom Tabs page appears. Under Visualforce Tabs, click **New**.
12. In the New Visualforce Tab section, find your Visualforce page and enter a **Tab Label**. It will populate the Tab Name automatically.
13. Select a **Tab Style** to use for your object.
14. Leave the default option of **None** for **Splash Page Custom Link**.
15. (Optional): Enter a **Description** for the tab. Click **Next**.
16. Select the **Profile** and **Visibility** options for this tab.
17. Select which **Apps** in the Salesforce® App Menu should include the tab.
18. Click **Save**. The new tab appears in the Salesforce Tab menu.
19. Create a permission set so that end users can access the Multi-Tabular Conga Grid. See Create a Permission Set for instructions.

When referencing custom objects, ensure that on the object setting page in Salesforce, "Allow in Search" is checked.

Multi-Tabular Related List (Automatic)

Follow these steps to create a Multi-Tabular page layout of related lists for a given object.

The page layout is configured as a Visualforce page.

1. Click **Conga Grid Setup**.
2. Select the object whose page layout will have the related tabs.
3. Click **Multi-Tabular (Automatic)**. The New Conga Grid MultiTab Page screen appears.
4. Enter a **Label**. The Name field will automatically populate based on the Label.
5. (Optional). Enter a **Description**.
6. (Optional). Select the **Add Page to Layout** check box to automatically add the resulting Visualforce page to a pre-determined location in the selected page layout. This option is disabled by default.

7. (Optional). Select the **Add Page to AG Users Permission Set** check box if you want to add the resulting Visualforce page to the Conga Grid Users permission set.
8. Click **OK**. When processing is complete, a Visualforce Page Created screen appears.
9. Choose the option:
   - To view the resulting Visualforce page in Salesforce Setup, click **View VF Page**.
   - To manually configure the location of the resulting Visualforce page in the page layout, click **Page Layout** and follow the steps in Configure the Page Layout below. You only need to do this if you did not select the **Add Page to Layout** check box.
   - Dismiss the dialog.

Here is an example of a Multi-Tabular page layout of related lists for a selected Account record:

Manually Configure the Page Layout

This section only applies if you did not select the **Add Page to Layout** check box in the New Conga Grid Related Lists MultiTab screen. Follow these steps to add the Multi-Tabular Conga Grid related list to a custom location within the selected page layout.

1. In the object Page Layout screen, click the **Edit** link next to the name of the new page layout.

2. In the object Layout section, click **Visualforce Pages**.

3. Drag the name of the Visualforce page to the desired location within the page layout.
4. Click **Save**.

Multi-Tabular Related List (Manual)

Follow these steps:

1. Click **Conga Grid℠ Setup**.
2. Select an object from the object list for which you want to make a Multi-Tabular page layout.
3. Select the **Tabs & Related Lists** tab.
4. Click **Multi-Tabular (Manual)**. The New Conga Grid MultiTab Page screen appears.
5. Enter a **Label**. The Name field will automatically populate based on the Label.
6. (Optional). Enter a **Description**.
7. (Optional). Select the **Add Page to AG Users Permission Set** check box if you want to automatically add the resulting Visualforce page to the Conga Grid Users permission set.
8. Click **OK**. The New Conga Grid MultiTab Page screen appears.
9. From the **Select Relationship** drop-down list, choose an object relationship for the first tab in the Multi-Tabular Conga Grid.
10. From the **Default View** drop-down list, select an existing Conga Grid view for the selected object. Leave it blank for no view. Or, select **Create new view for this Page** to automatically create a view which you can later redefine when you load the tab.
11. Provide a **Label** for this tab.

12. Select the check boxes of features you want to enable in the Multi-Tabular Conga Grid page layout.
13. Click **Add Another Tab** to display another New Conga Grid MultiTab Page window. Repeat Steps 7 through 10 for each tab that you are creating.
14. Click **Save and Finish**. A success dialog appears.
15. In the success dialog, click **Page Layout**. The object Page Layout screen appears.
16. Click the **Edit** link next to the **Page Layout** name.
17. Select the **Visualforce Pages** option.
18. Drag the related list to your desired location on the Visualforce page.
19. Click **Save**.

Multi-Tabular Related List (Stacked)

Follow these steps to create a Multi-Tabular related list where the tabs are stacked vertically:

1. Click **Conga Grid Setup**.
2. Select **Global Defaults** from the object list.
3. Click the **Global Setup** tab.
4. Click **New MultiTab Page**.
5. Enter a **Label**. The Name field will automatically populate based on the Label.
6. (Optional). Enter a Description.
7. (Optional). Select the **Show the Side Bar** option if you want to display the Configure menu in the grid. This option is disabled by default.
8. (Optional). Select the **Show Header** option if you want to display the field (column) headers in the grid.
9. (Optional). Select the **Add Page to AG Users Permission Set** option if you want to automatically add the resulting Visualforce page to the Conga Grid Users permission

set.

10. Click **OK**. The New Conga Grid MultiTab Page screen appears.
11. From the **Select Object** drop-down list, choose an object for the first tab in the MultiTab Conga Grid℠.
12. From the **Default View** drop-down list, select an existing Conga Grid view for the selected object. Leave it blank for no view. Or, select **Create new view for this Page** to automatically create a view which you can later redefine when you load the tab.

13. Choose whether or not to enable various properties for this tab. Refer to the pertinent help topics for details on each feature.
14. Click **Add Another Tab**. Repeat steps 11 through 13. Add as many different tabs as you need.
15. Click **Save and Finish**. A Visualforce Page Created screen appears.
16. Click **View VF Page**.

17. Click **Edit**. The Visualforce Markup code should look similar to the following. This example has tabs for Accounts and Contacts.

18. Change the first several lines to the code shown below. Specifically, you will:
    - Add *<apex:pageBlock>* on line 2.
    - Press **ENTER** to make line 3 blank.
    - Remove *<li>Contacts</li>*.
    - Change *<div id="tabstrip">* to *<div class="tabstrip">* on line 4.
    - Press **ENTER** to make line 14 blank.
    - Add *</div>* on line 15.
    - Add </apex:pageBlock> on line 16.
    - Press ENTER to make line 17 blank.
    - Add <apex:pageBlock> on line 18.
    - Copy lines 4 and 5, and paste them into lines 19 and 20.

- Change line 20 to the appropriate object (Contacts, in this example).

19. Change the first several lines to the code shown below. Specifically, you will:
    - Add *</apex:pageBlock>* to line 31.
    - Press **ENTER** to make line 32 blank.

- Change lines 34-36 to the code shown below. Change the pound signs to periods.

20. Click **Save**.

The following example shows a stacked Multi-Tabular related list:

Multi-Tabular Related List (Visualforce Markup)

Follow these steps:

1. In Salesforce®, click Setup.
2. Click Develop and click Visualforce Pages.
3. Click New. A Page Information screen appears.
4. Enter a Label, which will be the name of the tab containing the Visualforce page.
5. Enter a Name that will make this list unique in Salesforce, using underscores instead of spaces.
6. (Optional). Enter a Description.
7. Click the Visualforce Markup tab.
8. Copy and paste the code below into the markup field. See the following examples:
    - Visualforce Multi-Tabular Related List

```
<apex:page sidebar= "false"  showHeader= "true"  doctype= "html-4.
01-strict" > <div id= "tabstrip" > <ul> <li  class = "k-state-
active" >This Month</li> <li>Last Month</li> <li>This Quarter</li>
<li>Next Quarter</li> <li>By Stage</li> <li>My Tasks</li> </ul>
<div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "Opport
unity"  ViewID= "a06o000000Hy0myAAB"  /> <CRMC_PP:DrillUp /> </
div> <div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "O
pportunity"  ViewID= "a06o000000Hy0n3AAB"  /> <CRMC_PP:DrillUp />
</div> <div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "
Opportunity"  ViewID= "a06o000000Hy0nDAAR"  /> <CRMC_PP:DrillUp />
</div> <div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "
Opportunity"  ViewID= "a06o000000Hy0nIAAR"  /> <CRMC_PP:DrillUp />
</div> <div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "
Opportunity"  ViewID= "a06o000000Hy0o0AAB"  /> <CRMC_PP:DrillUp />
</div> <div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "
Task"  ViewID= "a06o000000FCsz2AAD"  /> <CRMC_PP:DrillUp /> </
div></div><script>$(document).ready(function() { $( "#tabstrip" ).
kendoTabStrip({ });});</script></apex:page>
```

- Embedded Multi-Tabular Related List

  This example creates a Multi-Tabular Conga Grid℠ embedded in an Account page layout:

```
<apex:page standardController= "Account"  showHeader= "false" ><di
v id= "tabstrip" > <ul> <li  class = "k-state-active" >Contacts</
li> <li>Cases</li> <li>Opportunities</li> <li>Invoices</li>
<li>Assets</li> <li>Tasks</li> </ul> <div
style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "Contact"
  FKName= "AccountId"  FKValue= "{!Account.Id}"  />
<CRMC_PP:DrillUp /> </div> <div style= "overflow:hidden;" >
<CRMC_PP:Grid objectName= "Case"  FKName= "AccountId"  FKValue= "{
!Account.Id}"  /> <CRMC_PP:DrillUp /> </div> <div style= "overflow
:hidden;" > <CRMC_PP:Grid objectName= "Opportunity"  FKName= "Acco
untId"  FKValue= "{!Account.Id}"  /> <CRMC_PP:DrillUp /> </div>
<div style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "crmc_I
nvoice__c"  FKName= "BillTo_Account__c"  FKValue= "{!Account.Id}"
/> <CRMC_PP:DrillUp /> </div> <div style= "overflow:hidden;" >
<CRMC_PP:Grid objectName= "Asset"  FKName= "AccountId"  FKValue= "
{!Account.Id}"  /> <CRMC_PP:DrillUp /> </div> <div style= "overflo
w:hidden;" > <CRMC_PP:Grid objectName= "Task"  FKName= "AccountId"
  FKValue= "{!Account.Id}"  /> <CRMC_PP:DrillUp /> </div></
```

```
div><script>$(document).ready(function() { $( "#tabstrip" ).kendoT
abStrip({ });});</script></apex:page>
```

9. You can customize these examples as needed by following the instructions in Modify Multi-Tabular Related Lists.
10. Click Save.
11. Add the resulting Visualforce page to the Conga Grid Visualforce Pages permission set. See Security and Permission Sets for Visualforce Pages for instructions.

Modify Multi-Tabular Related Lists

Visualforce Page Markup

After creating a Multi-Tabular Conga Grid℠, you may need to change the objects that appear in the tabs. This involves editing the Visualforce Page Markup. Follow these steps to access the markup code:

1. In Salesforce®, click Setup.
2. Click Develop and click Visualforce Pages.
3. Locate the Multi-Tabular Visualforce page you want to edit, and click the Edit link next to it.
4. In the Visualforce Markup code, look for the list tags: *<li>...</li>*. These define the tab labels. Following the list tags are groups of division tags (*<div>...</div>*). Each division tag group corresponds to a specific tab label in the list tags above:

The division tags define what is underneath each tab. Immediately above the beginning *<div>* tag is a grey-colored HTML comment that specifies the name of the tab.

Notice how each *<CRMC_PP:Grid>* tag has an objectName field. This should be the same for every tab since the Multi-Tabular related list pertains to a specific object (in this case, Opportunity). If the specified object name is incorrect, then the data in the Multi-Tabular list will be incorrect.

If you used the Multi-Tabular (Automatic) option to create a page layout, the object name will automatically populate in the Visualforce Markup code. If you used the Multi-Tabular (Manual) option to create a page layout using your own Visualforce Markup code, be sure to use the singular form of the object name (for example, Opportunity versus Opportunities). This pertains to standard Salesforce objects. If the object you are creating a tab for is a custom object, then you need to find the API name for that object. To do this, go to Salesforce Setup, click Create and click Objects. Find the object in this list and click the link next to it. Locate the field named object Name. Type that name exactly after the CRMC_PP:Grid ObjectName text.

Edit Tab Names

To edit a tab name, update both the list tag and its corresponding HTML comment (between the <!--- and ---> characters). For example:

Delete Tabs

To delete a tab, remove the associated list item (*<li>…</li>*) containing the tag name. Then delete the entire division block (*<div>…</div>*) associated with that list. The following figure shows an example:

Add New Tabs

Each tab in the Multi-Tabular Visualforce page is based on a specific view. You can add more tabs (views) by following these steps:

Get the View ID

1. Click Conga Grid Explorer.
2. Select the object of interest (for example, Opportunity).
3. Click the Manage Views link. The Manage Views screen appears.
4. Select the view that will be associated with the new tab.
5. Click Actions and click Export To > Excel (with Record IDs).
6. Open the exported record in a spreadsheet application, and note the Id value of the selected record. You will need this in a later step.
7. Close the Manage Views screen.

Edit the Visualforce Markup

1. Go to the Visualforce Markup code for the selected Multi-Tabular Visualforce page. Refer to steps 1 through 3 at the beginning of this topic.
2. Decide where the tab should be located relative to other tabs.

3. Insert a new list item (<li>…</li>) in the group of list tags in the appropriate order. This example shows adding a new tab labeled This Quarter as the third tab:

4. Select any division block of text beginning with the <!--- comment characters and ending with the </div> tag. Copy this to your system clipboard (for example, Ctrl-C).

5. Paste the block of code among the division tags in the order where you want the tab to appear. For example, we pasted the code from Step 3 after the second <div>…</

div> group since the new tab will be third in the order.

6. Enter the new tab name in the grey HTML comment tag between the <!--- and ---> characters.

7. Replace the ViewID value (between the double quotes) with the Id value that you recorded earlier (see Get the View ID).

8. Click Save at the top of the Visualforce page editor.

To see the result, click Preview (in the same row as Save).

Conga Grid Actions

Learn how to use actions to perform essential tasks in Conga Grid℠. You can invoke actions by clicking the Actions button or by right-clicking on selected records.

- Batch Add in Conga Grid

- Clone
- Conga Composer Solutions - Single Record
- Conga Batch Solutions - Multiple Rows
- Custom Actions
- Export To
- Fill-Down
- Mass Update
- Send Email
- Summarize Columns
- View Records in a New Window
- View Records Using List Browse

Batch Add in Conga Grid

Use the Batch Add action to add Tasks, Notes, Chatter Messages, and other items to selected records. This is a convenient way to update multiple records with the same information.

Here are some examples:

- Add a Task to selected Leads.
- Add multiple Users to a permission set.
- Assign licenses to multiple Users.
- Add Follow Records to keep track of selected records in order to receive Chatter Notifications about updates about them.

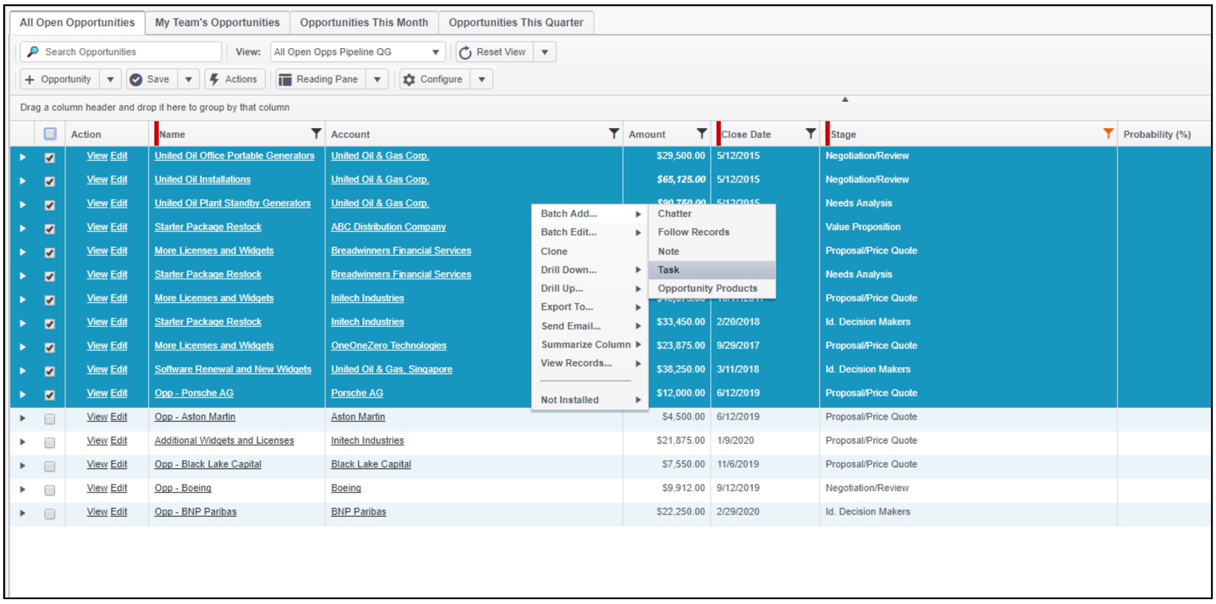
The options in the Batch Add menu vary, depending on the object that you are currently viewing.

You can also right-click on the grid to access the Action menu.

To Batch Add a Task to Opportunity records:

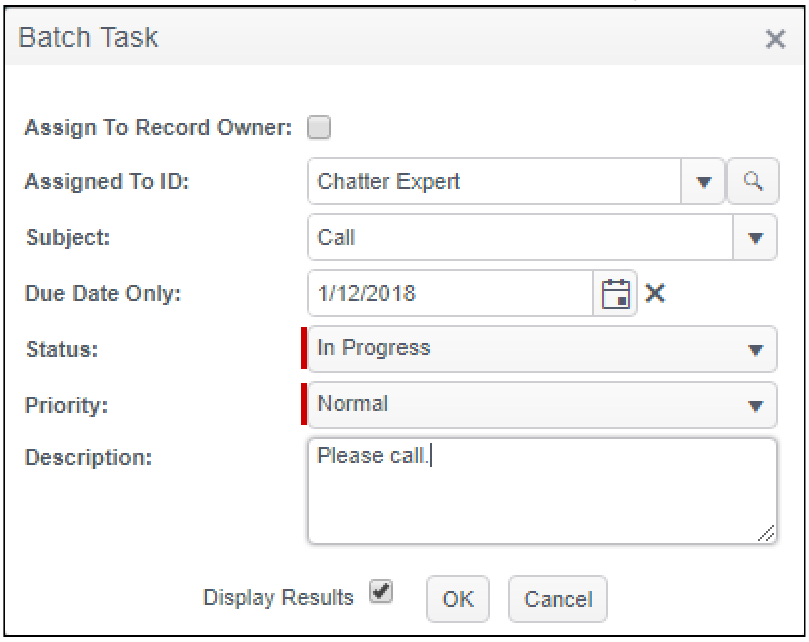
1. Multi-select the records to which you want to add a Task.

2. Right-click on the grid; hover over Batch Add and select **Task**.



	□	Action	Name	Account	Amount	Close Date	Stage	Probability (%)
▸	☑	View Edit	United Oil Office Portable Generators	United Oil & Gas Corp.	$29,500.00	5/12/2015	Negotiation/Review	
▸	☑	View Edit	United Oil Installations	United Oil & Gas Corp.	*$65,125.00*	5/12/2015	Negotiation/Review	
▸	☑	View Edit	United Oil Plant Standby Generators	United Oil & Gas Corp.	$90,750.00	5/12/2015	Needs Analysis	
▸	☑	View Edit	Starter Package Restock	ABC Distribution Company			Value Proposition	
▸	☑	View Edit	More Licenses and Widgets	Breadwinners Financial Services			Proposal/Price Quote	
▸	☑	View Edit	Starter Package Restock	Breadwinners Financial Services			Needs Analysis	
▸	☑	View Edit	More Licenses and Widgets	Initech Industries			Proposal/Price Quote	
▸	☑	View Edit	Starter Package Restock	Initech Industries	$33,450.00	2/20/2018	Id. Decision Makers	
▸	☑	View Edit	More Licenses and Widgets	OneOneZero Technologies	$23,875.00	9/29/2017	Proposal/Price Quote	
▸	☑	View Edit	Software Renewal and New Widgets	United Oil & Gas, Singapore	$38,250.00	3/11/2018	Id. Decision Makers	
▸	☑	View Edit	Opp - Porsche AG	Porsche AG	$12,000.00	6/12/2019	Proposal/Price Quote	
▸	□	View Edit	Opp - Aston Martin	Aston Martin	$4,500.00	6/12/2019	Proposal/Price Quote	
▸	□	View Edit	Additional Widgets and Licenses	Initech Industries	$21,875.00	1/9/2020	Proposal/Price Quote	
▸	□	View Edit	Opp - Black Lake Capital	Black Lake Capital	$7,550.00	11/6/2019	Proposal/Price Quote	
▸	□	View Edit	Opp - Boeing	Boeing	$9,912.00	9/12/2019	Negotiation/Review	
▸	□	View Edit	Opp - BNP Paribas	BNP Paribas	$22,250.00	2/29/2020	Id. Decision Makers	

3. Enter the Task details and the user to whom you want to assign the Tasks.



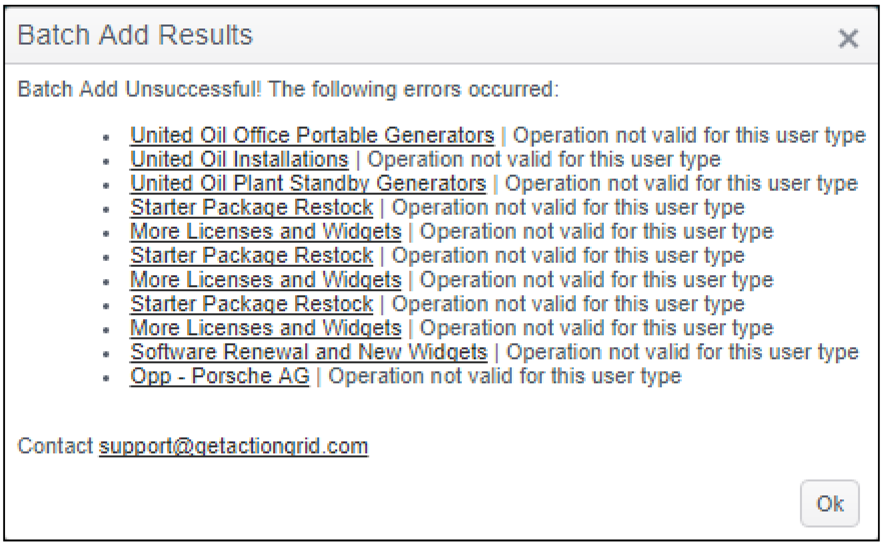
Batch Task

Assign To Record Owner:	□
Assigned To ID:	Chatter Expert
Subject:	Call
Due Date Only:	1/12/2018
Status:	In Progress
Priority:	Normal
Description:	Please call.

Display Results ☑ OK Cancel

4. Click **OK**.

5. Results of the Batch Add will appear. Click **OK**.



Batch Add Results

Batch Add Unsuccessful! The following errors occurred:

- United Oil Office Portable Generators | Operation not valid for this user type
- United Oil Installations | Operation not valid for this user type
- United Oil Plant Standby Generators | Operation not valid for this user type
- Starter Package Restock | Operation not valid for this user type
- More Licenses and Widgets | Operation not valid for this user type
- Starter Package Restock | Operation not valid for this user type
- More Licenses and Widgets | Operation not valid for this user type
- Starter Package Restock | Operation not valid for this user type
- More Licenses and Widgets | Operation not valid for this user type
- Software Renewal and New Widgets | Operation not valid for this user type
- Opp - Porsche AG | Operation not valid for this user type

Contact support@getactiongrid.com

Clone

Follow these steps:

1. Select a record that you want to clone.
2. Right-click on the record and click **Clone**. The Clone Options screen appears.
3. Select the desired **Number of clones**.
4. Click Create **Clones**.

Conga Composer Solutions - Single Record

Conga Grid detects any current Conga button code that is currently configured in your system. That is, Conga Grid will automatically detect if you have a list view or detail view button set to run a Composer template. Follow these steps to enable this action:

1. Click **Conga Grid Setup**.
2. In the object list, select the object for which the Conga solution is set up.
3. Click the **object Features** tab.
4. Click **Conga Action**. The Conga Action screen appears.
5. Enter a **Conga Action Label**. This will be the name of the action in the Conga Grid menu.
6. From the **Type of Conga Action** drop-down list, click **Conga Composer**.
7. Click the **Select Button/Formula Field** drop-down list and select the button name that will launch the Conga solution from the Conga Grid menu.
8. Click **OK**.
9. Set the desired feature security.
10. Click **Conga Grid Explorer**.

11. In the object list, select the object for which the Conga solution is set up.
12. Select a single row.
13. Right-click and select **Conga > action**. This will invoke Conga Composer.

Conga Batch Solutions - Multiple Rows

Before you can launch Conga Batch, you must first set up a Salesforce formula field. See the article *Convert a Conga Composer Button to a Formula Field Using the Conga Formula Builder* in the *Conga Batch* guide for instructions.

If your Conga Batch solution uses Salesforce Reports to gather additional data, you must modify the solution to use a SOQL query to gather the additional data when using Conga Grid. This is due to session limitations within Salesforce. Contact your Conga support representative to help with this conversion if needed.

> ⚠ Users can merge a maximum of 75 records when launching Conga Batch from Conga Grid.

Follow the steps below to configure Conga Batch for use with Conga Grid:

1. Go to the **Conga Grid Setup** tab in Salesforce
2. In the object list to the left of the screen, select the master object for which the Conga Batch solution is set up.
3. Click the **object Features** tab, and then click **Conga Actions.**
4. In the Conga Action window, enter a **Conga Action Label**. This will be the name of the action in the Conga Grid menu.
5. From the **Type of Conga Actions** drop-down list, click **Conga Batch (Conga Conductor in Grid version 1.102 and lower)**.
6. Click the **Select Buttons or Formula Fields** drop-down list and select the formula field name that will launch the Conga solution from the Conga Grid menu.
7. Click **Save**.
8. Set the desired feature security.
9. Go to the **Conga Grid Explorer** tab in Salesforce.
10. In the object list, select the master object for which the Conga Batch solution is set up.
11. Select multiple rows.
12. Right-click and select **Conga > action**. This will invoke your Conga Batch solution for the selected records.

Custom Actions

Warning

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. Please create Custom Actions at your own risk.

See the following sections for examples:

- Create Custom Actions
- Delete Custom Actions
- Rename Custom Actions

Important

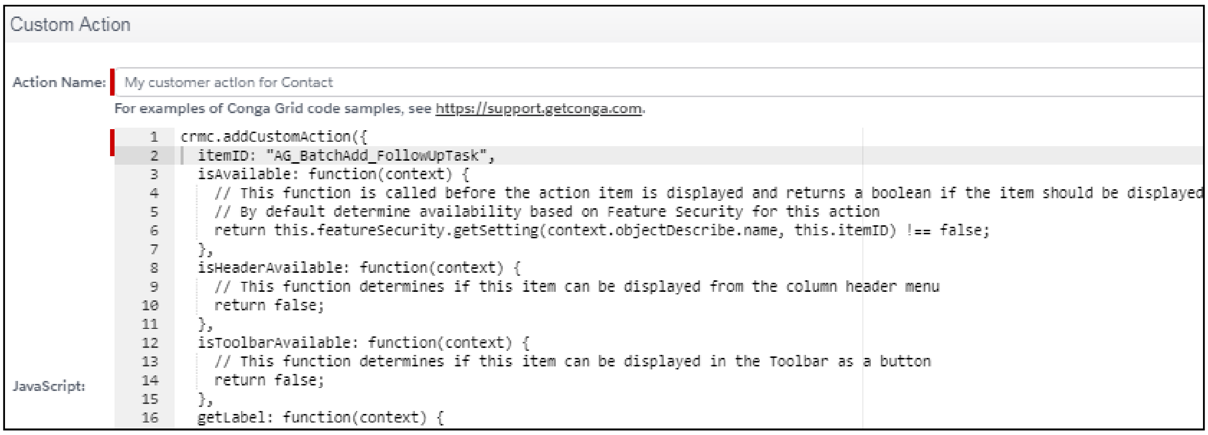
Ensure that the Enable Content Sniffing Protection setting is disabled in Salesforce Setup. If this setting is enabled, it can prompt an error when accessing custom action.
To disable this setting, navigate to **Setup - > Security Controls - > Session Settings** and then **uncheck the Enable Content Sniffing protection checkbox** under the Content Sniffing protection section.

Create Custom Actions

Follow these steps to create your own actions using JavaScript.

1. Click **Conga Grid Setup**.
2. Select the object for which you want to create a custom action.
3. Click the **Object Features** tab.
4. Click **New Custom Action**.
5. Enter an **Action Name**.

6. Enter the Javascript code to perform custom action. See Code Examples for Custom Actions.



Custom Action

Action Name: My customer action for Contact

For examples of Conga Grid code samples, see https://support.getconga.com.

JavaScript:

```
crmc.addCustomAction({
  itemID: "AG_BatchAdd_FollowUpTask",
  isAvailable: function(context) {
    // This function is called before the action item is displayed and returns a boolean if the item should be displayed
    // By default determine availability based on Feature Security for this action
    return this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false;
  },
  isHeaderAvailable: function(context) {
    // This function determines if this item can be displayed from the column header menu
    return false;
  },
  isToolbarAvailable: function(context) {
    // This function determines if this item can be displayed in the Toolbar as a button
    return false;
  },
  getLabel: function(context) {
```

7. Click **OK**.
   The custom action is located on the Actions menu.

You can also create a button in the Conga Grid toolbar for your custom action:

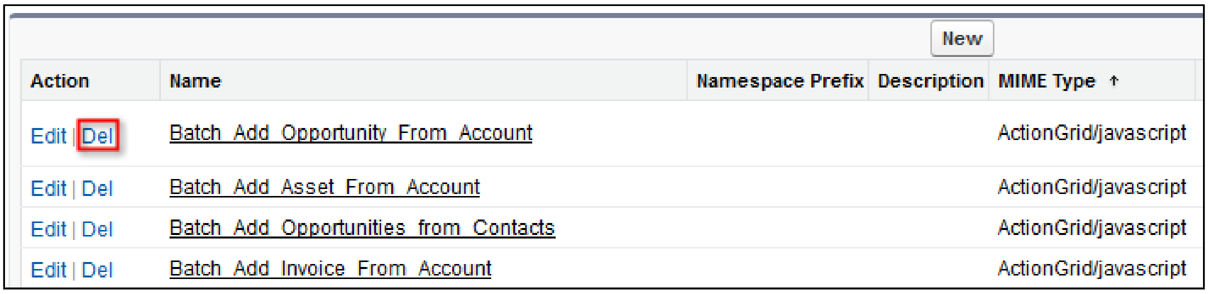
1. In the **JavaScript** field of the Custom Action screen, look for the *IsToolbarAvailable* function.
2. Change the *return* value to **true**.
3. Click **OK**.
4. Click **Conga Grid Explorer**.
5. Click the **Configuration** button to open the Configure menu on the right side of the Conga Grid.
6. Click **Toolbar**.
7. Enable the checkbox for your custom action. A button to invoke the custom action is located on the toolbar.

Delete Custom Actions

Conga Grid custom actions are stored as Salesforce static resources. You can delete custom actions from the Salesforce Static Resources page:

1. In Salesforce, click **Setup**.
2. Click Develop and click **Static Resources**.

3. Click the **Del** hyperlink next to the resource name. For example:

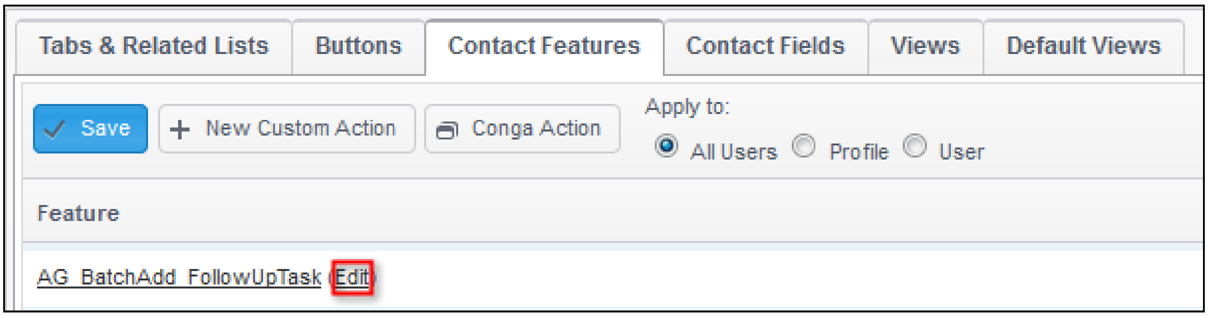


Action	Name	Namespace Prefix	Description	MIME Type ↑
Edit | Del	Batch_Add_Opportunity_From_Account			ActionGrid/javascript
Edit | Del	Batch_Add_Asset_From_Account			ActionGrid/javascript
Edit | Del	Batch_Add_Opportunities_from_Contacts			ActionGrid/javascript
Edit | Del	Batch_Add_Invoice_From_Account			ActionGrid/javascript

Rename Custom Actions

To rename a custom action:

1. Click **Conga Grid Setup**.
2. Select the object associated with the custom action.
3. Click the **object Features** tab.
4. Click the **Edit** link next to the name of the custom action. For example:



Apply to: ● All Users ○ Profile ○ User

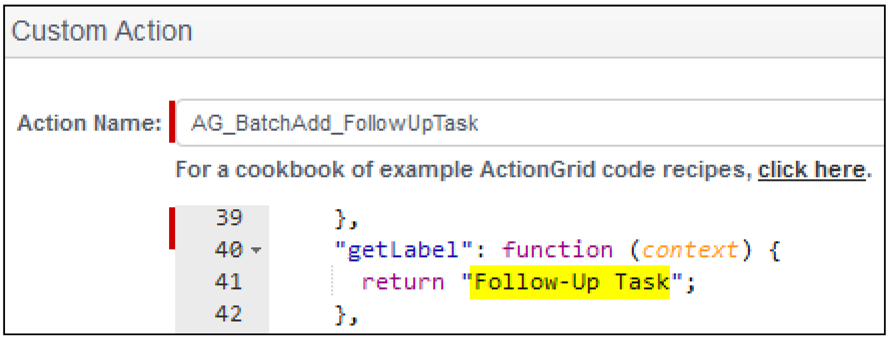
Feature

AG_BatchAdd_FollowUpTask (Edit)

A custom action name has two components:

- Action Name: The name of the action as it appears under the Actions tab of Conga Grid Setup. The figure in Step 4 shows an example.
- Label: The name of the action as it appears in the Actions menu in the Conga Grid Explorer.

5. To change the Label, find the *getLabel* line of code in the Custom Action screen. Then modify the text in the *return* statement. For example:



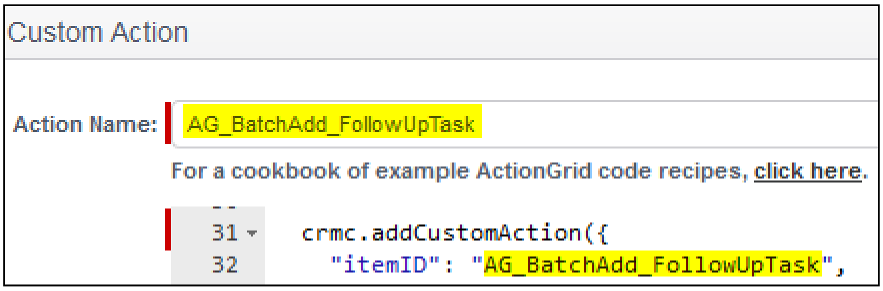
Custom Action

Action Name: AG_BatchAdd_FollowUpTask

For a cookbook of example ActionGrid code recipes, click here.

```
    },
    "getLabel": function (context) {
      return "Follow-Up Task";
    },
```

6. To change the Action Name, enter the same name in the **Action Name** field and in the itemID line of code. The names must exactly match.



Custom Action

Action Name: AG_BatchAdd_FollowUpTask

For a cookbook of example ActionGrid code recipes, click here.

```
crmc.addCustomAction({
  "itemID": "AG_BatchAdd_FollowUpTask",
```

7. Click **OK**.

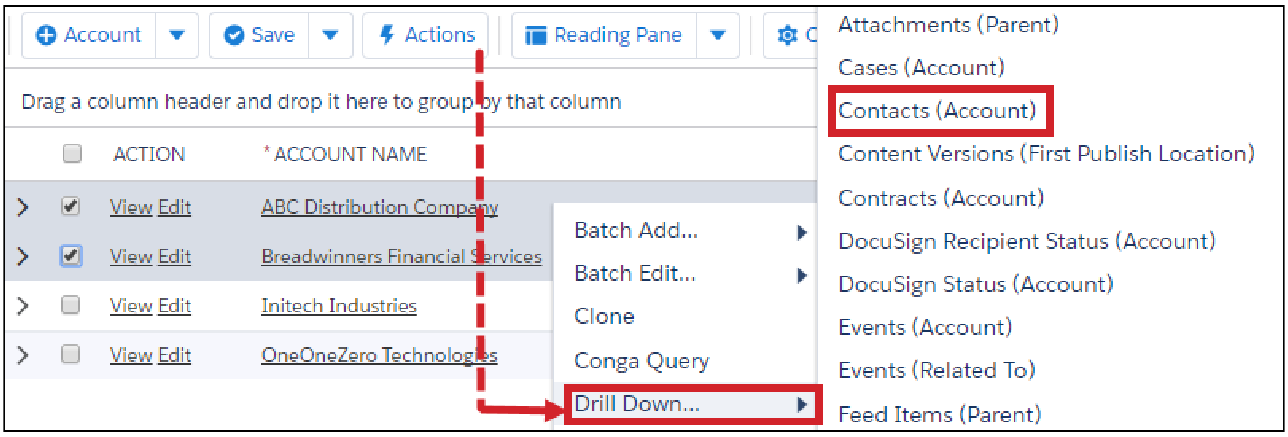
> ⚠ The itemID must have the same value throughout the custom action's Javascript code.

Drill Down/Up

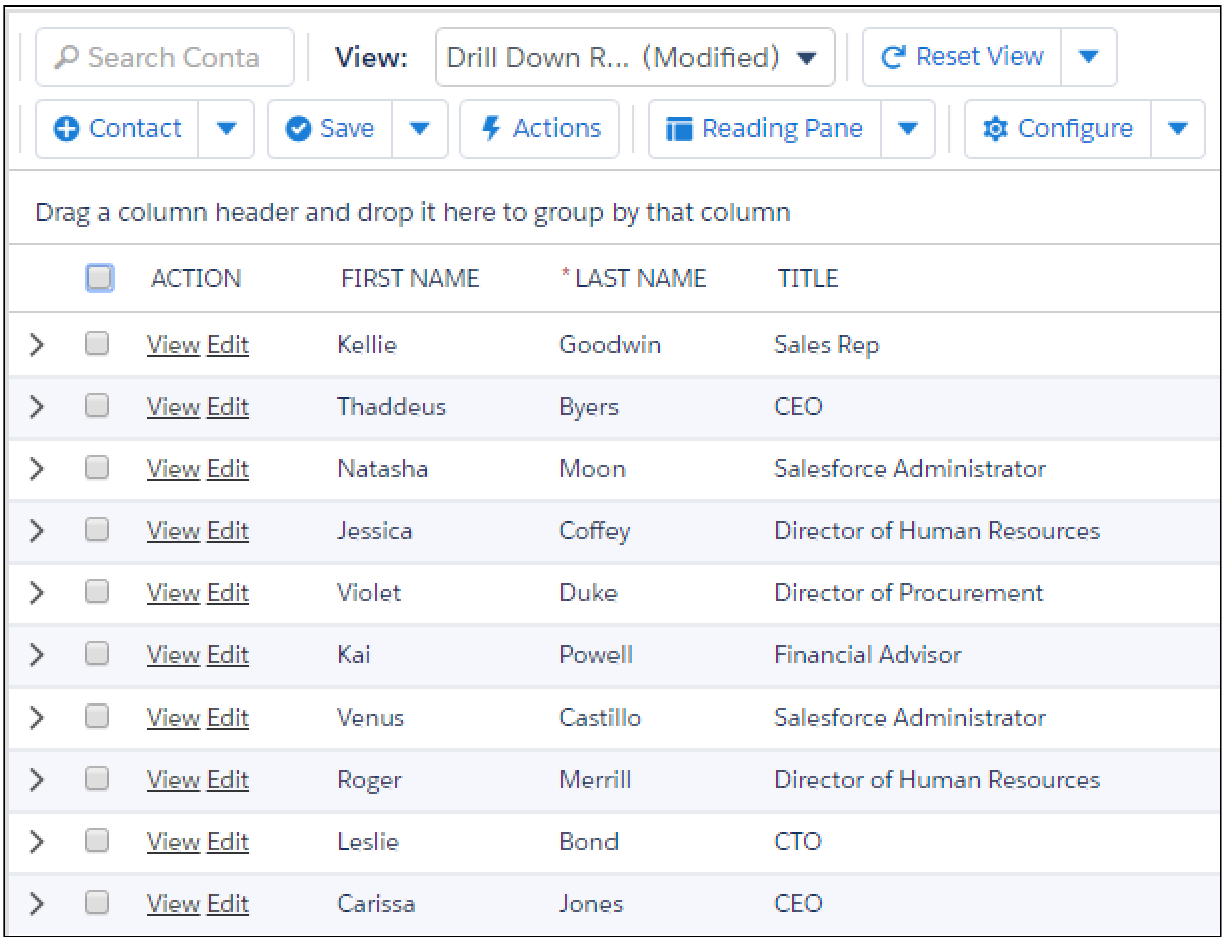
Use the Drill Up action to access parent data, or use the Drill Down action to access child data.

For example, if you select two Account records and each one has five Contacts, Drill Down will display a grid of the total ten Contacts between the two accounts.

> **Tip**
>
> The Default View determines the view seen after Drilling Down or Up. To set the Default View through the Configure drop down, select Manage Views, then check a view as Default.



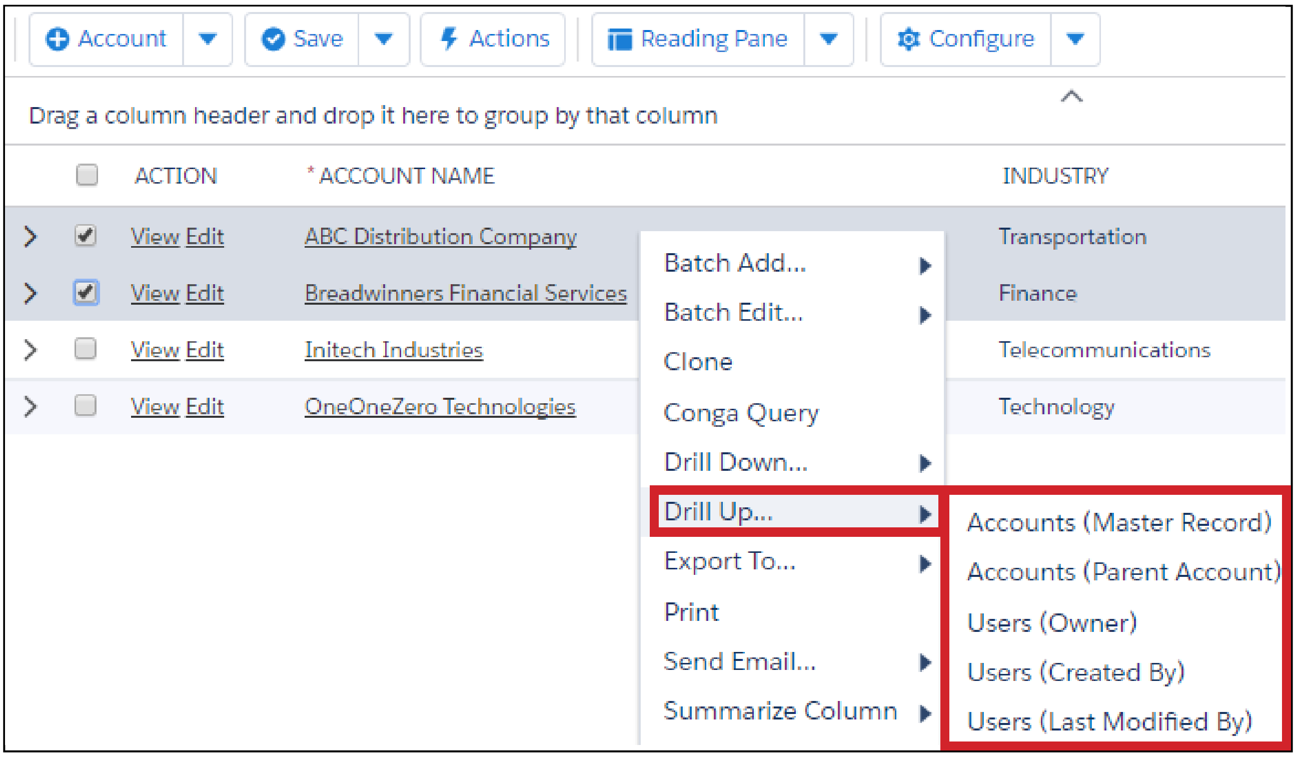
Result:



Drag a column header and drop it here to group by that column

	□	ACTION	FIRST NAME	* LAST NAME	TITLE
>	□	View Edit	Kellie	Goodwin	Sales Rep
>	□	View Edit	Thaddeus	Byers	CEO
>	□	View Edit	Natasha	Moon	Salesforce Administrator
>	□	View Edit	Jessica	Coffey	Director of Human Resources
>	□	View Edit	Violet	Duke	Director of Procurement
>	□	View Edit	Kai	Powell	Financial Advisor
>	□	View Edit	Venus	Castillo	Salesforce Administrator
>	□	View Edit	Roger	Merrill	Director of Human Resources
>	□	View Edit	Leslie	Bond	CTO
>	□	View Edit	Carissa	Jones	CEO

Here is an example of the parent fields that are available by selecting Drill Up from an Account record:



	☐	ACTION	* ACCOUNT NAME	INDUSTRY
>	☑	View Edit	ABC Distribution Company	Transportation
>	☑	View Edit	Breadwinners Financial Services	Finance
>	☐	View Edit	Initech Industries	Telecommunications
>	☐	View Edit	OneOneZero Technologies	Technology

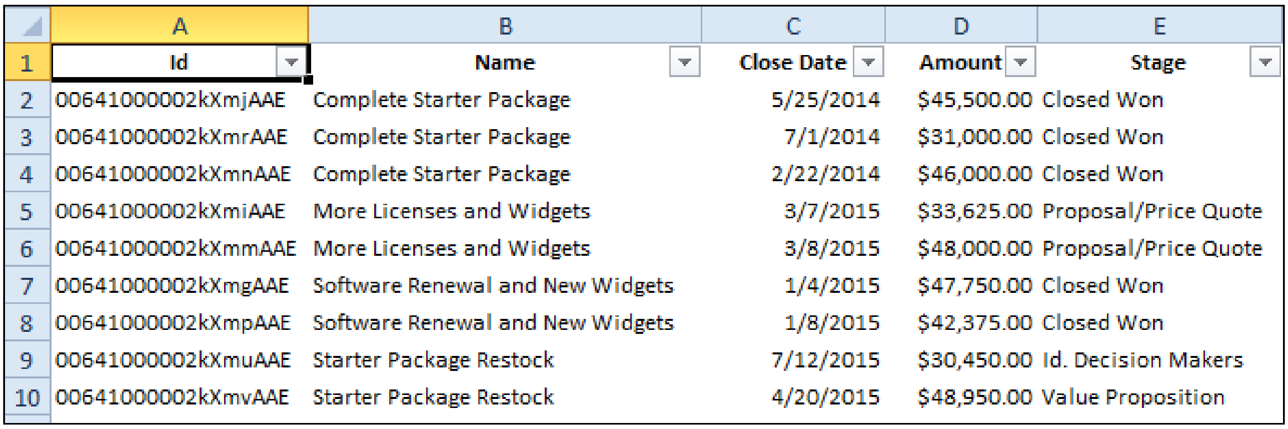
Export To

Use the Export To action to export selected records to a CSV or Excel® file.

Follow these steps:

1. Select one or more records that you want to export.
2. Click **Actions**.
3. Click **Export To** and select an option:
   - **CSV**
   - **CSV (with Record IDs)**
   - **Excel**
   - **Excel (with Record IDs)**

The following example shows Opportunity records that were filtered by Amount values over $30,000. These records were exported to Excel with record IDs:



	Id	Name	Close Date	Amount	Stage
2	00641000002kXmjAAE	Complete Starter Package	5/25/2014	$45,500.00	Closed Won
3	00641000002kXmrAAE	Complete Starter Package	7/1/2014	$31,000.00	Closed Won
4	00641000002kXmnAAE	Complete Starter Package	2/22/2014	$46,000.00	Closed Won
5	00641000002kXmiAAE	More Licenses and Widgets	3/7/2015	$33,625.00	Proposal/Price Quote
6	00641000002kXmmAAE	More Licenses and Widgets	3/8/2015	$48,000.00	Proposal/Price Quote
7	00641000002kXmgAAE	Software Renewal and New Widgets	1/4/2015	$47,750.00	Closed Won
8	00641000002kXmpAAE	Software Renewal and New Widgets	1/8/2015	$42,375.00	Closed Won
9	00641000002kXmuAAE	Starter Package Restock	7/12/2015	$30,450.00	Id. Decision Makers
10	00641000002kXmvAAE	Starter Package Restock	4/20/2015	$48,950.00	Value Proposition

Tip

If multi-currency is enabled, exporting currency fields to Excel will only output the number value that matches the record's currency. This can result in a mixed batch of values. For example, if you have three records that are in GBP and three that are in USD, a mixed set of GDP and USD values will be exported to excel. To avoid confusion, it is recommended to export with the object's Currency ISO Code field on the grid.

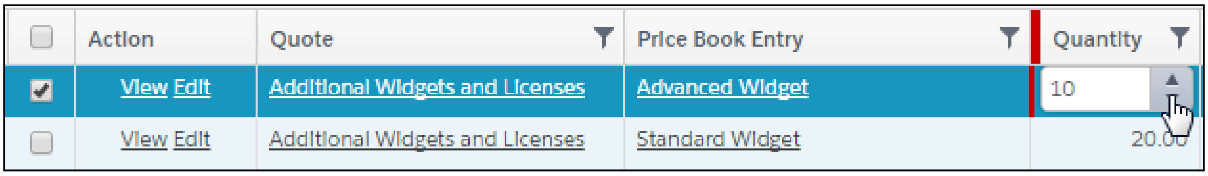
Warning

In Salesforce Lightning, you must pop out the window an perform the export process in the new window.

Fill-Down

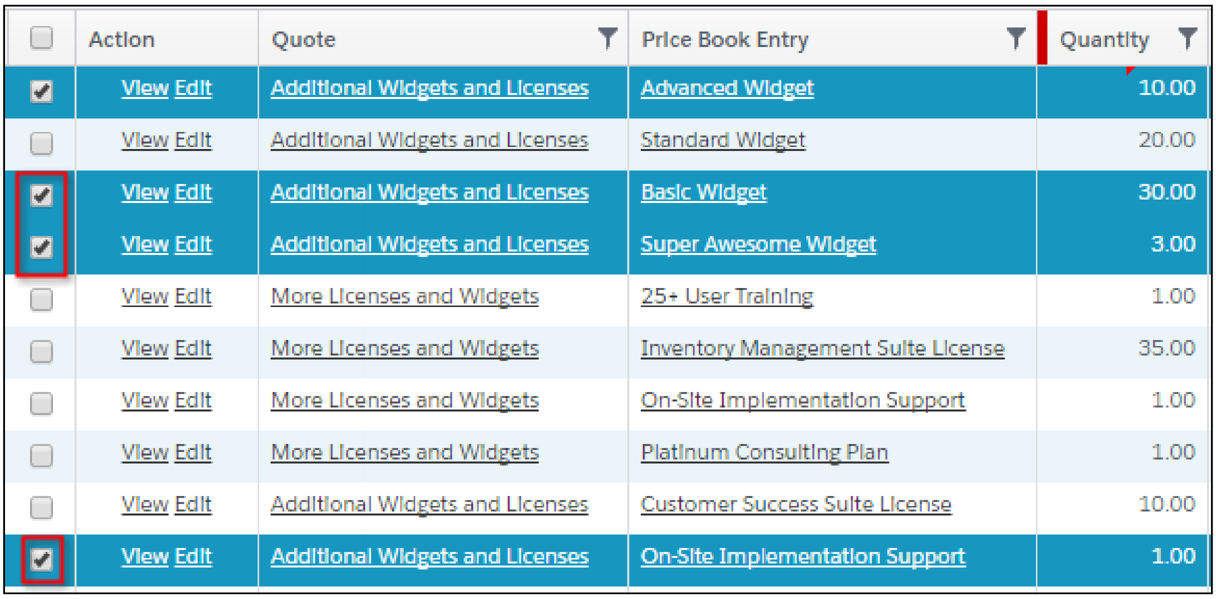
Use the Fill-Down action to update multiple records simultaneously, without having to use the Mass Update action. Fill-Down is the preferred method for modifying the data of a single column quickly. Follow these steps:

1. Edit a single record. Be sure to leave this record selected. The following figure shows an example of editing the Quantity field:



□	Action	Quote	Price Book Entry	Quantity
☑	View Edit	Additional Widgets and Licenses	Advanced Widget	10
□	View Edit	Additional Widgets and Licenses	Standard Widget	20.00

2. Select all records in the grid that you would like to change to the first record's value; for example:



☐	Action	Quote	Price Book Entry	Quantity
☑	View Edit	Additional Widgets and Licenses	Advanced Widget	10.00
☐	View Edit	Additional Widgets and Licenses	Standard Widget	20.00
☑	View Edit	Additional Widgets and Licenses	Basic Widget	30.00
☑	View Edit	Additional Widgets and Licenses	Super Awesome Widget	3.00
☐	View Edit	More Licenses and Widgets	25+ User Training	1.00
☐	View Edit	More Licenses and Widgets	Inventory Management Suite License	35.00
☐	View Edit	More Licenses and Widgets	On-Site Implementation Support	1.00
☐	View Edit	More Licenses and Widgets	Platinum Consulting Plan	1.00
☐	View Edit	Additional Widgets and Licenses	Customer Success Suite License	10.00
☑	View Edit	Additional Widgets and Licenses	On-Site Implementation Support	1.00

3. Click Actions and click **Batch Edit** > **Fill-Down**.
4. Click **Save**.

The Fill-Down option is only available in the Actions > Edit menu under the following conditions:

- You are in Edit mode.
- You have edited a record.
- You have selected records below the first edit.

Fill-Down only applies to edits made on the first selected row. If you edit a cell in a second row and apply Fill-Down, the edits from the second row will not be filled down.

Find/Replace

Use the Find/Replace action to replace values in text fields. You can also use this action to add new text to empty fields. Find/Replace only pertains to short and long text fields, not to lookup fields such as dates, email addresses, numeric fields, or formula fields. It only applies to the selected records on the current page. Follow these steps:

1. Select one or more records.
2. Click **Actions**.
3. Click **Batch Edit** > **Find/Replace**.
4. Select an available field from the drop-down menu.
5. Enter the **Old Value** (the current data in the record).
6. Enter the **New Value** (the replacement data).

7. Click **OK**. When the Find/Replace action is complete, click **Save** to save your edits.

Mass Update

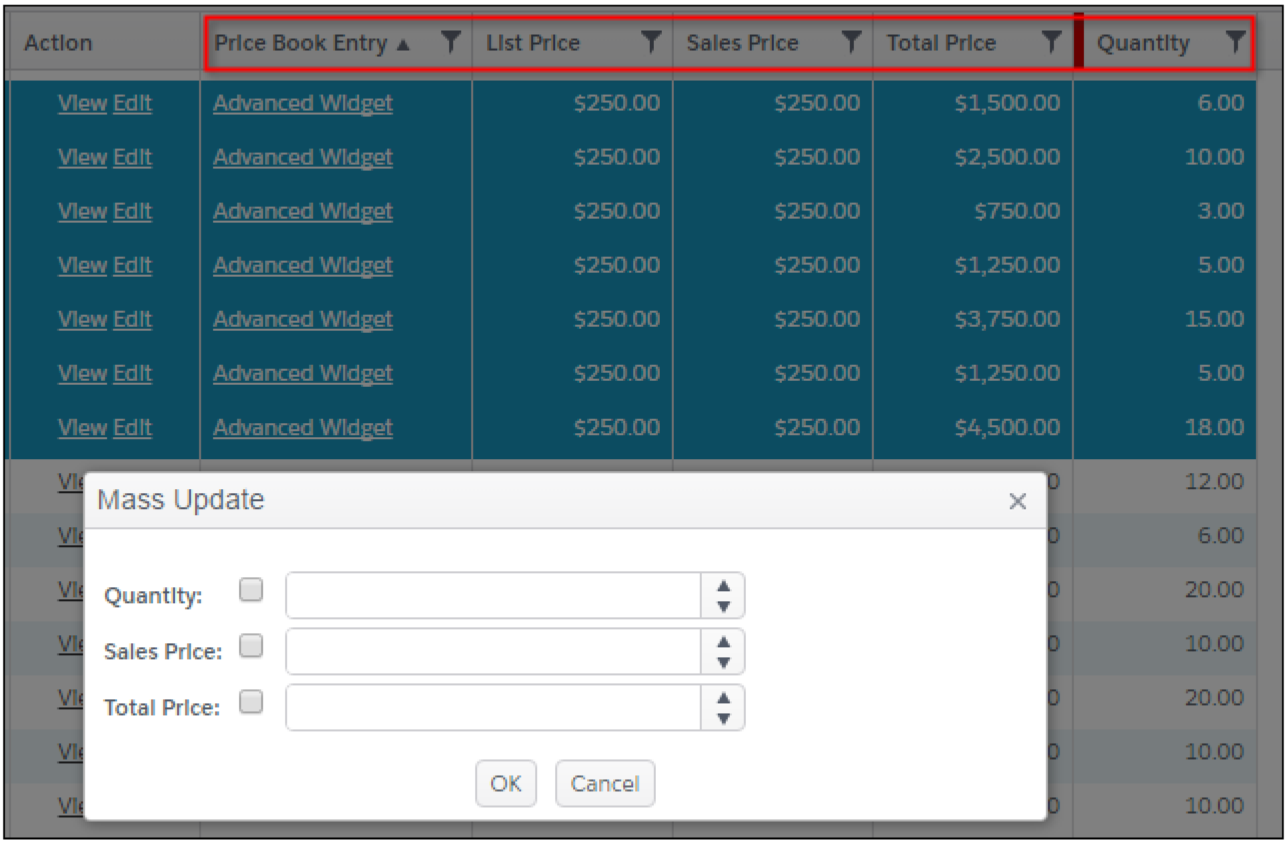
Use the Mass Update action to update several records simultaneously. Mass Update applies to all fields displayed in the grid, unless you specify otherwise in Conga Grid Setup. You can update more fields by adding them from the Field Chooser.

Mass Update is disabled for Filtered Lookups. This prevents incorrect values being assigned to records and is designed to keep your data clean. If you would like to use Mass Update with a lookup, filters must be disabled.

By default, Mass Update is restricted to the first page of records. This ensures that you do not accidentally update large numbers of records at once. You can change this setting in Feature Security using the Mass Update (Single Page) and Mass Update (Large Batch) features.

Follow these steps to use Mass Update:

1. Select the records to update.
2. Click **Actions**.
3. Click **Batch Edit > Mass Update**.
4. In the Mass Update screen, click the check boxes of the fields to update, then enter a new value for these fields.
5. Click **OK**.



Action	Price Book Entry ▲	List Price	Sales Price	Total Price	Quantity
View Edit	Advanced Widget	$250.00	$250.00	$1,500.00	6.00
View Edit	Advanced Widget	$250.00	$250.00	$2,500.00	10.00
View Edit	Advanced Widget	$250.00	$250.00	$750.00	3.00
View Edit	Advanced Widget	$250.00	$250.00	$1,250.00	5.00
View Edit	Advanced Widget	$250.00	$250.00	$3,750.00	15.00
View Edit	Advanced Widget	$250.00	$250.00	$1,250.00	5.00
View Edit	Advanced Widget	$250.00	$250.00	$4,500.00	18.00
Vi				0	12.00
Vi				0	6.00
Vi				0	20.00
Vi				0	10.00
Vi				0	20.00
Vi				0	10.00
Vi				0	10.00

Mass Update

Quantity: □

Sales Price: □

Total Price: □

Administrators can decide which fields to mass update by using Conga Grid Setup. Select the object, followed by the **object Fields** tab. In the Mass Update column, clear the check boxes of fields that should not be updated. Boxes with a line through them indicate they will adhere to default security in Salesforce, while empty check boxes indicate that they cannot be mass updated. Checked boxes indicate fields that can be mass updated.

Watch a video here detailing the Mass Update action.

Send Email

Use the Send Email action to send emails directly from Conga Grid.

1. Select one or more records that contain an email address.
2. Right-click on the selected records and choose one of the following options:
   - **From Template:** A new Salesforce window opens where you can send an email, select an email template, and attach a file to the email.
   - **Compose (Default)**: Your default email program opens. Enter message and click **Send**.
   - **Compose**: The Salesforce New Email window opens Enter message and click **Send**.

- **Batch**: In the Batch Send Email prompt, select a template, name it, and click **OK**. Batch Email utilizes Salesforce Mass email and only works with Leads, Contacts, and Users.
    - See the Apex Developer Guide for what can and cannot be batch emailed with Salesforce.

You can send as many emails as your Salesforce API limit allows.

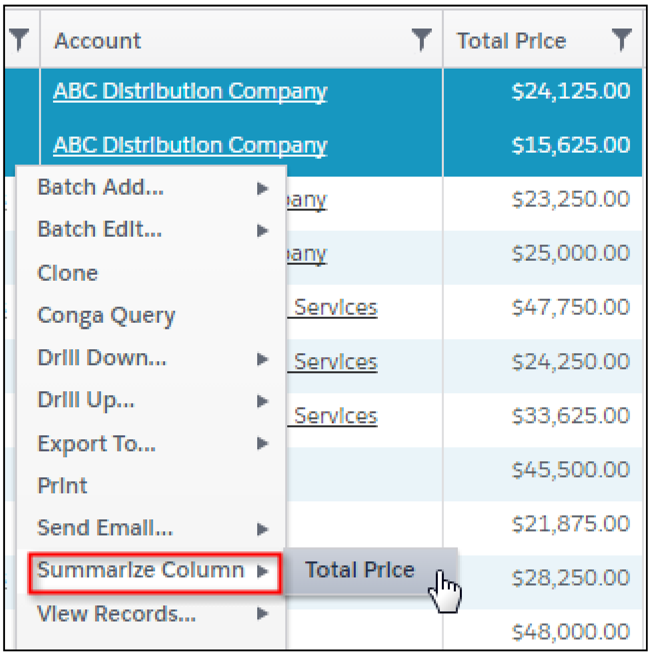
Summarize Columns

Use the Summarize Columns action to summarize rows that contain columns of numeric data types. This action respects any filters that you have set up.

In addition to summarizing, you can also show the average, minimum, or maximum. Sum is the current default.

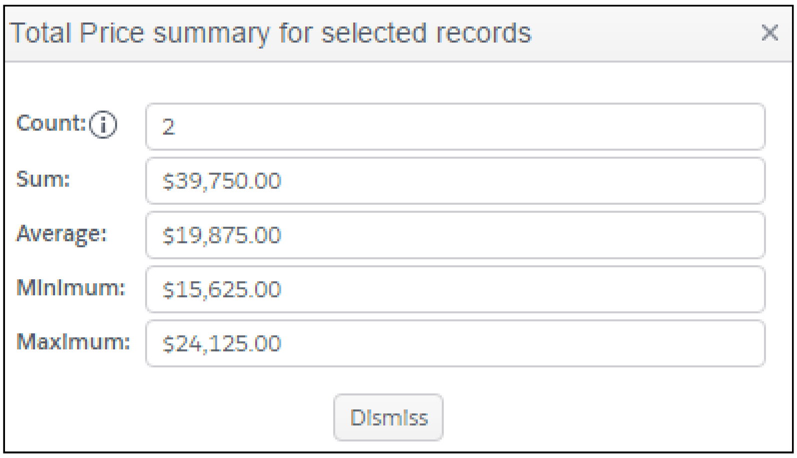
Follow these steps:

1. Select multiple records or all records.
2. Click **Actions**.
3. Click **Summarize Column**, followed by the field that you want to summarize.



Account	Total Price
ABC Distribution Company	$24,125.00
ABC Distribution Company	$15,625.00
any	$23,250.00
any	$25,000.00
Services	$47,750.00
Services	$24,250.00
Services	$33,625.00
	$45,500.00
	$21,875.00
	$28,250.00
	$48,000.00

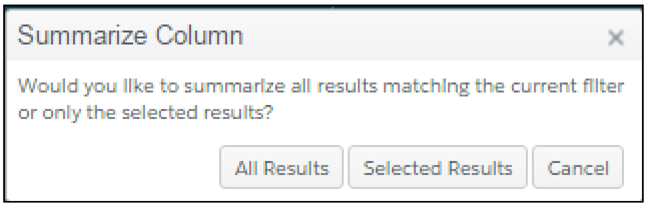
A window appears with the results of the summary:



Total Price summary for selected records

Count:	2
Sum:	$39,750.00
Average:	$19,875.00
Minimum:	$15,625.00
Maximum:	$24,125.00

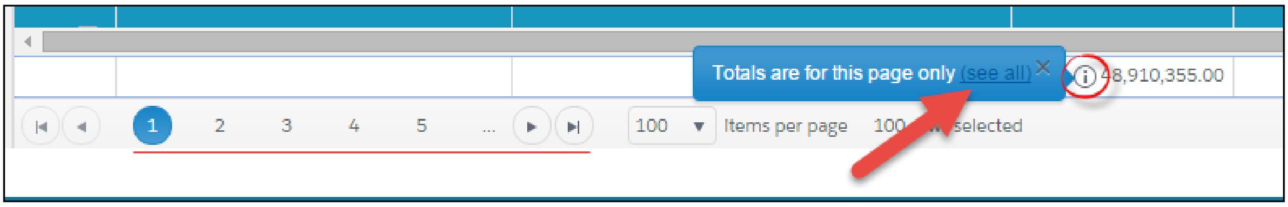
If you select all records and your data set spans the page or even the 10,000-row limit, the following prompt appears. The summation will operate on up to 50 million rows if that's how big the data set is.



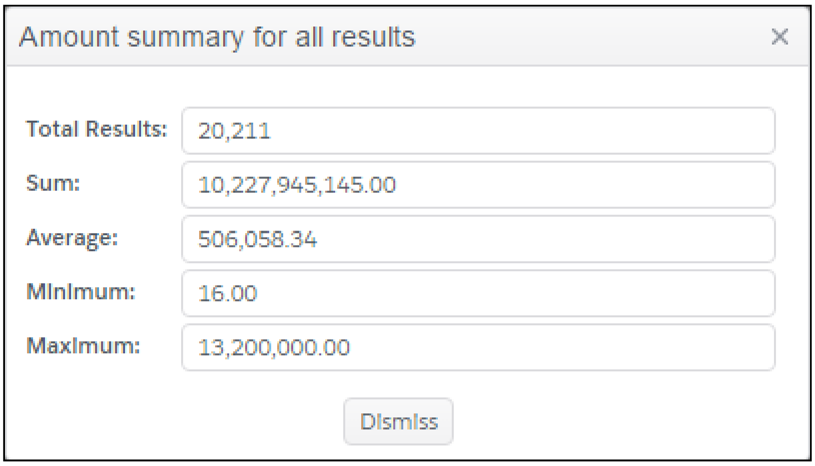
Summarize Column

Would you like to summarize all results matching the current filter or only the selected results?

Alternatively, if the view does not display all records due to page size, you can click on the Information bubble at the bottom of the Conga Grid. Click the **See All** link to view a popup with the summarized totals of that field.



Totals are for this page only (see all)



Amount summary for all results

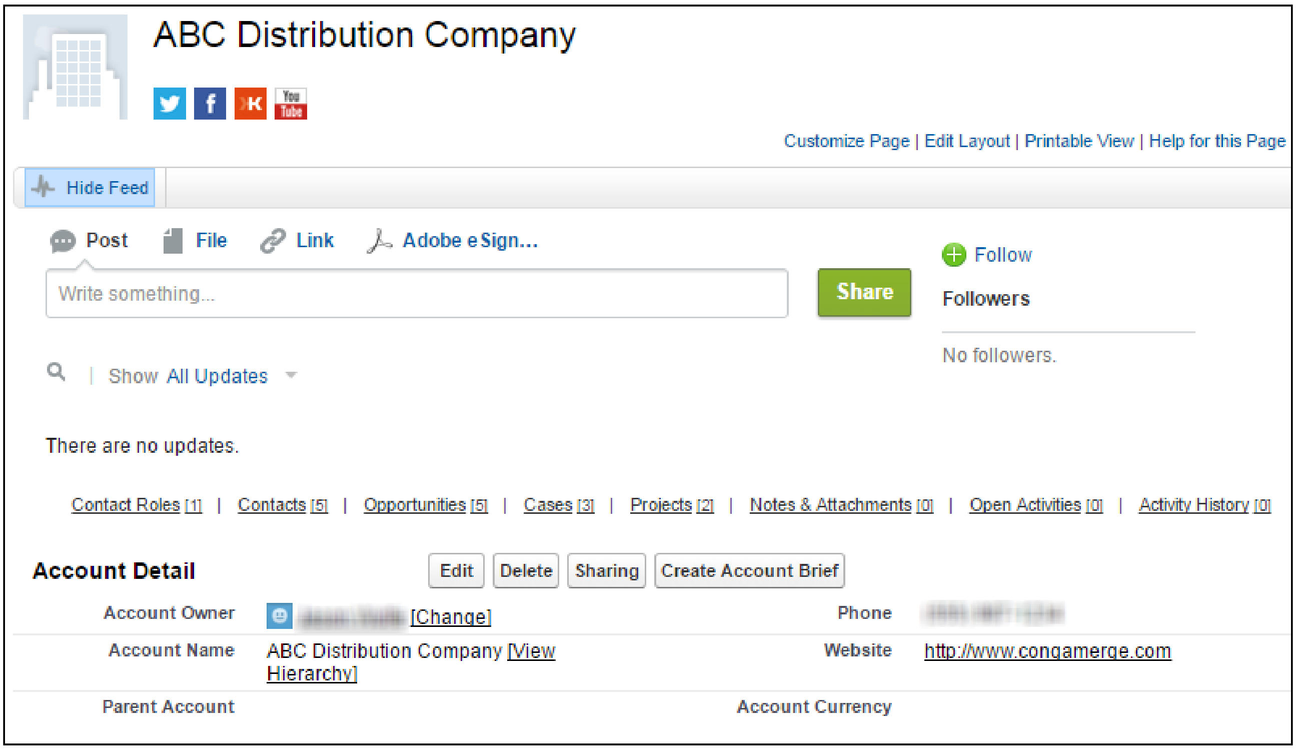
Total Results:	20,211
Sum:	10,227,945,145.00
Average:	506,058.34
Minimum:	16.00
Maximum:	13,200,000.00

View Records in a New Window

Use the New Window action to open a selected record in a new Salesforce window, based on the page layout of the object. Follow these steps:

1. Select a record.
2. Click **Actions**.
3. Click **View Records > New Window**.

Here is an example of a selected Account record opened in a new browser window:



ABC Distribution Company

Customize Page | Edit Layout | Printable View | Help for this Page

Hide Feed

Post File Link Adobe e Sign...

Write something...

Share

Follow

Followers

No followers.

Show All Updates

There are no updates.

Contact Roles [1] | Contacts [5] | Opportunities [5] | Cases [3] | Projects [2] | Notes & Attachments [0] | Open Activities [0] | Activity History [0]

Account Detail Edit Delete Sharing Create Account Brief

Account Owner	[redacted] [Change]	Phone	[redacted]
Account Name	ABC Distribution Company [View Hierarchy]	Website	http://www.congamerge.com
Parent Account		Account Currency	

View Records Using List Browse

Use the List Browse action to view multiple records in a List Browse window if you have the List Browse app installed. Follow these steps:

1. Select the records of interest in the Conga Grid Explorer.
2. Click **Actions**.
3. Click **View Records > List Browse**. A List Browse window shows the page layout for each of the selected records. Use the top navigation bar to move between records.

Conga Grid Views

Learn how to manage views in Conga Grid℠.

- Manage Views
- Create a "Recently Viewed" View
- Save Views
- Privacy Settings for Views
- Page Context
- Sticky Views
- Hard-Code a Default View using a View ID

Manage Views

Click the drop-down arrow next to Configure and click Manage Views. The Manage Views screen appears and lists the available Conga Grid views. Use this screen to do the following:

- Load Views
- Open Views
- Delete Views
- Share Views
- Designate a Default View

⚠ See the Save Views topic for instructions on saving views.

Load Views

Here are some different options for loading an existing view:

- Select a view from the **View** drop-down list in the toolbar.
- Select a view in the Manage Views screen, and click **Load View**.

Open Views

Follow these steps to open a selected view in a Salesforce page layout (separate from Conga Grid) for further editing or sharing:

1. In the Manage Views screen, choose a view by selecting its check box.
2. Click **Open View**.

Delete Views

Follow these steps to delete a view:

1. In the Manage Views screen, choose a view by selecting its check box.
2. Click **Delete View**. The Confirm Deletion screen appears.
3. Click **Yes**.

Share Views

Follow these steps to share private views with other users:

1. In the Manage Views screen, choose a private view by selecting its check box.
2. Click **Share View**. This opens a Salesforce tab containing a list of all users who have access to the view.
3. Click **Add**. The New Sharing screen appears.
4. Select **Users** from the Search drop-down list.
5. Select the users to share with from the **Available** list, and click **Add** to add them to the **Share With** list.
6. Select **Read Only** or **Read/Write Access** from the **Access Level** drop-down list.
7. Click **Save**.

Designate a Default View

A default view is the first view you see in Conga Grid when you access a Salesforce object or when you click **Reset**. Follow these steps to make an existing view the default view:

1. In the Manage Views screen, choose a view by selecting its check box.
2. Select the **Default** check box for that view.
3. Click **Save**.

You can also hard-code a view to be the default view. See Hard-Code a Default View using a View ID.

Create a "Recently Viewed" View

This topic shows an example using the Contact object. The view is similar to a Recent Contacts list view.

1. Click **Conga Grid Setup**.
2. Select the desired object from the list of objects (for example, Contact).

3. Click the **object Fields** tab (for example, Contact Fields).
4. Scroll to the **Last Viewed Date** field, and select the **Visible** check box to make the field visible.
5. Click **Save**.
6. Click **Conga Grid Explorer**.
7. Select the same object (for example, Contact).
8. Click the drop-down arrow next to **Configure** and click **Fields**.
9. Add the **Last Viewed Date** and **Owner** fields to the grid.
10. Filter the **Last Viewed Date** field by:
    - **Is not empty** AND
    - **Is equal to** the **Last 7 Days**
11. Click the **Last Viewed Date** field header twice to sort by descending order.
12. (Optional). Filter the **Owner** field to your records by selecting the **My** check box under the filter.
13. Click **Filter**.
14. When you are done filtering, save the new view. In this example, the view is named Contact - Recently Viewed.
15. To turn this view into a Multi-Tabular layout, see Multi-Tabular Conga Grid.

> ⚠ Filtering by Owner lets you only the see recently viewed records that you own, not all of the records that you have recently viewed. Use caution when filtering by owner.

Save Views

Once you have organized and configured your Conga Grid, you can save that view for later use or share it with others. To do so, follow these steps:

1. Click the drop-down arrow next to **Save** and click **Save View As**. The Save View / Save View As screen appears.
2. Enter a **View Name**.
3. From the **Privacy** drop-down list, select **Public** or **Public Read Only**. You cannot set default views to Private. See Privacy Settings for Views.
4. Select the **Default** check box if you want this view to be the default view. A default view is the first view you see in Conga Grid when you access a Salesforce object or when you click the **Reset** button.
5. (Optional). Enter a **Description** for the view.
6. Select whether or not to enable Inline Edit Mode, the Reading Pane, and the Toolbar in this view.

7. Click **OK**.

> **Warning**
>
> If you are not the owner of the current view and you modify the view and click the **Save** button, it will only save edits you made to your data. It will not save any changes you made to the view. Be sure to click the drop-down arrow next to Save and select Save View to retain any changes you made to a view.

You can also save a view by expanding the **Toolbar** item in the Configure menu and clicking the **Save View** link.

Privacy Settings for Views

Views have three different Privacy settings: Public, Private, and Public Read Only.

Public views can be viewed by any user that has access to the object.

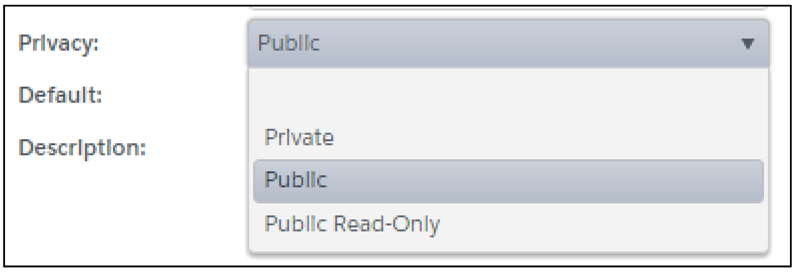
Private views can only be viewed by:

- The user who created the view
- System administrators
- Any users that were selected to share the view

Public Read-Only views are just like Private views except that they cannot be modified by anyone other than the user who created it, or a system administrator. You can save data.

You have two options to set the privacy of a view:

- Click the drop-down arrow next to **Save** and click **Save View**. The Save View / Save View As screen appears. Click the **Privacy** drop-down menu and select the desired option.


Privacy: Public

- Click the drop-down arrow next to **Configure** and click **Manage Views**. The Manage Views screen shows all of the existing views. Modify the Privacy field as needed, then click **Save**.

To update the privacy of multiple views simultaneously (especially if you have a large number of views), consider using the Mass Update action. Select the views to update in the

Manage Views screen. Click **Actions** (in the Manage Views screen) and select **Batch Edit > Mass Update**. Then update the privacy for the views.

Page Context

The three contexts are Visualforce Page, Explorer, and Page Layout. The key difference among them is where they are accessed from in Salesforce. Views with Explorer and Visualforce Page contexts are available from the Conga Grid Explorer or a Visualforce page of the object.

To change the context of a view, click the drop-down arrow next to **Configure** and select **Manage Views**. Then modify the **Context** field in the Manage Views screen.

> ⓘ **Tip**
>
> If the Context field is not visible, click **Configure** in the Manage Views screen. Click the Context check box under Conga Grid View Fields.

Views with a Page Layout context will not be available from the Conga Grid Explorer or Visualforce pages. They can only be accessed from a related list Conga Grid on a Salesforce page layout.

Sticky Views

This feature loads the most recent state of your grid without requiring you to save a view. It works in conjunction with your local browser cache. If you clear your browser cache, Conga Grid will load the default view instead.

Sticky views are enabled by default throughout Conga Grid.

You can turn off this feature by clearing the Enable Sticky Views check box when creating the following items:

- Conga Grid related lists
- Conga Grid Visualforce pages

Sticky views will only work when the Default View box is blank, as with the above example.

You can also disable sticky views for Visualforce pages once they have been created. Follow these steps:

1. In Salesforce, click **Setup**.
2. Click Develop and click **Visualforce Pages**.
3. Click the link next to the name of the Visualforce page, for example:

4. Click the **Visualforce Markup** tab.
5. Set the *EnableStickyViews* setting to *false*.
6. Click **Save**.

Hard-Code a Default View using a View ID

1. Click **Conga Grid Explorer**.
2. In the object list, click **Conga Grid Views**.
3. Open the view that you want to set as the default view. You can double-click the associated record or click the **View** link.
4. Copy the 18-digit Salesforce ID in your web browser URL, for example:
5. In Salesforce, click **Setup**.
6. Click **Develop** and **Visualforce Pages**.
7. Click the **Edit** link next to the Visualforce page associated with the related list (view).
8. Click the **Visualforce Markup** tab.
9. In the *ViewID* field, paste the view ID field that you copied earlier.
10. Click **Save**.

Opening this Visualforce page will now load the specified view as the default view.

This overrides the default settings of a view. So if you hard-code a view that is not set as the default view, the Visualforce page treats the hard-coded view as the default view.

Manually Updating a Grid VisualForce page

Grid Visualforce pages are created through the Grid Setup Page or the Conga Grid QuickStart as Single Tabs, MultiTabs, or Related Lists.

Single Tab Grid

```
<apex:page sidebar="false" showHeader="true" docType="html-4.01-strict">
<CRMC_PP:Grid ObjectName="Account" DelayLoad="false" ViewID="a1B1U000001q6wRUAQ"
EnableNewButton="true" EnableNewInline="true" EnableEdit="true" EnableActions="true"
EnableFieldChooser="true"
EnableStickyViews="true" EnableToolbar="true" EnableViews="true" EnableFormatting="true"
EnableReadingPane="true" />
<CRMC_PP:DrillUp />
</apex:page>
```

Above is an example of a Grid of Accounts that are set in a Salesforce Custom Tab.

The Components each have a value of True or False. This is the only value necessary to change in each component. The exception is the ViewID. See Hard-coding a View ID for more information.

Component	Definition
SideBar	• Determines if the Salesforce Sidebar is displayed. • Value is either true or false.
ShowHeader	• Determines if Salesforce Header is displayed. • Value is either true of false.
Grid ObjectName	• Determines the Salesforce Object in the Grid via the Salesforce Object Name.
DelayLoad	• Only effective in MultiTab Grids. • Determines if the Tab loads without being selected. • This value should be False if it is not the Active Tab. • More information about DelayLoad is located in the MultiTab Grid section.
ViewId	• Sets the Default view for this Tab. • This behavior overrides the "Default" setting on the View itself and and will load if using Reset to Default in the view. • To get the ID of the View, see Hard-coding a View ID.
EnableNewButton	• Determines if the New Record button can create a new record in a new window. • Value is either true or false.
EnableNewInline	• Determines if the New Records button can create a new record inline. • Value is either true or false.
EnableEdit	• Determines if the Grid has the Edit button in the Toolbar. • If this is set to false and the Grid is in Inline Edit mode, the Save button no longer appears on the Toolbar. • If the Grid is not in Edit mode, the button appears on the toolbar as Save View and Save View As, but does not have the ability to save data. • Value is either true or false.

EnableActions	• Determines if Actions are invoked on the Grid. • Note: This does not remove the Actions button from the Grid, but disables clicking it and the ability to invoke the Actions menu by right-clicking on the View. • Value is either true or false.
EnableFieldChooser	• Determines if the Field Chooser under the Configure menu is enabled for the Grid. • Value is either true or false.
EnableStickyViews	• Determines if the Grid View is cached in a browser, so that if a user navigates away from the Tab the Grid lives on after making changes to the view, the changes are still there when the user comes back. • Value is either true or false.
EnableToolbar	• Determines if the Toolbar is available for the Grid. • Value is either true or false.
EnableViews	• Determines if the View selection dropdown is available. • Value is either true or false.
EnableFormatting	• Determines if Conditional Formatting is available in the Configure menu. • Any existing Conditional Formatting Rules applied to your view is still active if this is set to false. • Value is either true or false.
EnableReadingPane	• Determines if the ReadingPane is available. • Value is either true or false.

Related List Grid

```
<apex:page standardController= "Account"  showHeader= "false"  docType= "htm
l-4.01-strict" > <CRMC_PP:Grid
ObjectName= "Contact"  FKName= "AccountId"  FKValue= "{!Account.Id}"
 DelayLoad= "false"  ViewID= ""
EnableNewButton= "true"  EnableNewInline= "true"  EnableEdit= "true"
 EnableActions= "true"  EnableFieldChooser= "true"
```

```
EnableStickyViews= "true"  EnableToolbar= "true"  EnableViews= "true"
 EnableFormatting= "true"
EnableReadingPane= "true"  /><CRMC_PP:DrillUp /></apex:page>
```

Above is an example of a Related List Grid of Contacts on an Accounts Page Layout. It contains all the same elements as the Single Tab above, but has these components that are specifically different:

Component	Definition
standardController	• Determines the Parent Object the Related List Grid is available for.
FKName	• The name of the Foreign Key Field that filters the Grid Values. • Standard field name will be the ID of the Parent, ie "AccountID". : This is created automatically and is rarely modified.
FKValue	• The value of the FKName. • Standard value is the ID of the parent • Example: "{!Account.Id}" : This is created automatically and is rarely modified.
DelayLoad	• Only effective in MultiTab Grids. • Determines if the Tab loads without being selected. • This value should be False if it is not the Active Tab. • More information about that can be found below.

MultiTab Grid

```
<apex:page sidebar= "false"  showHeader= "true"  docType= "html-4.01-strict" >
<div id= "tabstrip"  style= "display:none;" >
<ul> <li  class = "k-state-active" >Pending Orders</li><li>This Month</
li><li>Last Month</li><li>This Year</li>
<li><All/li><li>Order Detail (This Year)</li></ul><!-- Pending Orders
--><div style= "overflow:hidden;" >
```

```
<CRMC_PP:Grid ObjectName= "Order"  DelayLoad= "false"  ViewID= "a1B1U000001q
6wM"  EnableNewButton= "true"
 EnableNewInline= "true"  EnableEdit= "true"  EnableActions= "true"
  EnableFieldChooser= "true"
 EnableStickyViews= "true"  EnableToolbar= "true"  EnableViews= "true"
  EnableFormatting= "true"
 EnableReadingPane= "true"  /> <CRMC_PP:DrillUp /> </div><!-- This
Month --><div
 style= "overflow:hidden;" > <CRMC_PP:Grid ObjectName= "Order"  DelayLoad= "
true"  ViewID= "a1B1U000001q6wN"
 EnableNewButton= "true"  EnableNewInline= "true"  EnableEdit= "true"
  EnableActions= "true"
 EnableFieldChooser= "true"   EnableStickyViews= "true"  EnableToolbar= "tr
ue"  EnableViews= "true"
 EnableFormatting= "true"  EnableReadingPane= "true"  /> <CRMC_PP:DrillUp /
>  </div><!-- Last Month -->
 <div style= "overflow:hidden;" > <CRMC_PP:Grid ObjectName= "Order"
  DelayLoad= "true"
 ViewID= "a1B1U000001q6xTMAM"   EnableNewButton= "true"  EnableNewInline= "
true"  EnableEdit= "true"
 EnableActions= "true"  EnableFieldChooser= "true"   EnableStickyViews= "tr
ue"  EnableToolbar= "true"
 EnableViews= "true"  EnableFormatting= "true"  EnableReadingPane= "true"  /
> <CRMC_PP:DrillUp /> </div><!--
 This Year --><div style= "overflow:hidden;" > <CRMC_PP:Grid ObjectName= "
Order"  DelayLoad= "true"
 ViewID= "a1B1U000001q6xU"   EnableNewButton= "true"  EnableNewInline= "tru
e"  EnableEdit= "true"
 EnableActions= "true"  EnableFieldChooser= "true"   EnableStickyViews= "tr
ue"  EnableToolbar= "true"
 EnableViews= "true"  EnableFormatting= "true"  EnableReadingPane= "true"  /
> <CRMC_PP:DrillUp /> </div><!--
 All --><div style= "overflow:hidden;" > <CRMC_PP:Grid ObjectName= "Order"
  DelayLoad= "true"
 ViewID= "a1B1U000001q6xY"   EnableNewButton= "true"  EnableNewInline= "tru
e"  EnableEdit= "true"
 EnableActions= "true"  EnableFieldChooser= "true"   EnableStickyViews= "tr
ue"  EnableToolbar= "true"
 EnableViews= "true"  EnableFormatting= "true"  EnableReadingPane= "true"  /
> <CRMC_PP:DrillUp /> </div>   <!--
 Order Detail - This Year --><div style= "overflow:hidden;" >
 <CRMC_PP:Grid ObjectName= "OrderItem"
 DelayLoad= "true"  ViewID= "a1B1U000001q6yY"   EnableNewButton= "true"
  EnableNewInline= "true"
```

```
EnableEdit= "true"  EnableActions= "true"  EnableFieldChooser= "true"  
 EnableStickyViews= "true"
EnableToolbar= "true"  EnableViews= "true"  EnableFormatting= "true"
  EnableReadingPane= "true"  />
<CRMC_PP:DrillUp /> </div></div><script>$(document).ready(function()
{
$( "#tabstrip" ).kendoTabStrip({ activate: function(e){
 setTimeout(function(){ $(window).resize(); }); }  });
$( "#tabstrip" ).css( "display" ,  "block" );}); </script></apex:page>
```

In the example above, Grid has several Tabs all based on the Order object and each with its own View ID. Each Grid is listed out by the Tab Strip labels. The MultiTab allows for a Grid Visualforce page to have multiple objects, or the same Object with different views loading by default, as exemplified above. This is also available as a Tab, or as a Related List. The components to be mindful are listed below:

Component	Definition
div Id Tabstrip	• This sets up the list of Tabs for each Grid instance. • Each individual tab is determined by their related List option, designated by <li>Name</li>. • The label between this appears in the actual Grid tab itself.
li class = "k state active"	• This determines what Grid Tab loads first. • It is typically the first List Option, but can be manually changed to a different one. • Note: the List option selected to be the k-state-active should have DelayLoad set to False.
DelayLoad	• Determines if the Grid needs to be clicked on to load. • Default is False. • If set to True, the Grid tab must be clicked before it will load. • If the Grid section is the one that corresponds with the K-state-active setting in the tab list, it should be set to False and all others should be set to True. This will cause the other Grid tabs to wait to load until the tab is clicked on, which will help with loading times.

If a user creates a MultiTab Grid and wants to either remove or add a new section, it must be done manually. To add a section, copy an existing segment and update the Object and View Id section, then update the List Options in the position to match.

Advanced Conga Grid Solutions

Learn how to use Visualforce markup code to further customize Conga Grid℠.

Access Parent Field Information

Warning

The content on this page is an example only and NOT supported.

Add the following line of code to the associated Visualforce page, inside of the <script> tag.

window.OBJECT_NAME ={FIELD_NAME:"{!OBJECT_NAME.FIELD_NAME}",...}

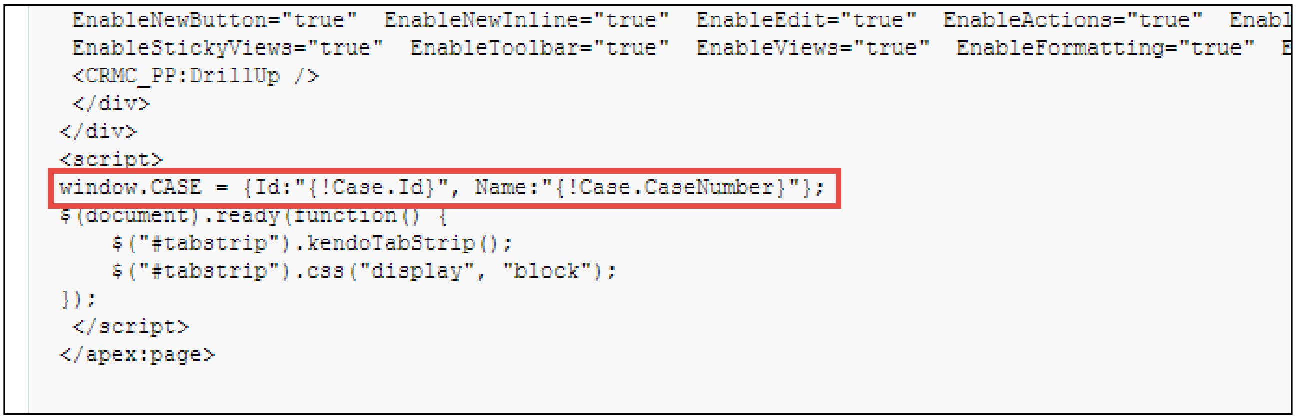
Where:

- OBJECT_NAME = The object name that the related list Visualforce page is located on
- FIELD_NAME: The name of the field that you want to access on the Visualforce page

You can add many instances of the following string as you need. Separate them with commas. This string allows you to access any field that you need.

FIELD_NAME:"{!OBJECT_NAME.FIELD_NAME}"

In the following example, *Case* is the object name,{!Case.Id} is a merge tag, and *CaseNumber* is the field name:



```
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"  Enabl
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  E
 <CRMC_PP:DrillUp />
 </div>
</div>
<script>
window.CASE = {Id:"{!Case.Id}", Name:"{!Case.CaseNumber}"};
$(document).ready(function() {
    $("#tabstrip").kendoTabStrip();
    $("#tabstrip").css("display", "block");
});
 </script>
</apex:page>
```

Accessing Grandchild Data

You can access grandchild data from a parent level in Conga Grid. For example, you can create a grid of Accounts that has a child grid of Contacts and a grandchild grid of the Contact's cases.

To create a grid of Accounts that has a child grid of Contacts and a grandchild grid of the Contact's cases:

1. Go to the Conga Grid Setup tab. Select Accounts > Tabs & Related Lists. In the Tab Creation section, click Single Conga Grid.
2. Follow the instruction to create a Single Conga Grid tab.
3. Return to Conga Grid Setup tab and select Account again. Click Single Conga Grid in the Page Layout section.
4. From the Select Relationship drop-down, select Contact (AccountId). If you already have a view saved for a Related List view of Contacts, select it from the dropdown. Modify the label and click OK.
5. Go to your Conga Grid tab of Accounts. Click Reading Pane > Configure to open the Configure Menu. Click Reading Pane > Visualforce Pages.
6. From the options in the Visualforce Page, check the box next to your grid of Contacts. It appears as a tab in your Reading Pane. Close the Configure Menu in the Accounts grid.
7. In the Contacts grid, click Reading Pane > Bottom.
8. When the Reading Pane in the Contacts grid opens, click the '+' icon to find the Related List selections for Contacts. Check the box next to Cases.

Now when you have a Record selected in the Account Conga Grid, you can see the Contacts related to the Account, and the selected Contact's Cases.

Add a Visualforce Page to a Conga Grid Tab

Before you begin, ensure that the Visualforce page has been added to the Conga Grid Visualforce Pages permission set. See Security and Permission Sets for Visualforce Pages for instructions.

Follow these steps:

1. In Salesforce, click Setup.
2. Click Develop and click Visualforce Pages.
3. Click the link next to the name of the Visualforce page.
4. Click Preview.

5. Copy the URL from your browser address bar, and paste it into your system clipboard.
6. Go back to the Visualforce Pages section of Salesforce Setup.
7. Click the Edit hyperlink next to the same Visualforce page.
8. Click the Visualforce Markup tab, and enter the following code immediately after the opening *<apex>* tag:
   ```
   <style> .fluidMedia { position: relative; padding-bottom: 40%; padding-top: 30px;
   height: 0; overflow: hidden; } .fluidMedia iframe { position: absolute; top: 0;
   left: 0; width: 100%; height: 100%; }</style>
   ```
9. Add a new list tag (*<li>...</li>*) with the name of the tab.
10. Insert the following code in the same order as it appears in the lists, relative to the other tabs. Replace the page name in this example with your own Visualforce page name. In the <iframe> tag, insert the URL that you copied from Step 5.
    ```
    Page name ---><div style="overflow:hidden;"> <div class="fluidMedia">https://etc"
    frameborder="0"> </iframe> </div></div>
    ```
11. Click Save.

Advanced Conditional Formatting Examples

Conga Grid uses JavaScript to create Conditional Formatting rules. For basics on creating Conditional Formatting rules, see Conditional Formatting.

Here are some examples of some more advanced Javascript Rules. Copy and paste the JavaScript Examples in the matrix below into the Conditional Formatting Advanced JavaScript Box.

: The Conga Support Team does not support JavaScript solutions and the creation of advanced formulas. Only best efforts are expected from the Conga Support team using the basic selection criteria.

: Conditional Formatting Order: One thing to keep in mind when creating Conditional Formatting rules is the order of the views, which is modified by clicking and dragging up and down. Rules at the top of the list will override rules below it, so if there are two rules that have conflicting data, that rule that is above the other is addressed.

Opportunity Rule Name and Description	Opportunity Rule Values

<table>
<tr><td>Big Money: Color the Amount field Green and bold/ italicize that field if the field value is above $30,000.</td><td>• Rule Name - Big Money
• Field - Amount
• Cell Color - Unchanged
• Text Color - Green
• Text Style - Bold and Italics enabled
• Apply to Row - False
• JavaScript:
Amount >= 30000</td></tr>
<tr><td>Low or No Activity: Highlight any open Opportunities that have not been modified for over a week.: This uses an OR function, where OR is represented as "||". We are checking in the first half if the current date is older than the LastAcitivityDate field by 7 days, OR if the LastActivityDate field is blank and is not Closed.</td><td>• Rule Name - Low Or No Activity
• Field - Last Activity
• Cell Color - Grey
• Text Color - Black
• Font Style - Bald
• Apply to Row - False
• JavaScript:
(((((new Date() - LastActivityDate) / (1000*60*60*24)) -1) > 7) || (LastActivityDate == null)) && !StageName.contains('Closed')</td></tr>
</table>

Case Rule Name and Description	Case Rule Values
Older than 1 Hour: If a case with a 1 hour SLA is open for more than an hour, highlight the entire record Red. : This is using the same time variation as before, but it checks against hours instead of days. Remove the 24 hour segment of the Time and focus on Hours (the second 60).	• Rule Name - Older than 1 Hour • Field - Contact • Cell Color - Red • Text Color - Black • Font Style - Bold • Apply to Row - True • JavaScript ((((new Date() - CreatedDate) / (1000*60*60)) -1) > 0)

Canceling Conga Sign Transactions in Conga Grid

ⓘ **Important**

> You must have Conga Sign version 1.43.0 and Conga Grid version 2.72 or higher installed in your Salesforce org to leverage this feature. The **Conga Sign - Batch Cancel Transactions** feature is enabled by default when Conga Grid and Conga Sign are present in a Salesforce org.

To cancel Conga Sign Transactions, you must have a Conga Grid that displays Conga Sign Transaction records, or a Conga Grid that displays a record's related Conga Sign Transactions in the Reading Pane.

Easily cancel one or many Conga Sign Transactions in a Conga Grid. This allows Conga Grid users to cancel one or multiple Conga Sign Transactions without having to navigate to different object records.

> **Important**
>
> The Conga Sign Transaction must have a status of Draft or Sent to successfully cancel it. An error will occur when attempting to cancel a transaction that does not have a status of Draft or Sent.

To cancel Conga Sign Transactions in Conga Grid:

1. Navigate to a Conga Grid that displays Conga Sign Transactions or displays related Conga Sign Transactions in the Reading Pane.
2. Select one or multiple Conga Sign Transaction records.
3. Click Actions.
4. Hover over the Conga Sign action option.
    - If one Conga Sign Transaction is selected, click Cancel Transaction.
    - If multiple Conga Sign Transactions are selected, click Cancel Transactions.

Canceling a single transaction cancels the selected Conga Sign Transaction instantaneously. Canceling multiple transactions submits a **bulk cancellation request** (refer to the *Cancel Conga Sign Transactions in Bulk* section in the *Conga Sing* guide) using Apex code, and does not cancel the transactions instantaneously. Refresh your Conga Grid after canceling multiple transactions to ensure the statuses of the transactions update to Cancelled.

As the Conga Sign - Batch Cancel Transactions feature is enabled by default, you must disable the feature in Conga Sign Setup to prohibit users from canceling transactions in Conga Grid.

To disable the Conga Sign - Batch Cancel Transactions feature:

- Navigate to Conga Grid Setup.
- Select Global Defaults, or a specific object to disable the feature on.

- Locate the Conga Sign - Batch Cancel Transactions feature and change the value from Enable to Disable.
- Click Save.

Chatter

Conga Grid utilizes the Salesforce Chatter product to communicate with other members of their org. Conga Grid users can request actions be taken on specific fields in a grid and see that request in Chatter. Responsible parties receive a notification of the request and can see notification indicators on fields in their grid. The @mention tag is used to notify a specific person using Chatter. The @mentioned party sees the message in Chatter and is able to respond accordingly.

All activities, @mentions, and requests can be seen in Chatter; Salesforce's instant messaging product. For more information on Salesforce Chatter, see https://www.salesforce.com/products/chatter/overview/

If you want to request actions on fields in a grid, see Request Actions.

Formula Mass Update

The Formula Mass Update feature allows you to quickly update many records with values that are specific to each individual record using a formula you create. In the Conga Grid set up, you can use the Formula Mass Update feature to take a value created in a formula and mass populate a field to that value.

For example, you can use a formula to show you updated amounts in an Opportunity if a 20 percent discount is applied. You can apply the formula to update an individual record or use the formula mass update feature and apply the formula to multiple records.

Another example is you might have a formula that updates the close date to three weeks out from a date specified. You could use the formula mass update feature to apply that formula to multiple records so the date is updated on multiple records on the close date field.

Limiting View Privacy Choices for Users

You can limit which View Privacy choices are available to Standard Users.

To limit which Privacy options your users have when saving a view, create a new record type for the Conga Grid View object:

1. Navigate to Setup in Salesforce.
2. Click Create under the Build section and then click Objects.
3. Click Conga Grid View.
4. Hover over the Record Types link and click New.
5. Enter a Record Type Label and Description.

   > ⚠ The Name label is filled automatically.

6. Choose the Profiles that you want to set a privacy limit.
7. Click Next and then click Save.
8. On the resulting page, click Edit next to the Privacy field in the Picklists Available for Editing list section.
9. Select Public and Public Read Only, then remove them from the Selected Values area.
10. Click Save.

The profiles where you selected to have your new Record Type as the default should see the Private option only when you save views.

Manage Activities

Problem

Conga Grid functionality is limited with the ActivityHistory and OpenActivity objects. For example, you may not be able to add in place. However, these objects are frequently used when replacing default related lists on page layouts.

Solution

While Conga Grid can display data from the ActivityHistory and OpenActivity objects, Salesforce considers these read-only objects. They are special objects referred to as Views because they combine data from the Task and Event objects. In most cases, we recommend using the Task and Event objects directly and in separate grids. However, the disadvantage with this approach is that ActivityHistory and OpenActivity users are often unfamiliar with it. If you want to use the ActivityHistory and OpenActivity objects, be aware of the following limitations.

1. Adding in place is currently disabled because you must specify if the row is a Task or Event.
2. Editing in place currently works but is designed to update the underlying Task or Event.
3. Users without View all Data access will receive SOQL errors when trying to use ActivityHistory or OpenActivity objects.
4. ActivityHistory and OpenActivity records can only be used as related lists due to SOQL limitations with Read-Only Objects.

You can use the Visualforce page creation wizard in Conga Grid Setup to create an OpenActivity or ActivityHistory Visualforce page. Here are the critical attributes to define in the Visualforce page:

standardController: Specify the page layout object.

ObjectName: Set this to OpenActivity or *ActivityHistory*.

FKName and FKValue: Set these according to the page layout object and relationship.

FKRelationship: Required. Specify the relationship name, depending on *OpenActivities* or *ActivityHistory* and the page layout. This attribute is specific to Read-Only Objects. It also applies to objects such as *NotesAndAttachments*.

Examples:

<apex:page standardController="Account" showHeader="false" docType="html-4.01-strict"><c:Grid ObjectName="OpenActivity" FKName="AccountId" FKValue="{!Account.Id}" FKRelationship="OpenActivities" /><c:DrillUp /></apex:page>

<apex:page standardController="Account" showHeader="false" docType="html-4.01-strict"><c:Grid ObjectName="ActivityHistory" FKName="AccountId" FKValue="{!Account.Id}" FKRelationship="ActivityHistories" /><c:DrillUp /></apex:page>

To approximate a Conga Grid of Events or Tasks that shows the same data as the ActivityHistory view, create a saved view with the following filter criteria:

1. Task Closed field is true.
2. Due Date is in the past.

To approximate a Conga Grid of Events or Tasks that shows the same data as the OpenActivity view, create a saved view with the following filter criteria:

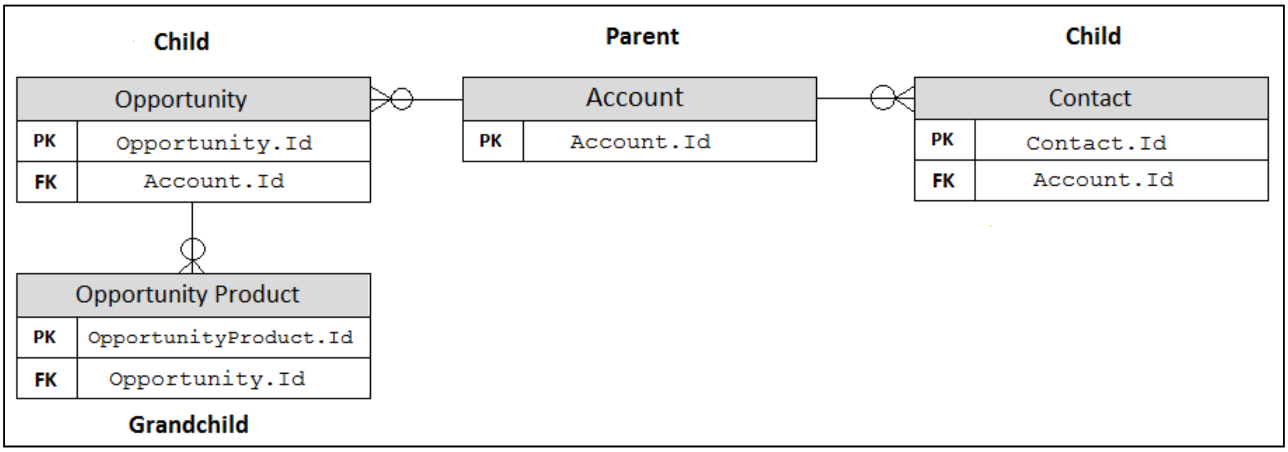
1. Task Closed field is false.
2. Due Date is in the future.

Manage Object Relationships in Related Lists

Conga Grid can display related lists. Related lists include:

- Loosely related (for example, a grandchild record) or even matched to another object through a common field value
- Is completely disconnected from the object that you're currently viewing

Object relationships in Conga Grid are managed with primary keys and foreign keys, just like any relational database management system. The following diagram shows an example of an Account hierarchy. An Account can have many Contacts and Opportunities, both of which are child objects. An Opportunity can have many Opportunity Products. An Opportunity Product is a grandchild of Account.



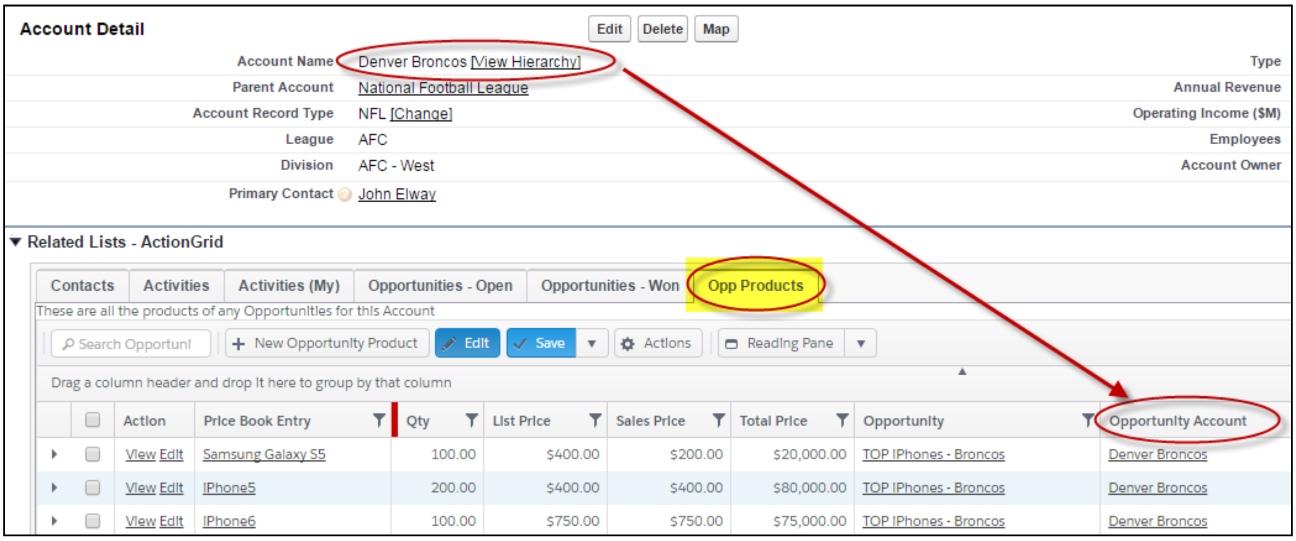
Once you have created a Multi-Tabular related list (automatically or manually), you can edit the Visualforce Markup code of the page layout to add new tabs for child and grandchild objects. You can pass any value from the page layout into the FKValue parameter. Use the FKName parameter (the matching value) to define fields that are one object up in the hierarchy.

See the following sections for details.

- Grandchild Relationships with Standard Objects
- Grandchild Relationships with Custom Objects
- Partially Disconnected Objects
- Completely Disconnected Objects
- Multi-Tabular Related List of Grandchild Objects

Grandchild Relationships with Standard Objects

This example shows an Account Multi-Tabular related list. We want to view the Opportunity Products (in a separate tab) related to a given Account record.



Account Detail Edit Delete Map

Account Name	Denver Broncos [View Hierarchy]	Type	
Parent Account	National Football League	Annual Revenue	
Account Record Type	NFL [Change]	Operating Income ($M)	
League	AFC	Employees	
Division	AFC - West	Account Owner	
Primary Contact	John Elway		

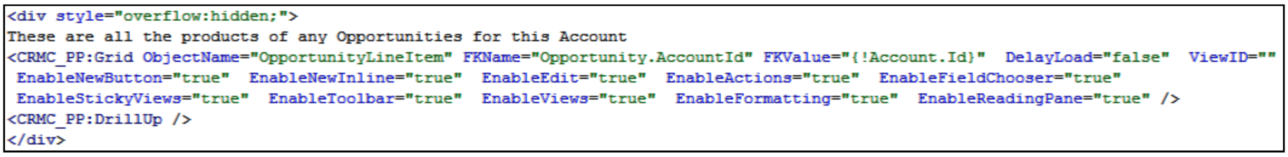
▼ Related Lists - ActionGrid

These are all the products of any Opportunities for this Account

Drag a column header and drop it here to group by that column

	□	Action	Price Book Entry	Qty	List Price	Sales Price	Total Price	Opportunity	Opportunity Account
▸	□	View Edit	Samsung Galaxy S5	100.00	$400.00	$200.00	$20,000.00	TOP IPhones - Broncos	Denver Broncos
▸	□	View Edit	IPhone5	200.00	$400.00	$400.00	$80,000.00	TOP IPhones - Broncos	Denver Broncos
▸	□	View Edit	IPhone6	100.00	$750.00	$750.00	$75,000.00	TOP IPhones - Broncos	Denver Broncos

Here is the Visualforce Markup code for the Opp Products tab. You can pass any value from the Account page to any object. This matches the FKValue to the FKName. Here, we pass in the AccountId to the matching Opportunity.AccountId.



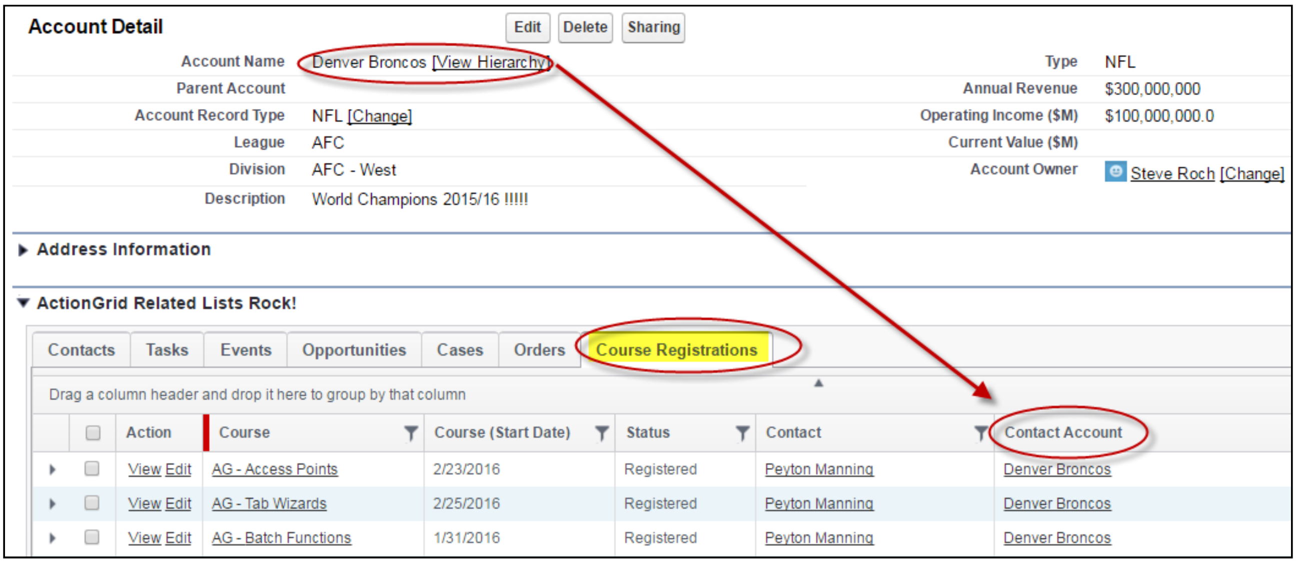
```
<div style="overflow:hidden;">
These are all the products of any Opportunities for this Account
<CRMC_PP:Grid ObjectName="OpportunityLineItem" FKName="Opportunity.AccountId" FKValue="{!Account.Id}"  DelayLoad="false"  ViewID=""
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"  EnableFieldChooser="true"
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  EnableReadingPane="true" />
<CRMC_PP:DrillUp />
</div>
```

The example is essentially running the following query:

SELECT CourseRegistrationsWHERE CourseRegistration.Contact.AccountId = @AccountId

Grandchild Relationships with Custom Objects

This example shows an Account Multi-Tabular related list that displays Course Registrations (a custom object) linked to an Account.



Account Detail Edit Delete Sharing

Account Name	Denver Broncos [View Hierarchy]	Type	NFL
Parent Account		Annual Revenue	$300,000,000
Account Record Type	NFL [Change]	Operating Income ($M)	$100,000,000.0
League	AFC	Current Value ($M)	
Division	AFC - West	Account Owner	Steve Roch [Change]
Description	World Champions 2015/16 !!!!!		

▸ Address Information

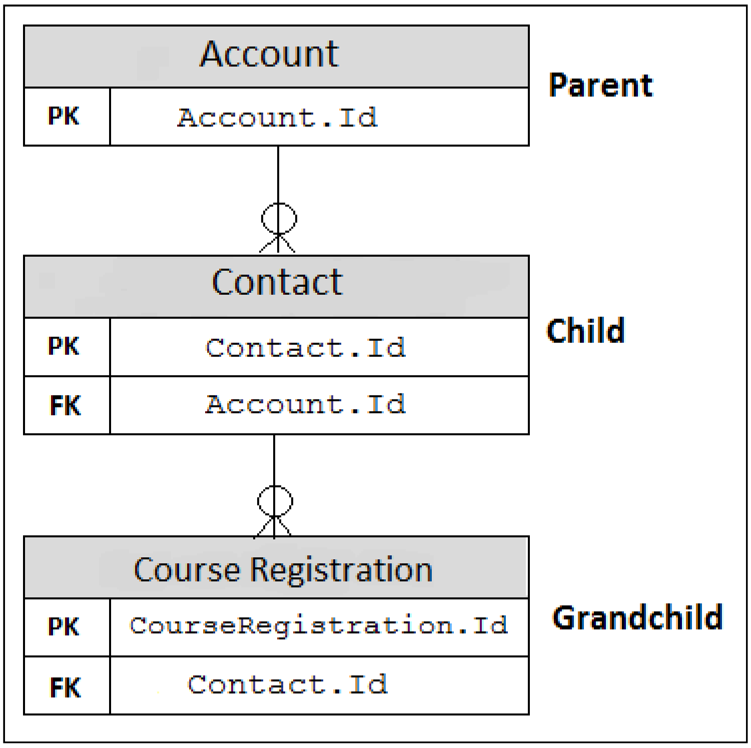
▾ ActionGrid Related Lists Rock!

Contacts | Tasks | Events | Opportunities | Cases | Orders | Course Registrations

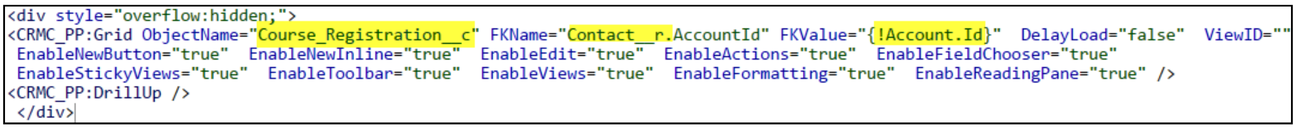
Drag a column header and drop it here to group by that column

	□	Action	Course	Course (Start Date)	Status	Contact	Contact Account
▸	□	View Edit	AG - Access Points	2/23/2016	Registered	Peyton Manning	Denver Broncos
▸	□	View Edit	AG - Tab Wizards	2/25/2016	Registered	Peyton Manning	Denver Broncos
▸	□	View Edit	AG - Batch Functions	1/31/2016	Registered	Peyton Manning	Denver Broncos

The following diagram shows the object hierarchy:



Below is the Visualforce Markup code for the Course Registrations tab. With custom objects, you must use the __r syntax to traverse upward in the hierarchy to match a field from the parent object. Here, we use the __r syntax with Contact to traverse to the parent object, Account.



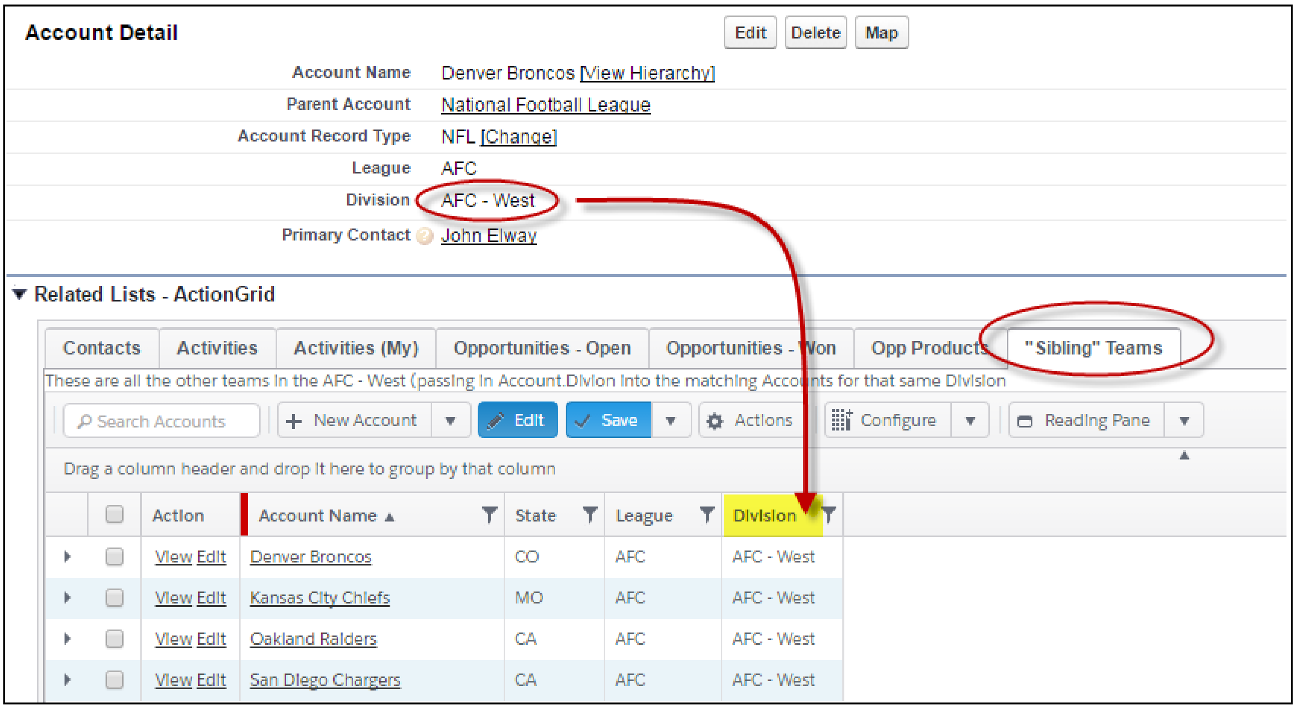
```
<div style="overflow:hidden;">
<CRMC_PP:Grid ObjectName="Course_Registration__c" FKName="Contact__r.AccountId" FKValue="{!Account.Id}"  DelayLoad="false"  ViewID=""
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"  EnableFieldChooser="true"
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  EnableReadingPane="true" />
<CRMC_PP:DrillUp />
 </div>
```

The example is essentially running the following query:

SELECT CourseRegistrationsWHERE CourseRegistration.Contact.AccountId = @AccountId

Partially Disconnected Objects

This example shows an Account Multi-Tabular related list. The Sibling Teams tab displays all other teams in the AFC-West division. It shows how you can pass in any field other than an Id field, to find matching records.



Account Detail

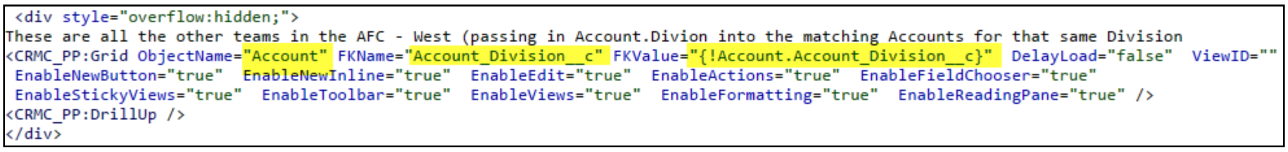
Account Name: Denver Broncos [View Hierarchy]
Parent Account: National Football League
Account Record Type: NFL [Change]
League: AFC
Division: AFC - West
Primary Contact: John Elway

Related Lists - ActionGrid

These are all the other teams in the AFC - West (passing in Account.Divion into the matching Accounts for that same Division

Action	Account Name ▲	State	League	Division
View Edit	Denver Broncos	CO	AFC	AFC - West
View Edit	Kansas City Chiefs	MO	AFC	AFC - West
View Edit	Oakland Raiders	CA	AFC	AFC - West
View Edit	San Diego Chargers	CA	AFC	AFC - West

Here, we pass in the Account.Division field from the current page to a matching Account object for the same division.

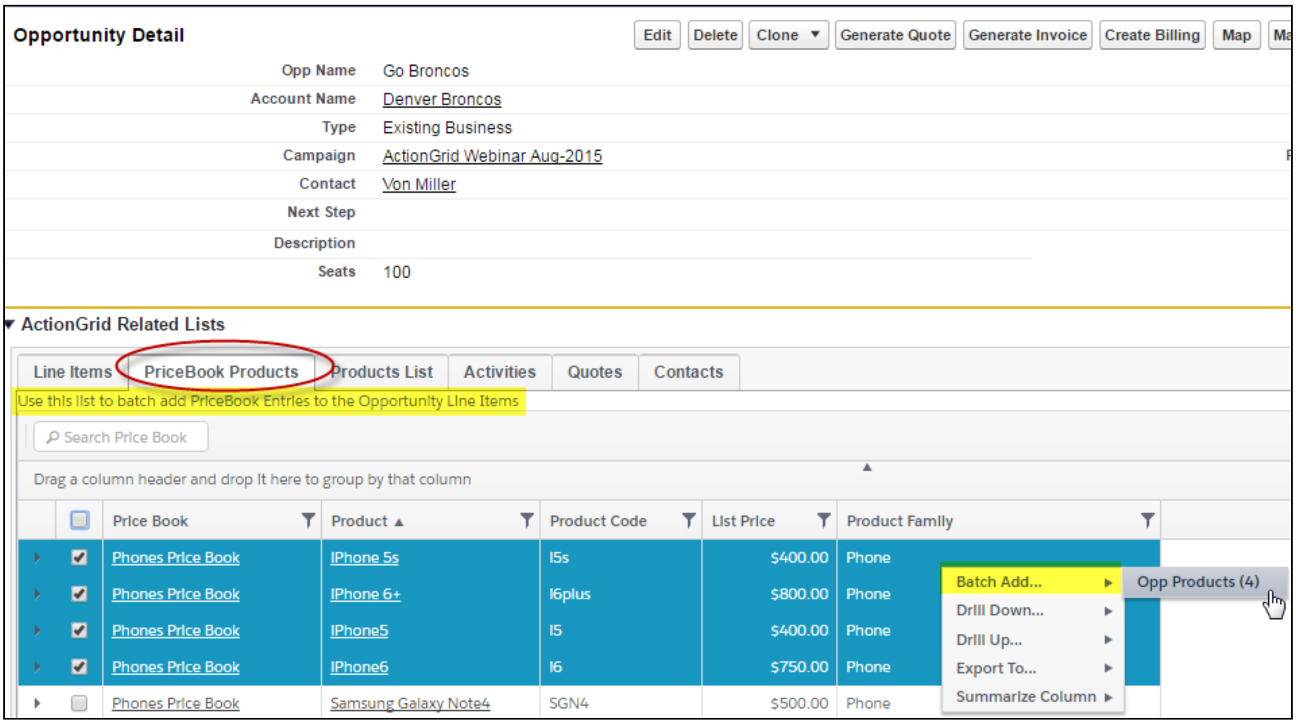


```
 <div style="overflow:hidden;">
These are all the other teams in the AFC - West (passing in Account.Divion into the matching Accounts for that same Division
<CRMC_PP:Grid ObjectName="Account" FKName="Account_Division__c" FKValue="{!Account.Account_Division__c}"  DelayLoad="false"  ViewID=""
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"  EnableFieldChooser="true"
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  EnableReadingPane="true" />
<CRMC_PP:DrillUp />
</div>
```

You could use a similar approach, for example, to match Leads to Contacts using the Lead.Email = Contact.Email match.

Completely Disconnected Objects

This example shows a related list of completely disconnected objects, where the objects do not match by any field or key. The list appears to look like a related list, but it actually just a list of active Price Book entries that you can batch-add to Opportunity Line Items.

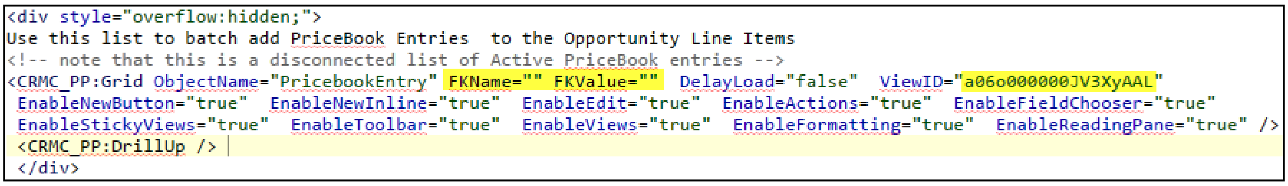


Opp Name	Go Broncos
Account Name	Denver Broncos
Type	Existing Business
Campaign	ActionGrid Webinar Aug-2015
Contact	Von Miller
Next Step	
Description	
Seats	100

Use this list to batch add PriceBook Entries to the Opportunity Line Items

	□	Price Book	Product ▲	Product Code	List Price	Product Family
▸	☑	Phones Price Book	iPhone 5s	i5s	$400.00	Phone
▸	☑	Phones Price Book	iPhone 6+	i6plus	$800.00	Phone
▸	☑	Phones Price Book	iPhone5	i5	$400.00	Phone
▸	☑	Phones Price Book	iPhone6	i6	$750.00	Phone
▸	□	Phones Price Book	Samsung Galaxy Note4	SGN4	$500.00	Phone

Here is the markup code for the PriceBook Products tab:



```
<div style="overflow:hidden;">
Use this list to batch add PriceBook Entries  to the Opportunity Line Items
<!-- note that this is a disconnected list of Active PriceBook entries -->
<CRMC_PP:Grid ObjectName="PricebookEntry" FKName="" FKValue=""  DelayLoad="false"  ViewID="a06o000000JV3XyAAL"
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"  EnableFieldChooser="true"
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  EnableReadingPane="true" />
 <CRMC_PP:DrillUp /> 
 </div>
```

Multi-Tabular Related List of Grandchild Objects

This example shows how to pass in the AccountId for the record you are viewing while matching it to the Account.ParentId to obtain all the grandchild records of the parent (such as Contacts and Opportunities) of all the subsidiary accounts.

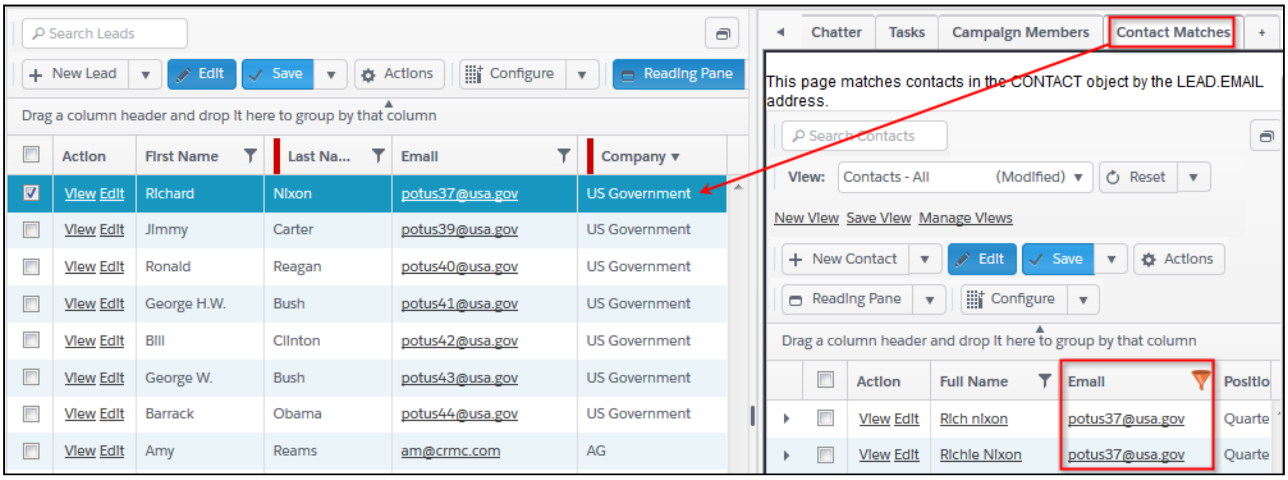
- Each *<div>...</div>* tag set corresponds to a tab, in a specific order.
- Use the FKName parameter (the foreign key name) to set the related objectName and its relationship to the primary object. Then pass in the ID of the primary object using the FKValue parameter.
- You can also pass in other ID fields to objects that may not be directly related to the primary object.
- Refer to the Add New Tabs topic for instructions on hard-coding a specific view to the list. You can do this instead of inheriting the default view or the last view used.
- Setting the DelayLoad parameter to False will not load the data in the grid until a user clicks the tab. This improves the page-loading performance.

```
 <apex:page standardController= "Account"  showHeader= "false"  DocType= "
html-4.01-strict" > <div id= "tabstrip"  style= "display:none;" > <!---
Define the actual tabs and their order ---> <ul> <li  class = "k-state-
active" >Sub Accounts</li> <li>Contacts in Account Hierarchy</li>
<li>Opportunities in Account Hierarchy</li> <li>Activity in Account
Hierarchy</li> </ul> <!--- Sub-Accounts (child accounts) ---> <div
style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "Account"
  FKName= "ParentId"  FKValue= "{!Account.Id}"  ViewId= ""
  EnableNewButton= "true"  EnableNewInline= "true"  EnableEdit= "true"
  EnableActions= "true"  EnableFieldChooser= "true"  EnableStickyViews= "
true"  EnableToolbar= "true"  EnableViews= "true"  EnableFormatting= "t
rue"  EnableReadingPane= "true"  DelayLoad= "true"  /> </div> <!---
CONTACTS - pass in Account.Id to the Contact.Account.ParentId to get
all the contacts at all the child Accounts ---> <div
style= "overflow:hidden;" > <CRMC_PP:Grid objectName= "Contact"
  FKName= "Account.ParentId"  FKValue= "{!Account.Id}"  ViewId= "a06o000
000Ix3PDAAZ"  EnableNewButton= "false"  EnableNewInline= "true"
  EnableEdit= "true"  EnableActions= "true"  EnableFieldChooser= "true"
  EnableStickyViews= "true"  EnableToolbar= "true"  EnableViews= "false"
  EnableFormatting= "true"  EnableReadingPane= "true"  DelayLoad= "true"
/> </div> <!--- OPPORTUNITIES - pass in Account.Id to the
Opportunity.Account.ParentId to get all the Opportunities at all the
child Accounts ---> <div style= "overflow:hidden;" > <CRMC_PP:Grid
objectName= "Opportunity"  FKName= "Account.ParentId"  FKValue= "{!
Account.Id}"  ViewId= "a06o000000Ix3QaAAJ"  EnableNewButton= "false"
  EnableNewInline= "true"  EnableEdit= "true"  EnableActions= "true"
  EnableFieldChooser= "true"  EnableStickyViews= "true"  EnableToolbar= "
true"  EnableViews= "false"  EnableFormatting= "true"
  EnableReadingPane= "true"  DelayLoad= "true"  /> </div> <!--- TASKS -
pass in Account.Id to the Task.Account.ParentId to get all the Tasks at
all the child Accounts ---> <div style= "overflow:hidden;" >
<CRMC_PP:Grid objectName= "Task"  FKName= "Account.ParentId"  FKValue= "
{!Account.Id}"  ViewId= "a06o000000Ix3Q1AAJ"  EnableNewButton= "true"
  EnableNewInline= "true"  EnableEdit= "true"  EnableActions= "true"
  EnableFieldChooser= "true"  EnableStickyViews= "true"  EnableToolbar= "
true"  EnableViews= "true"  EnableFormatting= "true"
  EnableReadingPane= "true"  DelayLoad= "true"  /> </div> </div>
<script> $(document).ready(function() { $( "#tabstrip" ).kendoTabStrip(
{ }); $( "#tabstrip" ).css( "display" ,  "block" ); }); </script> </
apex:page>
```

Match Leads to Contacts in a Reading Pane Tab

This scenario filters Contact records and only displays the matching ones in a Reading Pane tab. You can expand this to other objects as well.

The following image shows an example.



Drag a column header and drop it here to group by that column

☐	Action	First Name	Last Na...	Email	Company
☑	View Edit	Richard	Nixon	potus37@usa.gov	US Government
☐	View Edit	Jimmy	Carter	potus39@usa.gov	US Government
☐	View Edit	Ronald	Reagan	potus40@usa.gov	US Government
☐	View Edit	George H.W.	Bush	potus41@usa.gov	US Government
☐	View Edit	Bill	Clinton	potus42@usa.gov	US Government
☐	View Edit	George W.	Bush	potus43@usa.gov	US Government
☐	View Edit	Barrack	Obama	potus44@usa.gov	US Government
☐	View Edit	Amy	Reams	am@crmc.com	AG

This page matches contacts in the CONTACT object by the LEAD.EMAIL address.

Drag a column header and drop it here to group by that column

	☐	Action	Full Name	Email	Positio
▸	☐	View Edit	Rich nixon	potus37@usa.gov	Quarte
▸	☐	View Edit	Richie Nixon	potus37@usa.gov	Quarte

Follow these steps:

- Create a Visualforce Page
- Load the Visualforce Page as a Reading Pane Tab

Create a Visualforce Page

1. In Salesforce, click Setup.
2. Click Develop and click Visualforce Pages.
3. Click New.
4. Enter a Label and Name. We suggest the following, where RP stands for Reading Pane:
   - Label: Contact Matches
   - Name: AG_Lead_Contact_Match
5. Enter the following code in the Visualforce Markup area.
   ```
   <apex:page standardController="Lead" showHeader="false" docType="html-4.01-
   strict"> <CRMC_PP:Grid ObjectName="Contact" FKName="Email" FKValue="{!
   Lead.Email}" DelayLoad="false" ViewID="" EnableNewButton="true"
   EnableNewInline="true" EnableEdit="true" EnableActions="true"
   EnableFieldChooser="true" EnableStickyViews="true" EnableToolbar="true"
   EnableViews="true" EnableFormatting="true" EnableReadingPane="true" /
   ><CRMC_PP:DrillUp /></apex:page>
   ```

6. Click Save.

Load the Visualforce Page as a Reading Pane Tab

1. Click Conga Grid Explorer.
2. Select Lead from the object list.
3. Click Reading Pane.
4. In the row of tabs at the top of the Reading Pane, click the tab with the plus sign to display the Configure menu. This tab is at the end of the list; you may need to scroll through the tabs using the arrow icon.
5. Click Reading Pane and click Visualforce Pages.
6. Select the check box for Contact Matches. A Contact Matches tab appears in the Reading Pane.
7. Click the Save View link in the Conga Grid toolbar. This will save your reading pane tabs so they will be available in a future session.

Generic Template

Use the following code to create your own Visualforce page with objects other than Leads and Contacts.

- Replace *<startObject>* with the object that you are currently viewing.
- Replace *<filteredObject>* with the related object you would like filtered in the reading pane.
- Replace *<filteredField>* with the field name that will be matched to the <filteredObject>.
- Replace *<matchingField>* with the field you would like to be matched.

```
<apex:page standardController=<startObject>" showHeader="false" docType="html-4.01-
strict"> <CRMC_PP:Grid ObjectName=<filteredObject>" FKName="<filteredField>"
FKValue=<matchingField>"" DelayLoad="false" ViewID="" EnableNewButton="true"
EnableNewInline="true" EnableEdit="true" EnableActions="true" EnableFieldChooser="true"
EnableStickyViews="true" EnableToolbar="true" EnableViews="true"
EnableFormatting="true" EnableReadingPane="true" /><CRMC_PP:DrillUp /></apex:page>
```

Request Actions

The Grid Request Action is designed to help users make and track requests for data updates by guiding them through a simple request process:

1. A requester asks a user for a change to be made to a field.
2. A Chatter mention goes out letting the user know that a change has been requested and a reminder icon is placed on the field that needs updating.
3. A user makes a change on the requested field and a follow-up chatter is automatically sent to both parties letting them know that an update has been made.

This process closes the gaps that can occur with traditional request processes, keeping your data clean and up to date.

For a video on this feature, see How to Use the Request Action Feature in Grid.

Request Action Usage

You can request actions for a field on a grid to notify a user that action needs to be taken on their data. The user who made a request and the notified party see an icon on the requested field and activity in their Salesforce Chatter window.

To request actions on fields in a grid:

1. Select a record or multiple records. Right-mouse click on the field you would like to make a request for, then Batch Add > Request Action. The Request Action window appears.
2. In the Request Action From field, add the names of the users or groups you would like to request changes from.
3. In the Additional Comments field, add any additional notes about what you would like the users to update.

Request Action limits the number of users in the Request Action From field to 25 and the number of @mentions in the Additional Comments field to 25.

Once a request is made, it can be completed two ways:

1. A user can update the marked field on a grid or within Salesforce
2. The requester can cancel the request under the Actions Menu Batch Edit > Cancel Request

If a user updates the marked field on the grid, a confirmation chatter post will be automatically created and sent out to both the requester and any of the users added to the Request Action From field letting them know that a change has taken place.

A change made by a user on a page layout will still send an automatic chatter, though it may be delayed. A scheduled job checks for changes every hour.

If you would like to cancel the scheduled job or force the job to run, this can be done in Grid Setup by going to Global Defaults > Global Setup > Request Action Monitor.

Limitations

- Request Actions are not intended to work with un-editable fields such as Lookups, Formula Fields or Auto Number FieldsOther Considerations

If you would like to request an action from a group, you must configure Grid to receive email notifications for the group. You will not receive a notification to the group chat like you do when you use a direct @mention.

To configure Grid to receive email notifications for the group on every post:

1. From your org home screen, select Salesforce Chatter from the drop-down menu on the top-right of the window.
2. Click the Groups tab.
3. Click any group where you are a member. Select Email Daily Digest or Email on Every Post depending on your preference. If you select Email on Every Post, you see the email anytime someone @mentions that group.

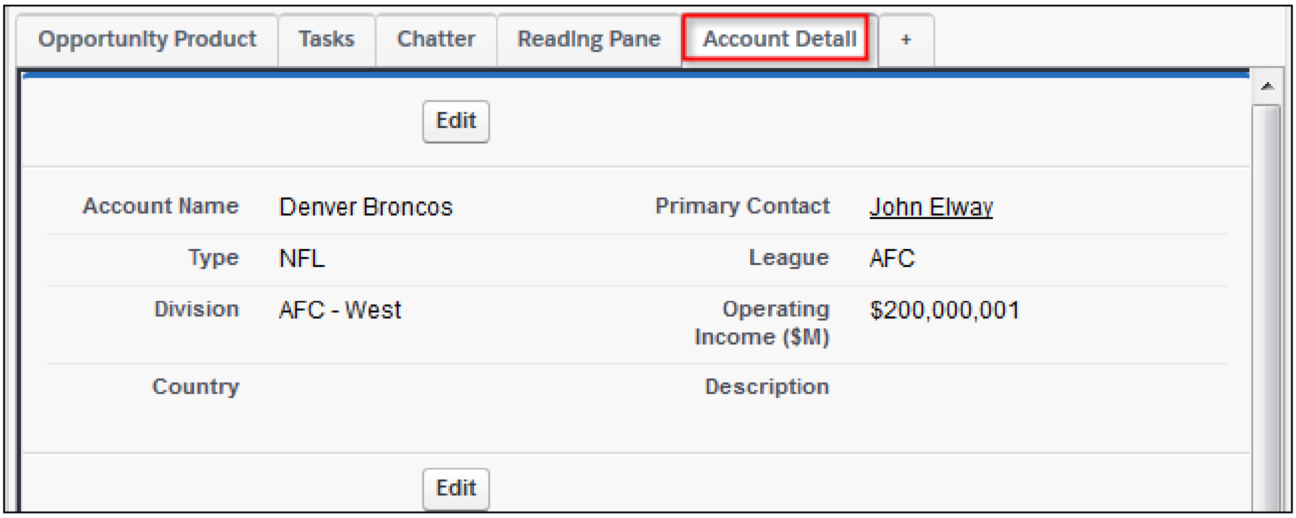
As the creator of the request in Chatter, you won't receive @mention email notifications to yourself or the group where you are a member.

Show Current Object Details in a Reading Pane Tab

Warning

Due to the ever-changing complex nature of programming languages, Conga does not support Visualforce or any elements related to APEX.

The following figure shows an example of Account Details:

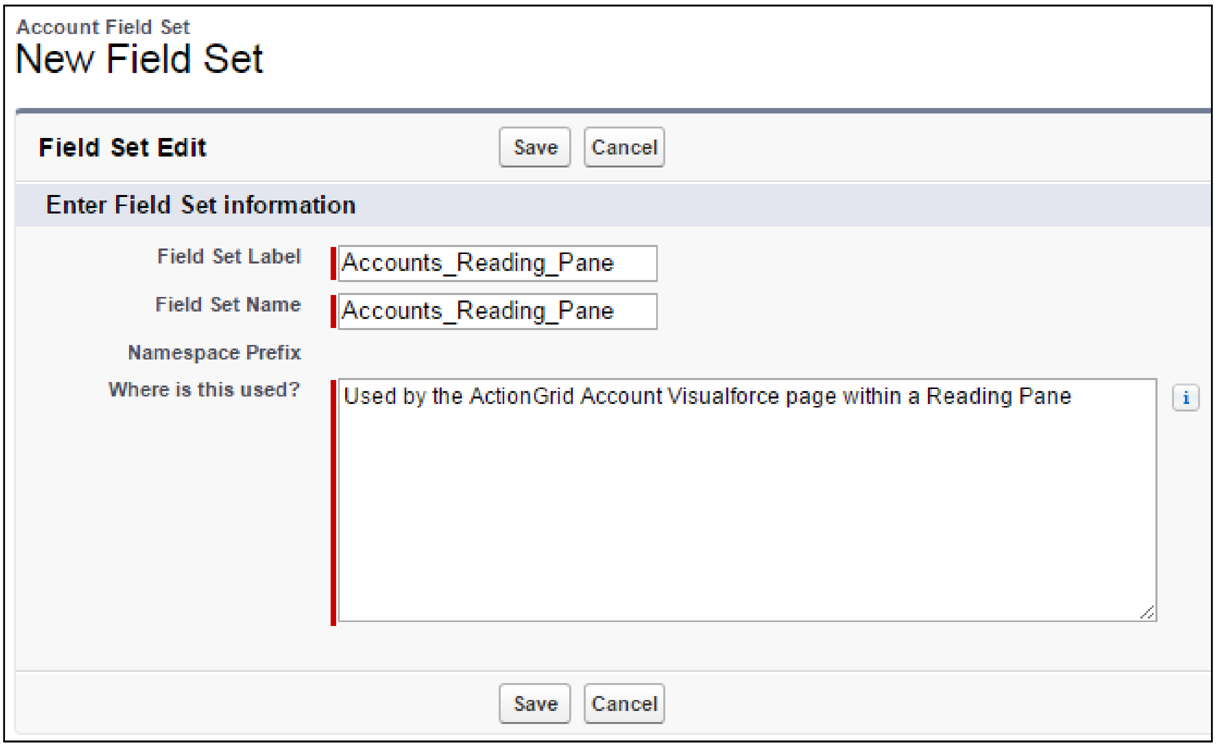


Account Name	Denver Broncos	Primary Contact	John Elway
Type	NFL	League	AFC
Division	AFC - West	Operating Income ($M)	$200,000,001
Country		Description	

Create a Field Set

First, create a field set that contains the details you want to review in the Reading Pane.

1. In Salesforce, click Setup.
2. Click Customize and click the object of interest.
3. Click Field Sets. The object Field Sets page appears.
4. Click New.
5. Enter a Field Set Label and Field Set Name. Write down the Field Set Label for later use.
6. In the Where is this used? box, provide a brief description of where and how the field set will be used. This information helps administrators determine how to populate the installed field set with their own fields. For example:



Account Field Set

New Field Set

Field Set Edit

Enter Field Set information

Field Set Label: Accounts_Reading_Pane

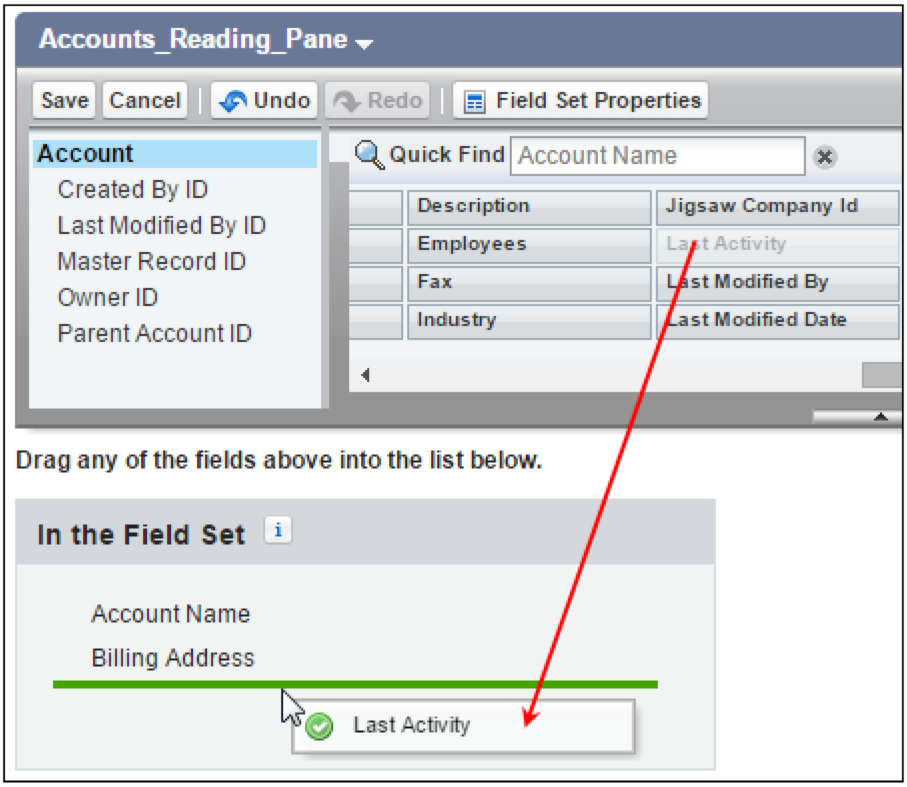
Field Set Name: Accounts_Reading_Pane

Namespace Prefix

Where is this used?: Used by the ActionGrid Account Visualforce page within a Reading Pane

7. Click Save.

8. Add fields to the field set. You can copy a current page layout by viewing it and choosing the fields that match. Or, create your own field set.



9. Click Save.

Create a Visualforce Page for the Object Details

1. In Salesforce, click Setup.
2. Click Develop and click Visualforce Pages.
3. Click New.
4. Enter a Name and Label using the following convention (where "RP" stands for Reading Pane):
   - Label: AG_Object_RPTab (for example, AG_Account_RPTab)
   - Name: Object_Details (for example, Account_Details)
5. Paste the following code in the Visualforce Markup space. Replace Visualforce Markup space. Replace <Object> with the label of the current object (for example, Account). Replace <Field_Set> with the field set label that you recorded in Step 4 of the Create a Field Set section.

```
<apex:page standardController= "<Object>"  showHeader= "false"
 docType= "html-4.01-strict" > <style> select.fullWidth{ width:200px }
</style> <div> <apex:form id= "refresh"  > <apex:messages />
```

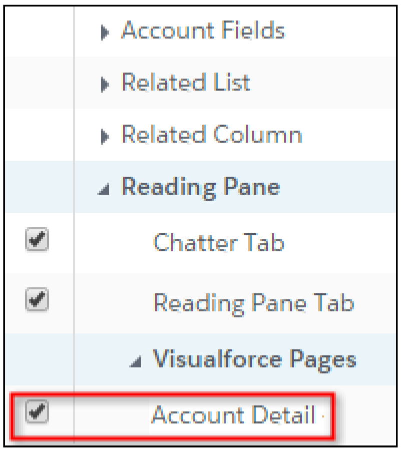
```
<apex:pageBlock id= "editBlock"  html-style= "display:none" >
<apex:pageBlockButtons > <apex:commandButton value= "Save"  action= "{!
quickSave}"  rerender= "refresh" /> <apex:commandButton onclick= "locat
ion.reload()"  value= "Cancel"  rerender= "refresh" /> </
apex:pageBlockButtons> <apex:pageBlockSection columns= "2" >
<apex:repeat value= "{!$ObjectType.<Object>.FieldSets.<Field_Set>}"
 var= "field" > <apex:inputField value= "{!<Object>[field]}"
 styleClass= "fullWidth"  /> </apex:repeat> </apex:pageBlockSection> </
apex:pageBlock> <apex:pageBlock id= "viewBlock" >
<apex:pageBlockButtons > <apex:commandButton
onclick= "document.getElementById('{!
$Component.editBlock}').parentElement.parentElement.parentElement.style
.display = ''; document.getElementById('{!
$Component.viewBlock}').style.display = 'none'; return false "  value= "
Edit"  /> </apex:pageBlockButtons> <apex:pageBlockSection columns= "2" >
 <apex:repeat value= "{!$ObjectType.<Object>.FieldSets.<Field_Set>}"
 var= "field" > <apex:outputField value= "{!<Object>[field]}"
 styleClass= "fullWidth"  /> </apex:repeat> </apex:pageBlockSection> </
apex:pageBlock> </apex:form> </div> </apex:page>
```

6. Click Save.

Load the Visualforce Page as a Reading Pane Tab

1. Click Grid Explorer.
2. Select the object for which you created the Visualforce page.
3. Click Reading Pane.
4. In the row of tabs at the top of the Reading Pane, click the tab with the plus sign. This tab is at the end of the list; you may need to scroll through the tabs using the arrow icon.
5. Click the Reading Pane drop-down list.
6. Click the Visualforce Pages drop-down list.

7. Select the check box next to the name of your Visualforce page, for example:



▸ Account Fields
▸ Related List
▸ Related Column
◢ Reading Pane
☑ Chatter Tab
☑ Reading Pane Tab
◢ Visualforce Pages
☑ Account Detail

8. Click Save in the Grid toolbar. This saves your reading pane tabs so they will be available in a future session.

Show Parent Object Details in a Reading Pane Tab

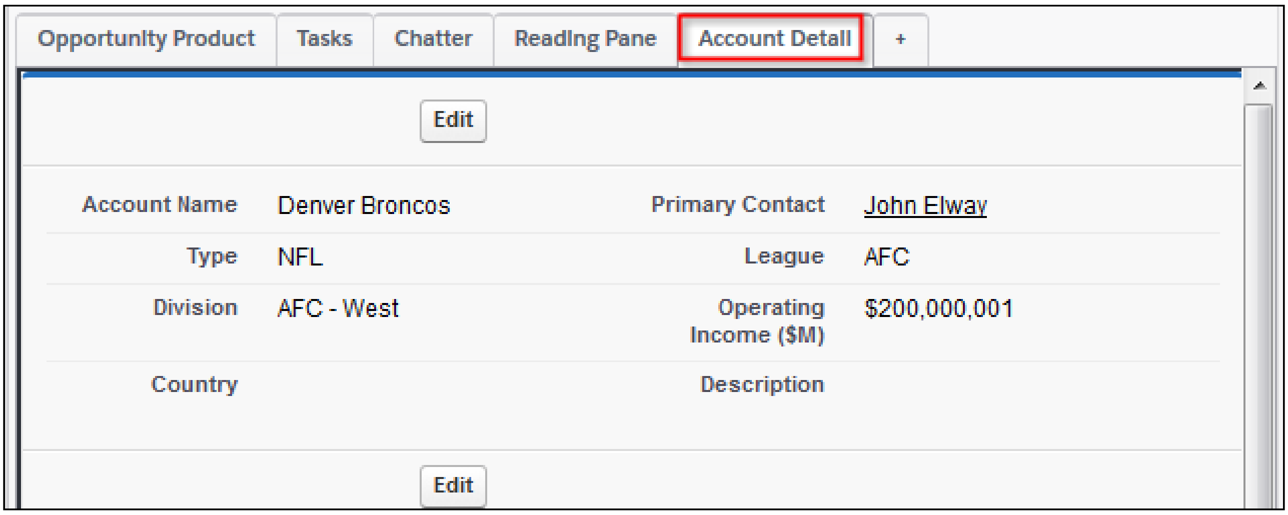
Warning

Due to the ever-changing complex nature of programming languages, Conga does not support Visualforce or any elements related to APEX.

Object: The object you are viewing in the grid.

Parent Object: The Parent Object you want to see details about from the Object. The object is related to the Parent Object in the Lookup or Master-Detail window.

The following figure shows an example of Account details:

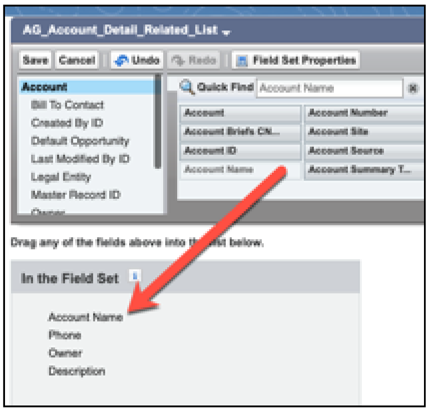


Account Name	Denver Broncos	Primary Contact	John Elway
Type	NFL	League	AFC
Division	AFC - West	Operating Income ($M)	$200,000,001
Country		Description	

Create a Field Set

First, create a field set that contains the details you want to review in the Reading Pane.

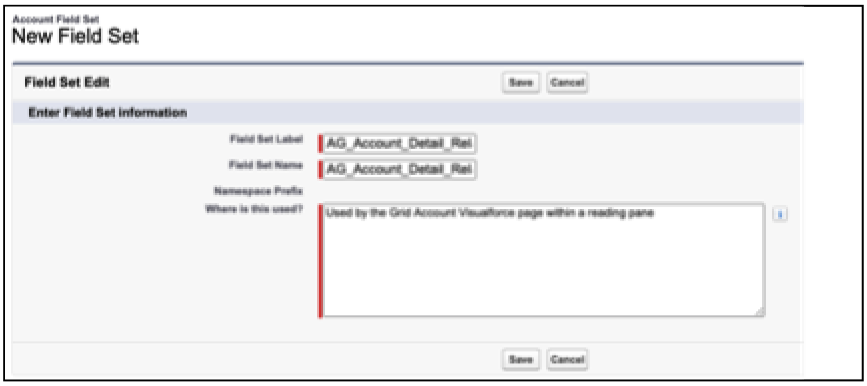
1. In Salesforce, click Setup.
2. Click the Object Manager tab.
3. Click the object of interest.
4. Click Field Sets. The object Field Sets page appears.
5. Click New.
6. Enter a Field Set Label and Field Set Name. Write down the Field Set Label for later use.
7. In the **Where is this used?** box, provide a brief description of where and how the field set will be used. This information helps administrators determine how to populate the installed field set with their own fields. For example:



Drag any of the fields above into the list below.

In the Field Set

Account Name
Phone
Owner
Description



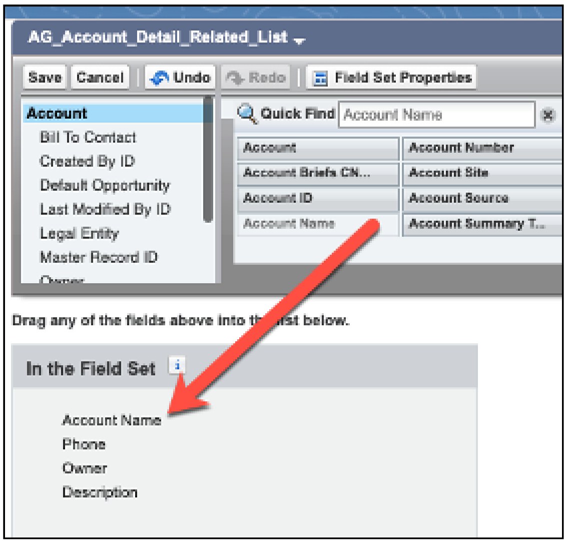
Field Set Label: AG_Account_Detail_Rel

Field Set Name: AG_Account_Detail_Rel

Namespace Prefix:

Where is this used?: Used by the Grid Account Visualforce page within a reading pane

8. Click Save.
9. Add fields to the field set. You can copy a current page layout by viewing it and choosing the fields that match. Or, create your own field set.

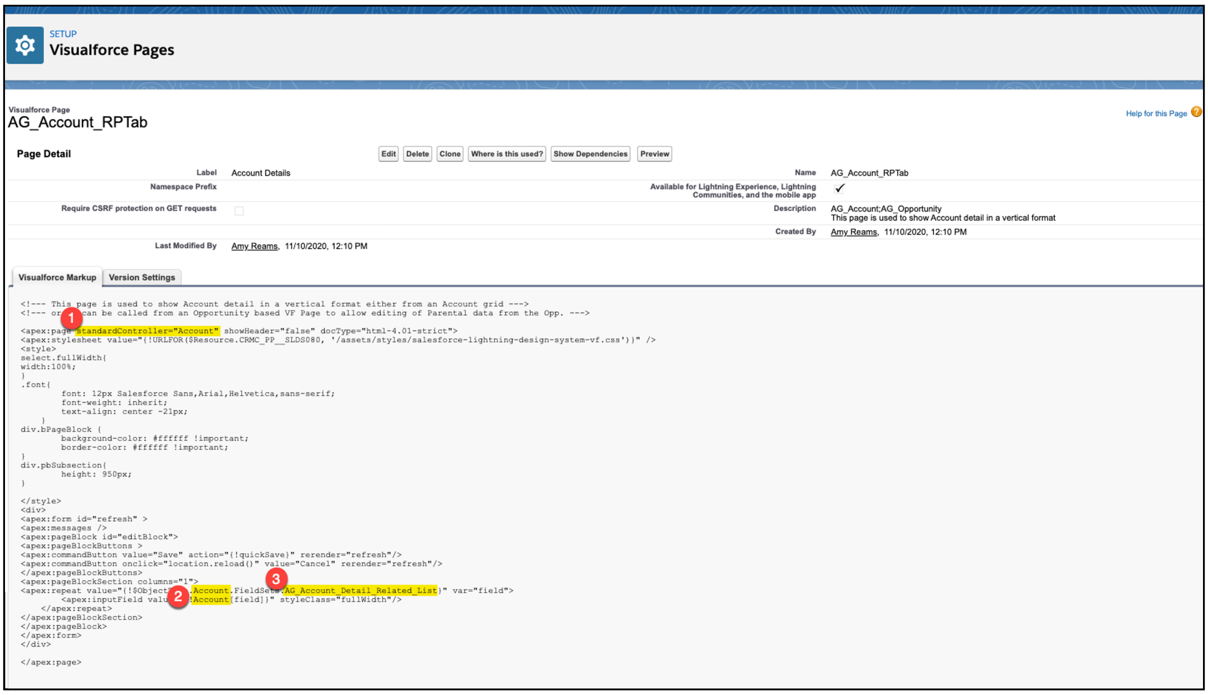


Drag any of the fields above into the list below.

10. Click Save.

Create a Visualforce Page for the Parent Object

1. In Salesforce, click Setup.
2. Under Quick Find search for Visualforce Pages and select Visualforce Pages
3. Click New.
4. Enter a Name and Label.
5. Paste the code from ParentFieldPage.



Visualforce Page

AG_Account_RPTab

Page Detail

Label	Account Details	Name	AG_Account_RPTab
Namespace Prefix		Available for Lightning Experience, Lightning Communities, and the mobile app	✓
Require CSRF protection on GET requests	□	Description	AG_Account;AG_Opportunity This page is used to show Account detail in a vertical format
		Created By	Amy Reams, 11/10/2020, 12:10 PM
Last Modified By	Amy Reams, 11/10/2020, 12:10 PM		

```
<!--- This page is used to show Account detail in a vertical format either from an Account grid --->
<!--- or  can be called from an Opportunity based VF Page to allow editing of Parental data from the Opp. --->

<apex:page standardController="Account" showHeader="false" docType="html-4.01-strict">
<apex:stylesheet value="{!URLFOR($Resource.CRMC_PP__SLDS080, '/assets/styles/salesforce-lightning-design-system-vf.css')}" />
<style>
select.fullWidth{
width:100%;
}
.font{
        font: 12px Salesforce Sans,Arial,Helvetica,sans-serif;
        font-weight: inherit;
        text-align: center -21px;
    }
div.bPageBlock {
        background-color: #ffffff !important;
        border-color: #ffffff !important;
}
div.pbSubsection{
        height: 950px;
}

</style>
<div>
<apex:form id="refresh" >
<apex:messages />
<apex:pageBlock id="editBlock">
<apex:pageBlockButtons >
<apex:commandButton value="Save" action="{!quickSave}" rerender="refresh"/>
<apex:commandButton onclick="location.reload()" value="Cancel" rerender="refresh"/>
</apex:pageBlockButtons>
<apex:pageBlockSection columns="1">
<apex:repeat value="{!$Object  .Account.FieldSets.AG_Account_Detail_Related_List}" var="field">
        <apex:inputField valu  !Account[field]}" styleClass="fullWidth"/>
    </apex:repeat>
</apex:pageBlockSection>
</apex:pageBlock>
</apex:form>
</div>

</apex:page>
```

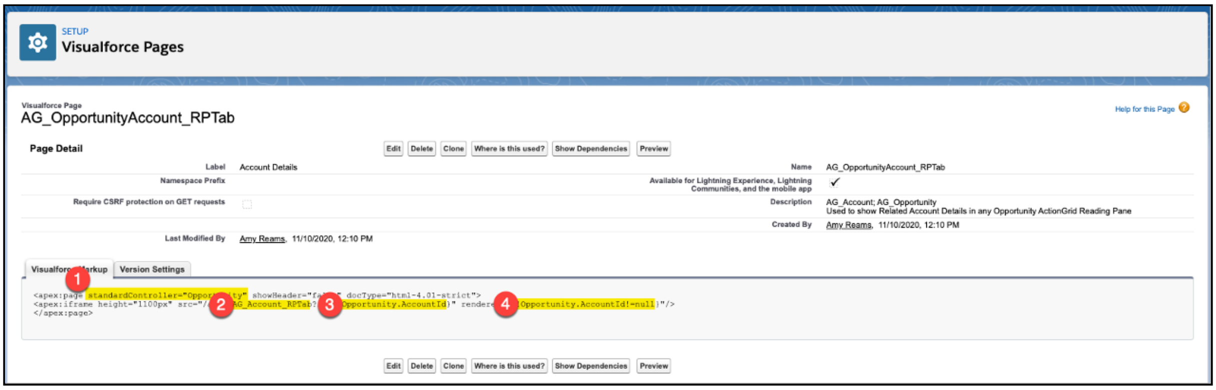
- For the section labeled 1, keep the quotes and replace Account with whichever Parent Object you'd like to display fields for. This should natch the Object you created a field set for.
- For the section labeled 2, replace Account with the Parent Object you'd like to display.
- For the section labeled 3, replace AG_Account_Detail_Related_List with the name of the field set

6. Click Save.

Create a Wrapper Visualforce Page

1. In Salesforce, click Setup.
2. Under Quick Find search for Visualforce Pages and select Visualforce Pages.
3. Click New.
4. Enter a Name and Label. The Label you use will be the name of the Reading Pane tab.

5. Paste the code from CHildVFPage.



SETUP
Visualforce Pages

Visualforce Page
AG_OpportunityAccount_RPTab

Help for this Page

Page Detail

Label: Account Details
Namespace Prefix:
Require CSRF protection on GET requests: □
Last Modified By: Amy Reams, 11/10/2020, 12:10 PM
Name: AG_OpportunityAccount_RPTab
Available for Lightning Experience, Lightning Communities, and the mobile app: ✓
Description: AG_Account; AG_Opportunity
Used to show Related Account Details in any Opportunity ActionGrid Reading Pane
Created By: Amy Reams, 11/10/2020, 12:10 PM

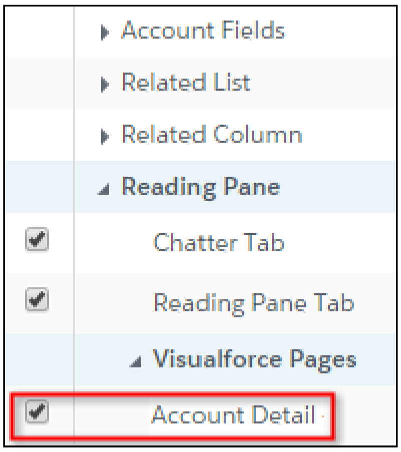
```
<apex:page standardController="Opportunity" showHeader="false" docType="html-4.01-strict">
<apex:iframe height="1100px" src="/apex/AG_Account_RPTab?id={!Opportunity.AccountId}" rendered="{!Opportunity.AccountId!=null}"/>
</apex:page>
```

- For the section labeled 1, replace Opportunity with the name of the child object you want to display your field page on.
- For the section labeled 2, replace AG_Account_RPTab with the API Name of your Parent Visualforce Page. Keep the apex/ on the front.
- For the section labeled 3, replace Opportunity.AccountId with the Lookup Field relationship from the Child Object to the Parent Object.
- For the section labeled 4, replace the Opportunity.AccountID with the same Lookup Field Relationship. This is handling for Opportunities that have no Account associated.

Load the Visualforce Page as a Reading Pane Tab

1. Click Conga Grid Explorer.
2. Select the object for which you created the Visualforce page.
3. Click Reading Pane.
4. In the row of tabs at the top of the Reading Pane, click the tab with the plus sign. This tab is at the end of the list; you may need to scroll through the tabs using the arrow icon.
5. Click the Reading Pane drop-down list.
6. Click the Visualforce Pages drop-down list.

7. Click the checkbox next to the name of your Visualforce page, for example:



- ▸ Account Fields
- ▸ Related List
- ▸ Related Column
- ◢ **Reading Pane**
  - ☑ Chatter Tab
  - ☑ Reading Pane Tab
  - ◢ **Visualforce Pages**
    - ☑ Account Detail

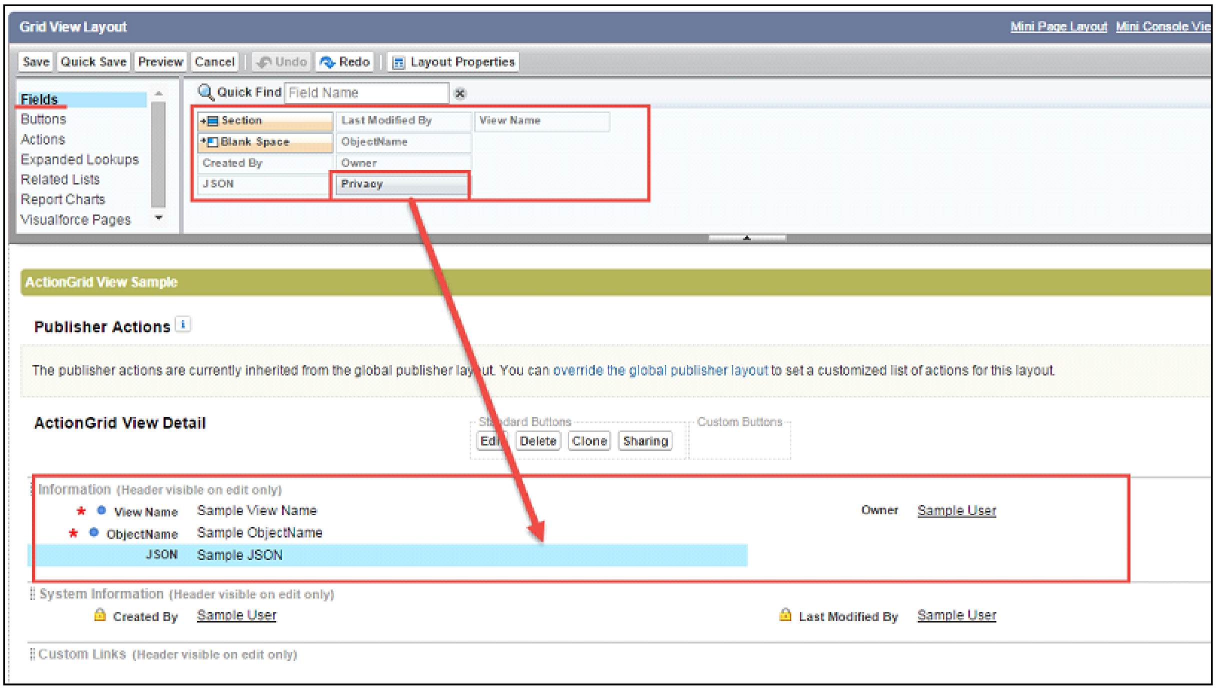
8. Click Save in the Conga Grid toolbar. This saves your reading pane tabs so they will be available in a future session.

Update Page Layouts

The page layout must expose all available fields in order to work properly. This only applies to Professional Edition since custom profiles are not supported in this edition as they are with Enterprise Edition and above. Follow these steps:

1. In Salesforce, click Setup.
2. In the App Setup section, click Create Objects.
3. Click the Grid View label.
4. Scroll down to the Page Layouts section and click Edit next to Grid View Layout.

5. In the Fields section, add every listed field to the page layout. If any fields are still in the section, drag and drop them into the Grid View Detail section.



The publisher actions are currently inherited from the global publisher layout. You can override the global publisher layout to set a customized list of actions for this layout.

View Name: Sample View Name
ObjectName: Sample ObjectName
JSON: Sample JSON
Owner: Sample User
Created By: Sample User
Last Modified By: Sample User

6. Save the new page layout.

Next, you need to override the Actions for View and Edit using the Visualforce pages provided by Grid. This is because you just modified the page layout for Grid in the previous steps, and the View and Edit states are not out-of-the-box layouts. This is necessary to support Salesforce 1 accessibility. The following steps describe how to modify the View and Edit buttons so that the correct screen appears when they are pressed.

1. Under the App Setup section in Salesforce Setup, click Create > Objects.
2. Scroll down to the Buttons, Links, and Actions section and click Edit next to the Edit label.
3. In Override Properties, click Visualforce Page and set it to Grid_View_Edit [CRMC_PP__Grid_View_Edit].
4. Click Save.
5. In the Buttons, Links, and Actions section, click Edit next to the View label.
6. In Override Properties, click Visualforce Page and set it to Grid_View [CRMC_PP__Grid_View].
7. Click Save.

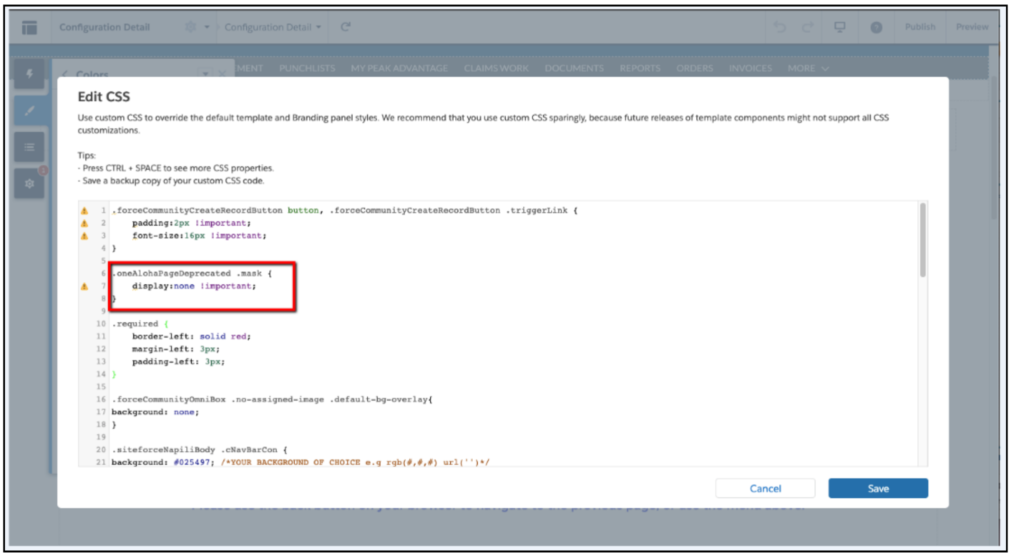
Viewing a Grid Related List in a Community

When Community users view a Grid Related List on a Salesforce page layout, Salesforce converts the Grid into a link. When a user clicks on this link they will see a Salesforce error (the error messages vary). To resolve this issue, please follow the following steps:

1. In the Community Builder, click on the Theme option
2. Click on the dropdown arrow next to Colors
3. Click Edit CSS
4. In the CSS editor, insert the following Class:

```
.oneAlohaPageDeprecated .mask {
        display:none !important;
}
```

Example Edit CSS Page:



Edit CSS

Use custom CSS to override the default template and Branding panel styles. We recommend that you use custom CSS sparingly, because future releases of template components might not support all CSS customizations.

Tips:
- Press CTRL + SPACE to see more CSS properties.
- Save a backup copy of your custom CSS code.

```
.forceCommunityCreateRecordButton button, .forceCommunityCreateRecordButton .triggerLink {
    padding:2px !important;
    font-size:16px !important;
}

.oneAlohaPageDeprecated .mask {
    display:none !important;
}

.required {
    border-left: solid red;
    margin-left: 3px;
    padding-left: 3px;
}

.forceCommunityOmniBox .no-assigned-image .default-bg-overlay{
background: none;
}

.siteforceNapiliBody .cNavBarCon {
background: #025497; /*YOUR BACKGROUND OF CHOICE e.g rgb(#,#,#) url('')*/
```

[/task/taskbody/result {""}) The Related List Grid will now no longer be treated as a link by Salesforce in your Community. (result]

Code Examples for Custom Actions in Conga Grid

Use these code examples to create custom actions in Conga Grid℠. The custom actions are automatically added to the Actions menu, providing quick access to common tasks.\

Important

Due to the ever-changing complex nature of programming languages, Conga does not support custom code or changes to code that include Visualforce or APEX elements.

- Manage Records in Conga Grid
- Accounting - Accounts Payable in Conga Grid
- Accounting - Billing in Conga Grid
- Accounting - Miscellaneous in Conga Grid

Manage Records in Conga Grid

Use these code examples to create custom actions for managing records in Conga Grid℠.

- Add Case Comments from Cases
- Add template
- Attach File
- Conga Query Code Example
- Convert Leads
- Find and Replace E-mail Domain Strings
- Hypernavigate Selected Cases from Cases

Add Case Comments from Cases

Custom Action Name: Batch_Add_Case_Comment

Objects affected: Case and Case Comment

Description: This custom action adds a case comment from the Case object. Use this on the Case object level. You can modify the fields that are set on the Case Comment by adding and removing fields from the *BATCHADD_CASECOMMENT* field set on the Case object.

Use Cases: Batch add the same comment to a group of selected cases to track the status of a case or to record new information on a group of cases.

Steps: Follow these steps to create the custom action:

1. Click Conga Grid Setup.
2. Select the Cases tab.
3. Select the Features tab on the Cases object.
4. Create a field set on the Case object with a name of BATCHADD_CASECOMMENT
5. Add the Case's "Description" field to that field set
6. Click New Custom Action. The Custom Action screen appears.

7. Delete the sample code from the Javascript field.
8. Copy and paste the following code into the Javascript field.

```
// Load in a CRMC library for entry window (note these are undocumented
and unsupported for now)
crmc.require([ "KendoEntry" , "KendoPopup" , "ListButton" ],
function(prompt, popup, navigate) {

/**
   * @author CRMCulture
   * @description Basic batch create example
*/

/**
   * "ITEM_ID" Is the ID that uniquely identifies our Action item.
   * "CONTEXT_OBJECT" Is the context object that your action is based
on.
   * "CREATE_OBJECT" Is the name of the object that you are batch
creating and/or updating.
   * "SINGULAR_NAME" Is the singular label text of CREATE_OBJECT.
   * "PLURAL_NAME" Is the plural version of the SINGULAR_NAME.
*/
var ITEM_ID = "Batch_Add_Case_Comment" ;
var CONTEXT_OBJECT = "Case" ;
var CREATE_OBJECT = "CaseComment" ;
var SINGULAR_NAME = "Case Comment" ;
var PLURAL_NAME = "Case Comments" ;

/**
   * "FIELD_SET_NAME" Is the name of a field set based on the object
that you are creating.
       This is the template of fields used when creating the modal form
window.
   * "DEFAULT_FORM_VALUES" Is a list of Field Names and values used to
populate the modal form fields.
       Each line item must be in the format of "Field Name":
"Value" (include comma if not only or last item).
*/
var FIELD_SET_NAME = "BATCHADD_CASECOMMENT" ;
var DEFAULT_FORM_VALUES = {

}
```

```
/** Language strings.*/
var TEXT_BATCH_ADD_OBJECT =  "Batch Add"  + PLURAL_NAME;
var TEXT_ERROR =  "There was an error: " ;
var TEXT_MESSAGE =  "Added {0} "  + PLURAL_NAME +  ", would you like to
ActionGrid the results?" ;
var TEXT_BATCH_ADD_RESULTS =  "Batch Add Results" ;
var TEXT_YES =  "Yes" ;
var TEXT_NO =  "No" ;

/**
   * "SET_CUSTOM_RECORD_VALUES" Is a function that sets the specified
values "Under the hood" rather than by a form. Please note that you do
not have access to fields that are not currently loaded into the grid.
   * @row
   * @record
*/
var SET_CUSTOM_RECORD_VALUES = function(SELECTED_ROW, NEW_RECORD){
     /**Required, set the Primary ID of each row to the Foreign Key
relationship.*/
    NEW_RECORD.ParentId = SELECTED_ROW.Id;

}

//////////////////////////////////////
/**
   * Custom action core code below.
*/
//////////////////////////////////////

   // Define a custom action for batch adding
  crmc.addCustomAction({
     // Uniquely identifies our Action item
     "itemID" : ITEM_ID,
     // Indicates this item should appear under batch add menu
     "isBatchAddItem" :  true ,
     "isAvailable" : function (context) {
       // This function is called before the action item is displayed
and returns a boolean if the item should be displayed
       // By default determine availability based on Feature Security
for this action
      var isEnabled =  this .featureSecurity.getSetting(context.objectD
escribe.name,  this .itemID) !==  false ;
```

```
        // Only allow batch adding from Accounts object for now
      var isAccessible = context.objectDescribe.name == CONTEXT_OBJECT;

      var multipleSelected = context.selectedRows &&
context.selectedRows.length >  0 ;
       return  isAccessible && isEnabled && multipleSelected;
      },
       "getLabel" : function (context) {
         // This function returns the display label of the action item
and is calld before the item is shown
         // Note the use of kendo ui library can be used in any actions

         return  kendo.format(PLURAL_NAME +  " ({0}) " ,
context.selectedRows.length);
       },
        "click" : function (context) {
         var records = [];
         var row_Ids = [];
          // Show a prompt that contains the fields in FIELDSETNAME

          // modified to pass in CONTEXT OBJECT because Case Comments
doesn't support Field Sets
         prompt.fieldSetEntry(TEXT_BATCH_ADD_OBJECT, CONTEXT_OBJECT ,
FIELD_SET_NAME, DEFAULT_FORM_VALUES, function(values) {
            // For every selected row
           context.selectedRows.map(function(row) {
               var record =  new  sforce.SObject(CREATE_OBJECT);
               SET_CUSTOM_RECORD_VALUES(row, record);
               row_Ids.push(row.Id);
                // Set values specified in prompt
                for  (var field in values) {
                                 // Hardcoding the field name because
we're only prompting for a single field but using the
                                 // Case objects field set so the
fields don't match
                 record.CommentBody = values[field].value;
               }
               records.push(record);
            });
            var onfailure = function(error) {
              alert(TEXT_ERROR + (error.message || error.faultstring));
            };
             // Insert opportunities using sforce ajax toolkit
            sforce.connection.create(records, {
               onSuccess: function(results) {
                  // Get the new record Ids
                 var recordIds = [];
                 $.each(results, function(i, row) {
                    if  (row.errors) {
                      // Handle first error if any
                     onfailure(row.errors);
                      return ;
                    }
                    recordIds.push(row.id);
                    });  if  (results.length == recordIds.length) {
                            // Prompt to navigate the user to the
results
                           var message = kendo.format(TEXT_MESSAGE,
results.length);
                           var buttons = [{
                             label: TEXT_YES,
                             click: function() {
                                // Navigate to a list of just these
recordIds
                               window.open(kendo.format( "/apex/
CRMC_PP__crmc_grid?object={0}&Ids={1}" , CONTEXT_OBJECT,
row_Ids.join()));
                              }
                             },
                             {
                              label: TEXT_NO
                             }];

popup.popupWithButtons(TEXT_BATCH_ADD_RESULTS, message, buttons);
                            }
                           },
                           onFailure: onfailure
                         });
                       });
                      }
                     });
                   });
```

9. Copy the itemID value and paste it as the Action Name.
10. Click OK.

Add template

The Add template custom action adds a new option under the Batch Add actions menu. The action is set to batch add an Opportunity to the selected accounts. You can modify this action by changing the appropriate variables in the code example (following the code comments). For example, change the following variables to the appropriate values for the Contact object, to batch add a Case to selected accounts:

CREATE_OBJECTSINGULAR_NAMEPLURAL_NAME

Then create a field set on the Contact object called FIELDSETBATCHADD with the fields that you want to format upon creation. The required fields need to be included in this field set.

```
crmc.require(["KendoEntry", "KendoPopup", "ListButton", "sfdc"],
function(prompt, popup, navigate, sfdc) { /** * @author CRMCulture *
@description Basic batch create example */ /** * "ITEM_ID" Is the ID that
uniquely identifies our Action item. * "CONTEXT_OBJECT" Is the context
object that your action is based on. * "CREATE_OBJECT" Is the name of the
object that you are batch creating and/or updating. * "SINGULAR_NAME" Is the
singular label text of CREATE_OBJECT. * "PLURAL_NAME" Is the plural version
of the SINGULAR_NAME. */ var ITEM_ID = "Batch_Add_Example"; var
CONTEXT_OBJECT = "Account"; var CREATE_OBJECT = "Opportunity"; var
SINGULAR_NAME = "Opportunity"; var PLURAL_NAME = "Opportunities"; /** *
"FIELD_SET_NAME" Is the name of a field set based on the object that you are
creating. This is the template of fields used when creating the modal form
window. * "DEFAULT_FORM_VALUES" Is a list of Field Names and values used to
populate the modal form fields. Each line item must be in the format of
"Field Name": "Value" (include comma if not only or last item). */ var
FIELD_SET_NAME = "FIELDSETBATCHADD"; var DEFAULT_FORM_VALUES = { /
**Example*/ //Probability: "60", //Amount: "0" } /** Language strings.*/ var
TEXT_BATCH_ADD_OBJECT = "Batch Add" + PLURAL_NAME; var TEXT_ERROR = "There
was an error: "; var TEXT_MESSAGE = "Added {0} " + PLURAL_NAME + ", would
you like to Conga Grid the results?"; var TEXT_BATCH_ADD_RESULTS = "Batch
Add Results"; var TEXT_YES = "Yes"; var TEXT_NO = "No"; /** *
"SET_CUSTOM_RECORD_VALUES" Is a function that sets the specified values
"Under the hood" rather than by a form. Please note that you do not have
access to fields that are not currently loaded into the grid. * @row *
@record */ var SET_CUSTOM_RECORD_VALUES = function(SELECTED_ROW, NEW_RECORD)
{ /**Required, set the Primary ID of each row to the Foreign Key
relationship.*/ NEW_RECORD.AccountId = SELECTED_ROW.Id; /** Set other
default values below.*/ //Example// NEW_RECORD.ContactId =
SELECTED_ROW.ContactId; //Example// NEW_RECORD.Comments =
SELECTED_ROW.Description; } //////////////////////////////////////// /** *
Custom action core code below. */ //////////////////////////////////////// //
Define a custom action for batch adding crmc.addCustomAction({ // Uniquely
identifies our Action item "itemID": ITEM_ID, // Indicates this item should
appear under batch add menu "isBatchAddItem": true, "isAvailable": function
(context) { // This function is called before the action item is displayed
and returns a boolean if the item should be displayed // By default
determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for
now var isAccessible = context.objectDescribe.name == CONTEXT_OBJECT; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0;
return isAccessible && isEnabled && multipleSelected; }, "getLabel":
function (context) { // This function returns the display label of the
action item and is calld before the item is shown // Note the use of kendo
ui library can be used in any actions return kendo.format(PLURAL_NAME +
" ({0}) ", context.selectedRows.length); }, "click": function (context)
{ var records = []; // Show a prompt that contains the fields in
FIELDSETNAME prompt.fieldSetEntry(TEXT_BATCH_ADD_OBJECT, CREATE_OBJECT,
FIELD_SET_NAME, DEFAULT_FORM_VALUES, function(values) { var recordIds = [];
var onfailure = function(error) { alert(TEXT_ERROR + (error.message ||
error.faultstring)); }; var onsuccess = function(results){ $.each(results,
function(i, row) { if (row.errors) { // Handle first error if any
onfailure(row.errors); return; } recordIds.push(row.id); }); if
(results.length == recordIds.length) { // Prompt to navigate the user to the
results var message = kendo.format(TEXT_MESSAGE, results.length); var
buttons = [{ label: TEXT_YES, click: function() { // Navigate to a list of
just these recordIds window.open(kendo.format("/apex/CRMC_PP__crmc_grid?
object={0}&Ids={1}", CREATE_OBJECT, recordIds.join())); } }, { label:
TEXT_NO }]; popup.popupWithButtons(TEXT_BATCH_ADD_RESULTS, message,
buttons); } } var process = function(){ sforce.connection.create(records,
{ onSuccess: onsuccess, onFailure: onfailure }); } // For every selected
row. // Set custom values. // Set prompt values.
context.selectedRows.map(function(row) { var record = new
sforce.SObject(CREATE_OBJECT); SET_CUSTOM_RECORD_VALUES(row, record); for
(var field in values) { record[field] = values[field].value; }
records.push(record); }); if(context.selectedRows.length > 200)
{ sfdc.batchInsert(records, onsuccess) } else { process(); } }); } });});
```

Attach File

Custom Action Name

AG_AttachFile

Objects affected

Any objects that can have Attachments.

Description

This custom action adds an Attachment from any object that can have them. Use this on the level of the object you wish to attach something to. This will open a new window for each selected record, where you can add a Salesforce attachment using the native Salesforce attachment mechanisms.

> **Warning**
>
> This will show up on the Action Menu for all objects that allow attachments.

Use Cases

This is currently the main method for adding an attachment to a group of records via the Conga Grid. This will open a new window/browser tab for each selected record, allowing a different attachment to be added to each selected record.

Steps

```
crmc.require(["KendoEntry", "KendoPopup", "ListButton", "sfdc"],function(prompt, popup,
navigate, sfdc) { crmc.addCustomAction({ "itemID": "AG_AttachFile", "isAvailable": function
(context) { var multipleSelected = context.selectedRows && context.selectedRows.length > 0
&& context.selectedRows.length <= 5; if (multipleSelected &&
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==false){ var
children = context.objectDescribe.childRelationships; for (var i = 0; i < children.length; i++) { if
(children[i].childSObject == 'Attachment'){ return true; } }; } return false; }, "getLabel":
function (context) { return "Attach File"; }, "createSubmenuItems": function (context) { return
[]; }, "click": function (context) { var ids = []; context.selectedRows.map(function(row)
{ ids.push(row.Id); });context.objectDescribe.name+" WHERE Id IN('" + ids.join("','") + "')"); for (var
i = 0; i < data.length; i++) { var row = data[i]; if(row.Name != undefined && row.Name != null)
{ window.open("/p/attach/NoteAttach?pid=" +
row.Id+"&parentname="+encodeURIComponent(row.Name)); } else{ window.open("/p/attach/
NoteAttach?pid="+row.Id); } }; } });});
```

Conga Query Code Example

Normally when you run the Conga > Batch Invoice action on multiple records (see Conga Batch Solutions - Multiple Rows), Conga Batch creates separate output documents. The custom action code below will add a Conga Query item to the Action menu. It will pass the records into a single Conga Composer solution so that you can merge the records into one document.

```
var templateId = 'a2to0000000YX8d';var queryId = 'a2lo0000000u0pl';var label = 'Conga
Query';crmc.addCustomAction({ "itemID": "CongaQuery", fieldName: '', "isAvailable": function
(context) { // This function is called before the action item is displayed and returns a
boolean if the item should be displayed // By default determine availability based on
Feature Security for this action return
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; },
"getLabel": function (context) { // This function returns the display label of the action item
and is called before the item is shown return label; }, "createSubmenuItems": function
(context) { // If this function returns additional action item objects, they will appear as
submenu items return []; }, "click": function (context) { if (context.selectedRows.length > 0)
{ var ids = ''; Object.each(context.selectedRows, function(item) { if (ids.length > 0) ids += "'|'";
ids += item.Id; }); var url = "https://composer.congamerge.com" + "?sessionId=" +
sforce.connection.sessionId + "&ServerUrl=" +
context.actionGrid.settings.Partner_Server_URL_290 + "&Id=" +
sforce.connection.getUserInfo().userId + "&templateId=" + templateId + "&QueryId="+ queryId
+"?pv0='" + ids +"'"; window.open(url, 'Conga', 'width=700,height=450,menubar=0'); } else
{ kendoPopup.popup('Please select records', 'Select at least one record to send to Conga.');
return; } },});
```

Convert Leads

Custom Action Name

AG_Batch_LeadTo

Objects affected

Lead and possible Accounts, Contacts, Opportunities (depending on the conversion)

Description

Use this custom action to convert selected Leads into Opportunities or Contacts in your organization. Use it from the Leads level. Please see the code comments for specific conversion behavior.

Use Cases

Convert a single Lead or a group of Leads at once.

Convert a Single Lead

Follow these steps to convert a Lead to an Account, Contact, Opportunity, or Follow-Up Task.

1. Click Conga Grid Explorer.
2. Click Lead in the object list.
3. Select the check box of the record you want to convert.
4. Click Actions and select Convert Lead > Convert Lead (Form).
5. This opens the selected record in a Salesforce Convert Lead page where you can fill in the details about the selected lead. Refer to the Salesforce documentation for instructions on the Lead conversion process.Batch Convert LeadsFollow these steps to convert multiple Leads to Opportunities:
6. Click Conga Grid Explorer.
7. Click Lead in the object list.
8. Select one or more records to convert.
9. Click Actions and select Convert Lead > Batch Convert. The Batch Convert Options screen appears.
10. Click Create Opportunities.
11. Click OK. The results open in a new tab.

> **Warning**
>
> The use of JavaScript in Conga solutions is considered an advanced method, requiring JavaScript knowledge, and is not supported by Conga or the Conga support team. You are welcome to use this feature at your own risk

Code Example

```
/**
    * @author Conga Grid
    * @version 1.07
    * @description Batch convert leads.
    *
    * @id AG_Batch_LeadTo   Grid custom action id that is added to the name
field.
```

```
    * @param kendoEntry     Grid helper for Kendo UI modal entry forms.
    * @param kendoPopup     Grid helper for Kendo UI model message windows.
    * @param sfdc           Grid helper for Salesforce REST API's
*/
/** Notes
    Account - Match on the name.
            - Bring over all child data.
            - Contact - Match on email address.
            -  If the Contact is a dupe then we don't need the account.
-ISSUE
 Note: account is required if merging a contact otherwise you will get a new
account.
 I built a lookup to pull the parent account of the matched contact. This
will merge the
 lead and its child data with the contact and its parent account. Warning, if
that lead
 was somehow matched with a different account by name the contact parent
lookup will
 override that value, because a lead merging  with a contact must be paired,
and merged, with
 its parent account.
    Opportunity - Convert if checked.
*/

crmc.require([ "KendoEntry" ,  "KendoPopup" ,  "sfdc" ],
function(kendoEntry, kendoPopup, sfdc) {
    var LeadToForm =         {
            "isAvailable" : function (context) {
                var isEnabled =  true ;
                var isProject = context.objectDescribe.name ==  "Lead" ;
                var multipleSelected = context.selectedRows &&
context.selectedRows.length >  0 ;
                 return  isProject && isEnabled && multipleSelected;
            },
             "getLabel" : function (context) {
                 return   "Convert Lead (Form)" ;
            },
             "createSubmenuItems" : function (context) {
                 return  [];
            },
             "click" : function (context) {
                 if (context.selectedRows.length >  5 ){
                    var buttons = [{
                        label:  "Yes" ,
                        click: function() {
                            processURL();
                        }
                    },
                    {
                        label:  "No"
                    }];
                    kendoPopup.popupWithButtons( "WARNING" ,  "5 records are
recommended because each item will show in a new tab. Do you want to
continue?" , buttons);
                } else {
                    processURL();
                }
                function processURL(){
                    var ids = [];
                    context.selectedRows.map(function(row) {
                        ids.push(row.Id);
                    });
                     for  (var i =  0 ; i < ids.length; i++) {
                        window.open( "/lead/leadconvert.jsp?id="  + ids[i]);
                    };
                }
            }
    };
    var LeadToBatchConvert =
    {
         "isAvailable" : function (context) {
            var isEnabled =  true ;
            var isProject = context.objectDescribe.name ==  "Lead" ;
            var multipleSelected = context.selectedRows &&
context.selectedRows.length >  0  && context.selectedRows.length <=  30 ;
             return  isProject && isEnabled && multipleSelected;
        },
         "getLabel" : function (context) {
             return   "Batch Convert" ;
        },
         "createSubmenuItems" : function (context) {
             return  [];
        },
         "click" : function (context) {
             //Created Ids.
            var accountIds = [];
            var contactIds = [];
            var opportunityIds = [];
             //Lookup data.
            var records = [];
            var success =  0 ;
            var CANNOT_UPDATE_CONVERTED_LEAD =  0 ;
            var failCount =  0 ;
            var failed = [ "<ul>" ];
            var errorTrue =  false ;
             //Fields
            var doNotCreateOpportunity =  true ;
            function ConvertLead(records){
                sforce.connection.convertLead(records,{
                    onSuccess: function(results){

//results test data
context.kendoGrid._progress && context.kendoGrid._progress( false );
                         for  (var i =  0 ; i < results.length; i++) {
                             if (results[i].success && results[i].success ==
  "true" ){
                                success++;
                                results[i].accountId !=  null  ?
accountIds.push(results[i].accountId): function(){};
                                results[i].contactId !=  null  ?
contactIds.push(results[i].contactId): function(){};
                                results[i].opportunityId !=  null  ?
opportunityIds.push(results[i].opportunityId): function(){};
                            }
                             else   if (results[i].errors.statusCode ==
"CANNOT_UPDATE_CONVERTED_LEAD" ){
                                CANNOT_UPDATE_CONVERTED_LEAD++;
                            }
                             else {
                                errorTrue =  true ;
                                failCount++;
                                 if (results[i].errors != undefined &&
results[i].errors.message != undefined){
                                    failed.push( "<li>" +results[i].errors.m
essage+ "</li>" );
                                }
                                 else {
                                    failed.push( "<li>" +results[i]+ "</li>" )
;
                                }
                            }
                        };
                        failed.push( "</ul>" );

                         // Convert lead debug email example example. Remove
code comments and add preferred email addresses.
                         // var body = "Errors: " + errorTrue + " --
Records: " + records.toString() + " -- Results: " + results.toString();
                         // var singleRequest = new
sforce.SingleEmailMessage();
                         // singleRequest.replyTo = "[Please add Reply To
email address here]";
                         // singleRequest.subject = "Convert Lead Failures";
                         // singleRequest.plainTextBody = body;
                         // singleRequest.toAddresses = "[Please add To
email address here]";
                         // var sendMailRes =
sforce.connection.sendEmail([singleRequest]);

                        var buttons = [
                            {
                                label:  "Yes" ,
                                click: function() {
                                    window.open(kendo.format( "/apex/
CRMC_PP__crmc_grid?object={0}&Ids={1}" ,  "Account" , accountIds.join()));
                                    window.open(kendo.format( "/apex/
CRMC_PP__crmc_grid?object={0}&Ids={1}" ,  "Contact" , contactIds.join()));
                                     if (doNotCreateOpportunity ==  false ){
                                        window.open(kendo.format( "/apex/
CRMC_PP__crmc_grid?object={0}&Ids={1}" ,  "Opportunity" ,
opportunityIds.join()));
                                    }
                                     if (failCount> 0 ){

kendoPopup.popupWithButtons( "Errors" , failed.join( "" ), [{label:  "Ok" }]
, {width:  900 });
                                    }
                                }
                            },{
                                label:  "No" ,
                                click: function(){
                                     if (failCount> 0 ){
```

```
kendoPopup.popupWithButtons( "Errors" , failed.join( "" ), [{label:  "Ok" }]
, {width:  900 });
                                    }
                                }
                            }
                        ];
                         if (context.selectedRows.length == success){
                            var message = (doNotCreateOpportunity ?  "Review
the new/merged Account and Contact records?"  :  "Review the new/merged
Account, Contact, and Opportunity records?" );
                            kendoPopup.popupWithButtons( "Success!" ,
message, buttons);
                        }
                         else {
                             //Check if any records were converted
                            var anySuccess = (accountIds.length >  0  ||
contactIds.length >  0  || opportunityIds.length >  0 );
                            var label =  "" ;
                            var message = [
                                 "<ul>" ,
                                 "<li> Total selected: "  +
context.selectedRows.length +  "</li>" ,
                                 "<li> Already converted: "  +
CANNOT_UPDATE_CONVERTED_LEAD +  "</li>" ,
                                 "<li> New/Merged Accounts: "  +
accountIds.length +  "</li>" ,
                                 "<li> New/Merged Contacts: "  +
contactIds.length +  "</li>" ,
                                 "<li> New Opportunities: "  +
opportunityIds.length +  "</li>" ,
                                 "<li> Errors: "  + failCount +  "</li>" ,
                                (anySuccess ?  ""  :  "<li> Click 'To
Console' for more information </li>" ),
                                 "</ul>"
                            ];
                             if (anySuccess){
                                label =  "Warning: Not all records were
processed." ;
                                message.push((doNotCreateOpportunity ?
 "Review the new/merged Account and Contact records?"  :  "Review the new/
merged Account, Contact, and Opportunity records?" ) +  " <br/>" );
```

```
                                kendoPopup.popupWithButtons(label,
message.join( "" ), buttons, {width:  350 });
                            }  else  {
                                var buttons = [
                                    {
                                        label:  "Ok" ,
                                        click: function(){
                                             if (failCount> 0 ){
                                                kendoPopup.popupWithButtons( "
Errors" , failed.join( "" ), [{label:  "Ok" }], {width:  900 });
                                            }
                                        }
                                    }
                                ];
                                label =  "Warning: No records were
processed." ;
                                kendoPopup.popupWithButtons(label,
message.join( "" ), buttons, {width:  300 });
                            }
                        }
                        context.actionGrid.refresh();
                    },
                    onFailure: function(error) {
                        context.kendoGrid._progress &&
context.kendoGrid._progress( false );"
                        console.log( "Error: "  + error);
                        alert( "Error: "  + error);
                    }
                });
            }
            var convertedList = sfdc.query( "SELECT MasterLabel FROM
LeadStatus WHERE IsConverted = true" );
             if (convertedList.length >  1 ){
                var objectDescribe = sfdc.getSObjectDescribe( "Lead" );
                var picklistValues = [];
                 //Loop through the Lead fields.
                Object.each(objectDescribe.fields, function(field){
                     //Find the Status field.
                     if  (field.name ==  "Status" ){
                         //Loop through the LeadStatus Converted fields.
                         for  (var i =  0 ; i < convertedList.length; i++) {
                             //Loop through the Lead picklist values for
each LeadStatus.
```

```
                              for  (var j =  0 ; j <
field.picklistValues.length; j++) {
                                  //Match LeadStatus MasterLabel with Lead
Status picklist value.
                                  if (convertedList[i].MasterLabel ===
field.picklistValues[j].value){
                                      //Populate form droplist with matched
values.

picklistValues.push(field.picklistValues[j]);
                                }
                            };
                        };
                    }
                });
            }
            function GetFields(){
                 if (convertedList.length >  1 ){
                     return  [
                        {referenceTo: "User" , name:  "OwnerId" , label:
 "Owner" , type:  "reference" , required:  false },
                        {name:  "Status" , label:  'Status' , type:
 'picklist' , values: picklistValues, required:  true },
                        {name:  "createOpportunity" , label:  'Create
Opportunities' , type:  'boolean' , value:  false },
                        {name:  "sendNotificationEmail" , label:  'Notify
Owner' , type:  'boolean' , value:  false },
                        {name:  "overwriteLeadSource" , label:  'Overwrite
Lead Source' , type:  'boolean' , value:  false },
                    ];
               }
                 else  {
                     return  [
                        {referenceTo: "User" , name:  "OwnerId" , label:
 "Owner" , type:  "reference" , required:  false },
                        {name:  "createOpportunity" , label:  'Create
Opportunities' , type:  'boolean' , value:  false },
                        {name:  "sendNotificationEmail" , label:  'Notify
Owner' , type:  'boolean' , value:  false },
                        {name:  "overwriteLeadSource" , label:  'Overwrite
Lead Source' , type:  'boolean' , value:  false },
                    ]
                }
            }
            kendoEntry.entry( "Batch Convert Options" ,
                GetFields(),
                {
                    objectName: "Lead" ,
                    resizable:  true ,
                    autoSizeFields:  true ,
                    width:  '25%' ,
                    minWidth:  '410px'
                },
                 null ,
                function(selectedValues) {
                     try {
                        context.kendoGrid._progress &&
context.kendoGrid._progress( true );
                        doNotCreateOpportunity = !
selectedValues[ "createOpportunity" ].value;
                         //Query arrays.
                        var ids = [];
                        var leadCompanies = [];
                        var leadEmails = [];
                         //Convert Object.
                        var mapConvert = {};
                        var convert = [];
                        function DataCheck(string){
                             return  string.replace( "'" ,  "\\'" );
                        }
                         //Get a list of selected record Id's.
                        context.selectedRows.map(function(row) {
                            ids.push(row.id);
                        });
                         //Query for the Lead Company and email address.
                        var data = sfdc.query( "SELECT Id, Company, Email
FROM Lead WHERE Id IN('"  + ids.join( "','" ) +  "')" );
                         //Map lead names and build query arrays.
                         for  (var i =  0 ; i < data.length; i++) {
                            var record = data[i];
                             //Query arrays.
                            leadCompanies.push(DataCheck(record.Company));
                             if (record.Email !=  ""  && record.Email !=
 null  && record.Email != undefined){
                                leadEmails.push(record.Email);
                            }
```

```
                             //Create a new lead convert object mapping for
this lead.
                            mapConvert[record.Id] = {
                                leadId: record.Id,
                                Company: record.Company,
                                Email: record.Email,
                            };
                        };

                          //Match lead names with account names.
                        var accountMatch = sfdc.query( "SELECT Id, Name FROM
Account WHERE Name IN('"  + leadCompanies.join( "','" ) +  "') AND IsDeleted
= false" );
                        var joinEmails = leadEmails.join( "','" );
                         //Query only if there are contact emails to match.
                         if (joinEmails !=  '' ){
                             //Match lead emails with contact emails.
                            var contactMatch = sfdc.query( "SELECT Id,
AccountId, Email FROM Contact WHERE Email IN('"  + leadEmails.join( "','" )
+  "') AND IsDeleted = false" );
                        }
                         for  (var key in mapConvert) {
                             for  (var i =  0 ; i < accountMatch.length; i+
+) {
                                 var match = accountMatch[i];
                                 if (mapConvert[key].Company == match.Name){
                                     //Set the matching account.
                                    mapConvert[key].accountId = match.Id;
                                }
                            };
                             if (joinEmails !=  '' ){
                                 for  (var i =  0 ; i < contactMatch.length;
i++) {
                                    var match = contactMatch[i];
                                     if (mapConvert[key].Email ==
match.Email){
                                         //Set the parent account Id.
                                        mapConvert[key].accountId =
match.AccountId;
                                         //Set the matching contact.
                                        mapConvert[key].contactId =
match.Id;
                                    }
                                 };
                             }
                         }
                          for  (var key in mapConvert) {
                             var leadConvert =  new  sforce.LeadConvert();
                             leadConvert.leadId = mapConvert[key].leadId;
                             leadConvert.accountId =
mapConvert[key].accountId;
                             leadConvert.contactId =
mapConvert[key].contactId;
                             leadConvert.sendNotificationEmail =
selectedValues[ "sendNotificationEmail" ].value;
                             leadConvert.overwriteLeadSource =
selectedValues[ "overwriteLeadSource" ].value;
                              if (selectedValues.hasOwnProperty( "Status" )){
                                 leadConvert.convertedStatus =
selectedValues[ "Status" ].value;
                             }
                              else  {
                                 leadConvert.convertedStatus = convertedList[ 0
.MasterLabel;
                             }
                             leadConvert.doNotCreateOpportunity =
doNotCreateOpportunity;
                              if (selectedValues[ "OwnerId" ].value !=  "" )
leadConvert.ownerId = selectedValues[ "OwnerId" ].value;
                             convert.push(leadConvert);
                         };
                         ConvertLead(convert);
                     }
                      catch (error){
                         context.kendoGrid._progress &&
context.kendoGrid._progress( false );
                         console.log( "Error: " + error);
                         alert( "Error: "  + error)
                     }
                 }
             );
         }
     };
      crmc.addCustomAction({
         actions: [LeadToForm, LeadToBatchConvert],
          "itemID" :  "AG_Batch_LeadTo" ,
         isAvailable: function(context) {
             var available =  false ;
             Object.each( this .actions, function(subMenu){
                  if  (subMenu.isAvailable(context)){
                     available =  true ;
                 }
             });
              return  available;
         },
          "getLabel" : function (context) {
            return   "Convert Lead..." ;
         },
          "createSubmenuItems" : function (context) {
            return   this .actions;
         },
          "click" : function (context) {
         }
     });
 });
```

Find and Replace E-mail Domain Strings

Custom Action Name

AG_Generic_FindReplace

Objects affected

Any object with an email field that is editable

Description

Use this custom action to modify the domain of email fields of selected records. Use from any object with an editable email field.

Use Cases

Modify a group of selected records' email domains if they were to change companies or email services.

Steps

*/** * @author CRMCulture * @version 1.02 * @description Generic find replace for all text, textarea, phone, and email fields. Loops through each row locating the specified value and replaces it with the new value. Works in a ActionGrid with or without edit mode enabled. * * @id AG_Generic_FindReplace ActionGrid custom action id that is added to the name field. **

```
@param kendoEntry Conga Grid helper for Kendo UI modal entry forms. * @param
kendoPopup ActionGrid helper for Kendo UI model message windows. * @param sfdc
ActionGrid helper for Salesforce REST API's*/crmc.require(["KendoEntry", "KendoPopup",
"sfdc"], function(kendoEntry, kendoPopup, sfdc){ crmc.addCustomAction({ "itemID":
"AG_Generic_FindReplace", getLabel: function(context) { return "Find/Replace"; }, isAvailable:
function(context) { return true; }, click: function(context) { var fieldMeta =
context.actionGrid.fieldMeta; var rows = context.selectedRows; var recordIds = []; var
onfailure = function(error) { try{ return "<li> Status Code: " + error.statusCode + ", Message: "
+ error.message + "</li>"; } catch (err){ alert("Error: " + err); } }; var onsuccess =
function(results){ var updateResults = ["<style>div.overflow{overflow: scroll; max-height:
600px;}</style><div><div class='overflow'><ul>"]; $.each(results, function(i, row) { if
(row.errors) { updateResults.push(onfailure(row.errors)); return; } recordIds.push(row.id); }); if
(results.length == recordIds.length) { var TEXT_MESSAGE = kendo.format("{0} Record" +
(recordIds.length === 1 ? "" : "s") + " updated, would you like to Conga Grid the result" +
(recordIds.length === 1 ? "" : "s") + "?", recordIds.length); var TEXT_BATCH_ADD_RESULTS =
"Update Results"; var buttons = [{ label: "Yes", click: function() { window.open(kendo.format("/
apex/CRMC_PP__crmc_grid?object={0}&Ids={1}", context.objectDescribe.name,
recordIds.join())); } }, { label: "No", click: function() { context.actionGrid.refresh(); } }];
kendoPopup.popupWithButtons(TEXT_BATCH_ADD_RESULTS, TEXT_MESSAGE, buttons);
context.actionGrid.refresh(); } else{ updateResults.push("</ul></div></div>");
kendoPopup.popupWithButtons("Batch Update Errors", updateResults.join(""),
ErrorButtons(results), {width: 700}); } } function ErrorButtons (results){ return [ { label: "Send
To Console", click: function() { $.each(results, function(i, row) { console.log(row.errors); }); } },
{ label: "Ok" } ]; } function ButtonOk (){ return [ { label: "Ok" } ]; } function GetColumns(){ var
columns = []; for (var i = 0; i < fieldMeta.length; i++) { if(IsType(fieldMeta[i].type) &&
fieldMeta[i].name.indexOf(".") === -1){ columns.push({active: "true", defaultValue: "false",
label: fieldMeta[i].displayName, value: fieldMeta[i].name}); } }; return columns; }; function
IsType(type){ switch (type) { case "STRING": return true; case "TEXTAREA": return true; case
"PHONE": return true; case "EMAIL": return true; default: return false; } } function
BatchUpdate(sObject, resultFunction) { if (!(sObject instanceof Array)) sObject = [sObject];
var limit = 200; var i; var resultsList = []; var numBatches = sObject.length / 200; var
numCompletedBatches = 0; var handlerFunction = function(results) { resultsList =
resultsList.concat(results); numCompletedBatches++; if (numCompletedBatches >=
numBatches) { if (typeof(resultFunction) === "function") { resultFunction(resultsList); } } }; for
(i = 0; i < sObject.length; i += limit) { sforce.connection.update(sObject.slice(i, i + limit),
{ onSuccess: handlerFunction, onFailure: handlerFunction, }); } } function Replace(values)
{ var updates = []; for (var i = 0; i < rows.length; i++){ //Create a record to hold update. var
record = new sforce.SObject(context.objectDescribe.name); //Check if the user wants to
replace empty value with new value. if(rows[i][values.apiValue] === undefined &&
values.oldValue === ""){ record.Id = rows[i].Id; record[values.apiValue] = values.newValue;
updates.push(record); } else if(rows[i][values.apiValue] != undefined && values.oldValue != "")
{ var string = rows[i][values.apiValue].replace(values.oldValue, values.newValue); //Check if
there was a change. if(string != rows[i][values.apiValue]){ record.Id = rows[i].Id;
record[values.apiValue] = string; updates.push(record); } } } if(updates.length > 0)
{ if(updates.length > 200){ BatchUpdate(updates, onsuccess); } else
{ sforce.connection.update(updates, { onSuccess: onsuccess, onFailure: onfailure }); } } }
function EditableReplace(values){ for (var i = 0; i < rows.length; i++) { var ColumnIndex =
function(){ for (var columnIndex = 0; columnIndex < context.kendoGrid.columns.length;
columnIndex++) { if(context.kendoGrid.columns[columnIndex].field === values.apiValue)
return columnIndex; } } var field = context.kendoGrid.columns[i]; var cell = $
(kendo.format("{2} tr[data-uid={0}] td:not(.k-group-cell,.k-hierarchy-cell):eq({1})", rows[i].uid,
ColumnIndex(), context.actionGrid.getElementId("grid"))); //Check if the user wants to
replace empty value with new value. if(rows[i][values.apiValue] === undefined &&
values.oldValue === ""){ context.kendoGrid.editCell(cell); rows[i].set(values.apiValue,
values.newValue); context.kendoGrid.editCell(cell); } else if(rows[i][values.apiValue] !=
undefined && values.oldValue != ""){ var string = rows[i]
[values.apiValue].replace(values.oldValue, values.newValue); //Check if there was a
change. if(string != rows[i][values.apiValue]){ context.kendoGrid.editCell(cell);
rows[i].set(values.apiValue, string); context.kendoGrid.editCell(cell); } } }; } var fields =
GetColumns(); if(fields.length > 0){ kendoEntry.entry("Replace With…", [ {name: "Columns",
label: 'Column', type: "picklist", values: fields, required: true}, {name: "OldValue", label: 'Old
Value', type: "text", required: true}, {name: "NewValue", label: 'New Value', type: "text",
required: true} ], {width: 350}, null, function(selectedValues) { //Does the editable property
exist? Is ActionGrid in edit mode? var values = {apiValue: selectedValues["Columns"].value,
oldValue: selectedValues["OldValue"].value, newValue: selectedValues["NewValue"].value};
context.actionGrid.settings.hasOwnProperty("editable") &&
context.actionGrid.settings.editable ? EditableReplace(values) : Replace(values); } ); } else
{ kendoPopup.popupWithButtons("No Supported Fields", "There are no supported fields in the
view.", ButtonOk(), {width: 250}); } } });});
```

Hypernavigate Selected Cases from Cases

Custom Action Name

Batch_Navigate_Selected_ToCloseCase_From_Cases

Objects affected

None. This is purely a navigation action.

Description

This is an example of a combination drill up / drill down the action. It moves a user from sibling to sibling.

Use Cases

Navigate from a case to all the sibling cases based on the account.

Steps

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"Batch_Navigate_Selected_ToCloseCase_From_Cases", "isAvailable": function (context) { //
This function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var hasSecurity = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false var isCase = context.objectDescribe.name == ("Case"); return
hasSecurity && isCase }, "getLabel": function (context) { // This function returns the display
label of the action item and is called before the item is shown return "Close Selected"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { if
(context.selectedRows.length > 0) { var selectedIds = []; var queryParams = {object: 'Case',
viewId: 'a06o000000HQ4PeAAL'}; Object.each(context.selectedRows, function (item) { var id
= item["Id"]; if(id === undefined || id === null){ //Do nothing if empty. } else
{ selectedIds.push(id); } }); if (selectedIds.length > 200) { var postData = {Ids:
selectedIds.join()}; //sfdc.postToVFPage("c", "AG_Case_Nav_Selected", queryParams,
postData, false, '_BLANK'); sfdc.postToVFPage("CRMC_PP", "crmc_grid", queryParams,
postData, false, '_BLANK'); }else { queryParams.Ids = selectedIds.join(); //
sfdc.navigateToVFPage("c", "AG_Case_Nav_Selected", queryParams, false, '_BLANK');
sfdc.navigateToVFPage("CRMC_PP", "crmc_grid", queryParams, false, '_BLANK'); } } else
{ popup.popup('Please select records', 'Select at least one record for navigating selected
records'); } } });});
```

Email/Web/Pardot in Conga Grid

Use these code examples to create custom actions for email operations in Conga Grid℠.

Batch Email via Pardot Exchange

Custom Action Name

Batch_Send_Pardot_Emails

Objects affected

None

Description

This is an example of a custom action that integrates Conga Grid with a marketing email application.

Use Cases

Send Pardot emails from a Conga Grid. This could be really useful from a targeted list of contacts or leads.

Steps

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"Batch_Send_Pardot_Emails", "isAvailable": function (context) { // This function is called
before the action item is displayed and returns a boolean if the item should be displayed //
By default determine availability based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow this action from Contacts and Leads var isCorrectContext =
context.objectDescribe.name == "Contact" || context.objectDescribe.name == "Lead"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isCorrectContext && isEnabled && multipleSelected; }, "getLabel": function (context) { // This
function returns the display label of the action item and is called before the item is shown
return "Send Engage Email"; }, "createSubmenuItems": function (context) { // If this function
returns additional action item objects, they will appear as submenu items return []; }, "click":
function (context) { var sendToVFPage = function() { var selectedIds = []; if
(context.selectedRows.length > 0) { //Is the context Contact or Lead? Set the page to the
correct VisualForce page. var page = context.objectDescribe.name == "Contact" ?
"MicroCampaignContact" : "MicroCampaignLead"; Object.each(context.selectedRows,
function (item) { selectedIds.push({name: "ids", value: item["Id"]}); }); var parameters =
{ retURL: context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("pi", page, parameters,
selectedIds, false, "_blank"); } else { popup.popup('Please select records', 'Select at least one
record to process.'); } }; // Include static resource from managed namespace pre-creates
session and allows us to safely POST data without the redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://pi." + instance +
".visual.force.com/resource/loadingSpinner?ts=" + new Date().getTime(); //On error call page
logic because you will still have a session. script.onerror = function() { sendToVFPage(); }; //
Call page logic. script.onload = function() { sendToVFPage(); } } });});
```

Batch Post IDs to Visualforce Page

Custom Action Name

Web_Post_(PostObjectName)_From_(TableName)

Objects affected

None

Description

This custom action lets you post record IDs to a Visualforce page.

Use Cases

Pass a list of IDs to a Visualforce page to use them in a process that the Visualforce page can initiate.

Steps

```
// Load in a CRMC library for entry window (note these are undocumented and unsupported
for now)crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) {/** * @author CRMCulture
* @description Basic Salesforce POST example*//** * "ITEM_ID" Is the ID that uniquely
identifies our Action item, Paste it into the action name value. * "CONTEXT_OBJECT" Is the
context object that your action is based on, use API name. * "ACTION_LABEL" Is the lable
that will be displayed in the context menu.*/var ITEM_ID =
"Web_Post_(PostObjectName)_From_(TableName)";var CONTEXT_OBJECT = "Account";var
ACTION_LABEL = "POST Hello World";/** * @DESCRIPTION: In order to get garantee a
session for the user we must need to make a static resource call. Otherwise the first time a
user runs this action they will get a no records selected error... the resource must be of type
Javascript it's a known Salesforce bug and this is currently the most effective way to solve
the issue. * "APP_PACKAGE_NAME" Is the Package Name of the application you need to
create a session for. * "APP_NAMESPACE_PREFIX" Is the Prefix to the application you need to
create a session for. * "APP_STATIC_RESOURCE_MINE_TYPE" Is file MINE Type of the Static
Resource within the application. * "APP_STATIC_RESOURCE_NAME" Is the Name of the Static
Resource being called within the application. * "APP_APEX_CLASS_NAME" Is the apex class
name of your */var APP_PACKAGE_NAME = "Package Name";var APP_NAMESPACE_PREFIX =
"Prefix";var APP_STATIC_RESOURCE_MINE_TYPE = "text/javascript";var
APP_STATIC_RESOURCE_NAME = "Resource Name";var APP_APEX_CLASS_NAME = "Apex
Class Name";/** Language strings.*/var COULD_NOT_OBTAIN_SESSION_FOR = "Could not
obtain session for ";var IS_THE_APP_INSTALLED = " is the app installed?";var
PLEASE_SELECT_RECORDS = "Please select records";var
SELECT_AT_LEAST_ONE_RECORD_TO_POST = 'Select at least one record to
post.'////////////////////////////////////////** * Custom action core code
below.*//////////////////////////////////////// crmc.addCustomAction({ "itemID": "ITEM_ID",
"isAvailable": function (context) { // This function is called before the action item is displayed
and returns a boolean if the item should be displayed // By default determine availability
based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isAccessible =
context.objectDescribe.name == CONTEXT_OBJECT; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isAccessible && isEnabled
&& multipleSelected; }, "getLabel": function (context) { // This function returns the display
label of the action item and is called before the item is shown return "ACTION_LABEL"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { //https://
acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wrapMassAction=1&scontrolCaching=1 // Include static resource from managed namespace
pre-creates session and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = APP_STATIC_RESOURCE_MINE_TYPE;
head.appendChild(script); var domains = window.location.hostname.split("."); if
(domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; } //script.src =
"https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?
ts=" + new Date().getTime(); script.src = "https://" + APP_NAMESPACE_PREFIX + "." + instance
+ ".visual.force.com/resource/" + APP_NAMESPACE_PREFIX + "__" +
APP_STATIC_RESOURCE_NAME + "?ts=" + new Date().getTime(); script.onerror = function() { //
Couldn't load app script, assume not installed
popup.popup(COULD_NOT_OBTAIN_SESSION_FOR + APP_PACKAGE_NAME +
IS_THE_APP_INSTALLED); }; //POST script.onload = function() { //If any rows are selected.
POST with the given id's. if (context.selectedRows.length > 0) { var selectedIds = [];
Object.each(context.selectedRows, function (item) { selectedIds.push({name: "ids", value:
item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage(APP_NAMESPACE_PREFIX, APP_APEX_CLASS_NAME, parameters,
selectedIds); } else { popup.popup(PLEASE_SELECT_RECORDS,
SELECT_AT_LEAST_ONE_RECORD_TO_POST); } }; } });});
```

Batch Process URLs

Custom Action Name

createOrdersFromProjects

Objects affected

None

Description

Use this custom action to push multiple IDs into a URL.

Use Cases

Add IDs to a URL to use them in the loaded page.

Steps

```
crmc.addCustomAction({ "itemID": "createOrdersFromProjects", "isAvailable": function
(context) { // This function is called before the action item is displayed and returns a
boolean if the item should be displayed // By default determine availability based on
Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "crmc_Project__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Create Order(s)"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { var ids = [];
context.selectedRows.map(function(row) { ids.push(row.Id); }); window.open("/apex/
Project_List_Create_Order?projectRecordIds=" + ids.join('%2c')); }});
```

Follow-Up Event

```
/** * @author CRMCulture * @version 1.00 * @description BatchAdd a follow-up event from
a Task or Event. * * @id AG_BatchAdd_FollowUpEvent ActionGrid custom action id that is
added to the name field. * @param kendoEntry ActionGrid helper for Kendo UI modal entry
forms. * @param kendoPopup ActionGrid helper for Kendo UI model message windows. *
@param sfdc ActionGrid helper for Salesforce REST API's*/crmc.require(["KendoEntry",
"KendoPopup", "sfdc"], function(kendoEntry, kendoPopup, sfdc) { var CurrentObject = { object:
{Name: "", Plural: ""}, }; var IsContext = function(object){ objects = [ {Name: "Task", Plural:
"Tasks"}, {Name: "Event", Plural: "Events"} ]; for (var i = 0; i < objects.length; i++)
{ if(objects[i].Name === object){ CurrentObject.object = objects[i]; return true; } }; return
false; }; crmc.addCustomAction({ "itemID": "AG_BatchAdd_FollowUpEvent",
"isBatchAddItem": true, "isAvailable": function (context) { var isCorrectContext =
IsContext(context.objectDescribe.name) ; var multipleSelected = context.selectedRows &&
context.selectedRows.length > 0; var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false return
isCorrectContext && multipleSelected && isEnabled; }, "getLabel": function (context) { return
"Follow-Up Event"; }, "createSubmenuItems": function (context) { return []; }, "click": function
(context) { function IsPlural(){ return (context.selectedRows.length === 1 ? "" : "s"); } /**
Language strings.*/ var TEXT_BATCH_ADD_OBJECT = "Batch Add Events" var TEXT_ERROR
= "There was an error: "; var TEXT_MESSAGE = "Added {0} Event" + IsPlural() + ", review the
record" + IsPlural() + " with Preview or Browse Records."; var TEXT_BATCH_ADD_RESULTS =
"Batch Add Results"; var TEXT_YES = "Okay"; var TEXT_NO = "No"; function Process(Subject,
StartDateTime, EndDateTime, IsAllDayEvent, Description){ var ids = []; for (var i = 0; i <
context.selectedRows.length; i++) { ids.push(context.selectedRows[i].id); }; //Query for the
data that needs to be passed into new records. var results = sfdc.query("SELECT Id, WhoId,
WhatId FROM " + CurrentObject.object.Name + " WHERE Id IN('" + ids.join("','") + "')"); //Loop
through each result and pull the data. var records = []; for (var i in results) { var record =
new sforce.SObject('Event'); for (var j in ids) { if(ids[j] === results[i].Id){ record.Subject =
Subject; record.StartDateTime = StartDateTime; record.EndDateTime = EndDateTime;
record.IsAllDayEvent = IsAllDayEvent; record.Description = Description; record.WhoId =
results[i].WhoId; record.WhatId = results[i].WhatId; records.push(record); break; }; }; }; var
onfailure = function(error) { alert(TEXT_ERROR + (error.message || error.faultstring)); };
sforce.connection.create(records, { onSuccess: function(results) { // Get the new record Ids
var recordIds = []; $.each(results, function(i, row) { if (row.errors) { onfailure(row.errors);
return; } recordIds.push(row.id); }); if (results.length == recordIds.length) { var message =
kendo.format(TEXT_MESSAGE, results.length); var buttons = [{ label: TEXT_YES, click:
function() { window.open(kendo.format("/apex/CRMC_PP__crmc_grid?object={0}&Ids={1}",
'Event', recordIds.join())); } } ]; kendoPopup.popupWithButtons(TEXT_BATCH_ADD_RESULTS,
message, buttons); } }, onFailure: onfailure }); } function GetPicklist(table, field){ //Get the
object describe. var fields = sfdc.getSObjectDescribe(table).fields; //Loop through table
fields. for (var i = 0; i < fields.length; i++) { //Find the picklist. if (fields[i].name === field){ //
return the values. return fields[i].picklistValues; } }; } KendoEntry.entry("Batch Add Follow-Up
Events", [ {name: "Subject", label: 'Subject', type: "number", required: true}, {name:
"StartDateTime", label: 'Start', type: "datetime", required: true}, {name: "EndDateTime",
label: 'End', type: "datetime", required: true}, {name: "IsAllDayEvent", label: 'All-Day Event',
type: "boolean", required: true}, {name: "Description", label: 'Description', type: "textarea",
required: true} ], {width: 350}, null, function(selectedValues) { var Subject =
selectedValues["Subject"].value; var StartDateTime =
selectedValues["StartDateTime"].value; var EndDateTime =
selectedValues["EndDateTime"].value; var IsAllDayEvent =
selectedValues["IsAllDayEvent"].value; var Description =
selectedValues["Description"].value; Process(Subject, StartDateTime, EndDateTime,
IsAllDayEvent, Description); }); } });});
```

Follow-Up Task

```
/** * @author CRMCulture * @version 1.00 * @description BatchAdd a follow-up task from a
Task or Event. * * @id AG_BatchAdd_FollowUpTask ActionGrid custom action id that is
added to the name field. * @param kendoEntry ActionGrid helper for Kendo UI modal entry
forms. * @param kendoPopup ActionGrid helper for Kendo UI model message windows. *
@param sfdc ActionGrid helper for Salesforce REST API's*/crmc.require(["KendoEntry",
"KendoPopup", "sfdc"], function(kendoEntry, kendoPopup, sfdc) { var CurrentObject = { object:
{Name: "", Plural: ""}, }; var IsContext = function(object){ objects = [ {Name: "Task", Plural:
"Tasks"}, {Name: "Event", Plural: "Events"} ]; for (var i = 0; i < objects.length; i++)
{ if(objects[i].Name === object){ CurrentObject.object = objects[i]; return true; } }; return
false; }; crmc.addCustomAction({ "itemID": "AG_BatchAdd_FollowUpTask", "isBatchAddItem":
true, "isAvailable": function (context) { var isCorrectContext =
IsContext(context.objectDescribe.name) ; var multipleSelected = context.selectedRows &&
context.selectedRows.length > 0; var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false return
isCorrectContext && multipleSelected && isEnabled; }, "getLabel": function (context) { return
"Follow-Up Task"; }, "createSubmenuItems": function (context) { return []; }, "click": function
(context) { function IsPlural(){ return (context.selectedRows.length === 1 ? "" : "s"); } /**
Language strings.*/ var TEXT_BATCH_ADD_OBJECT = "Batch Add Tasks" var TEXT_ERROR =
"There was an error: "; var TEXT_MESSAGE = "Added {0} Task" + IsPlural() + ", review the
record" + IsPlural() + " with Preview or Browse Records."; var TEXT_BATCH_ADD_RESULTS =
"Batch Add Results"; var TEXT_YES = "Okay"; var TEXT_NO = "No"; function Process(Subject,
ActivityDate, Status, Description){ var ids = []; for (var i = 0; i < context.selectedRows.length;
i++) { ids.push(context.selectedRows[i].id); }; //Query for the data that needs to be passed
into new records. var results = sfdc.query("SELECT Id, WhoId, WhatId" +
(CurrentObject.object.Name === "Event" ? "" : ", Priority") + " FROM " +
CurrentObject.object.Name + " WHERE Id IN('" + ids.join("','") + "')"); //Loop through each result
and pull the data. var records = []; for (var i in results) { var record = new
sforce.SObject('Task'); for (var j in ids) { if(ids[j] === results[i].Id){ record.Subject = Subject;
record.ActivityDate = ActivityDate; record.Status = Status; record.Description = Description;
record.WhoId = results[i].WhoId; record.WhatId = results[i].WhatId; record.Priority =
results[i].Priority; records.push(record); break; }; }; }; var onfailure = function(error)
{ alert(TEXT_ERROR + (error.message || error.faultstring)); };
sforce.connection.create(records, { onSuccess: function(results) { // Get the new record Ids
var recordIds = []; $.each(results, function(i, row) { if (row.errors) { onfailure(row.errors);
return; } recordIds.push(row.id); }); if (results.length == recordIds.length) { var message =
kendo.format(TEXT_MESSAGE, results.length); var buttons = [{ label: TEXT_YES, click:
function() { window.open(kendo.format("/apex/CRMC_PP__crmc_grid?object={0}&Ids={1}",
'Task', recordIds.join())); } } ]; kendoPopup.popupWithButtons(TEXT_BATCH_ADD_RESULTS,
message, buttons); } }, onFailure: onfailure }); } function GetPicklist(table, field){ //Get the
object describe. var fields = sfdc.getSObjectDescribe(table).fields; //Loop through table
fields. for (var i = 0; i < fields.length; i++) { //Find the picklist. if (fields[i].name === field){ //
return the values. return fields[i].picklistValues; } }; } kendoEntry.entry("Batch Add Follow-Up
Tasks", [ {name: "Subject", label: 'Subject', type: "number", required: true}, {name:
"ActivityDate", label: 'Due Date', type: "datetime", required: true}, {name: "Status", label:
'Status', type: "picklist", values: GetPicklist('Task', 'Status'), required: true}, {name:
"Description", label: 'Description', type: "textarea", required: true} ], {width: 350}, null,
function(selectedValues) { var Subject = selectedValues["Subject"].value; var ActivityDate =
selectedValues["ActivityDate"].value; //ar Priority = selectedValues["Priority"].value; var
Status = selectedValues["Status"].value; var Description =
selectedValues["Description"].value; Process(Subject, ActivityDate, Status,
Description); }); } });});
```

Marketo

Marketo is an e-mail marketing service that can be installed on a Salesforce org. You can also modify this example to send from other objects with e-mail address fields.

```
/**
 * @author CRMCulture
 * @version 1.00
 * @description Custom Batch Email from Ooportunites.
 *
 * @param kendoEntry ActionGrid helper for Kendo UI modal entry forms.
 * @param kendoPopup ActionGrid helper for Kendo UI model message windows.
 * @param sfdc ActionGrid helper for Salesforce REST API's
*/

crmc.require(["KendoPopup", "sfdc" ], function(kendoPopup, sfdc) {
 crmc.addCustomAction({
 "itemID": "AG_MarketoEmail",
 "isAvailable": function (context) {
     var isSelected = context.selectedRows.length >= 1;
     var isAccessable = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false;
     var isCorrectObj = context.objectDescribe.name == "Lead";
     return isSelected && isAccessable && isCorrectObj;

 },
 "getLabel": function (context) {
 return "Send Marketo E-mail"; }, "createSubmenuItems": function (context) { return []; }, "click": function (context)
{ function processURL(){ var ids = []; var page = window.location.href; context.selectedRows.map(function(row)
{ ids.push(row.Id.substring(0,15)); }); window.open("/apex/mkto_si__Send_Marketo_Email?
contactType=Lead&contactIds=" + ids.join() + "&retUrl=" + page); }; if(context.selectedRows.length > 5){ var
buttons = [{ label: "Yes", click: function() { processURL(); } }, { label: "No" }];
kendoPopup.popupWithButtons("WARNING", "5 records are recommended because each item will show in a new
tab. Do you want to continue?", buttons); } else{ processURL(); } } });
```

Send to Pardot

It loops through every row and passes the ID and field data into URLs. If more than 10 are selected, it will batch process URLs to avoid invalid URLs.

```
crmc.addCustomAction({ "itemID": "SendToPardot", "isAvailable": function (context){return
this.featureSecurity.getSetting(context.objectDescribe.name,this.itemID)!==!1}, "getLabel":
function (context){return"Send to Pardot"}, "createSubmenuItems": function (context)
{return[]}, "click": function (context){ var makeURL = function(row, retUrl) { var email =
encodeURIComponent(row.Email); var id = encodeURIComponent(row.id); var partnerSrv =
encodeURIComponent(context.actionGrid.settings.Partner_Server_URL_290); var session =
encodeURIComponent(sforce.connection.sessionId); // console.log('Email: ' + row.Email + ', ID:
' + row.id); // console.log('Session: ' + session); // console.log('Partner Server: ' + partnerSrv);
return 'https://pi.pardot.com/prospect/sync/email/'+ email +'?contact_id='+ id
+'&sessionid='+ session +'&serverurl='+ partnerSrv +'&redirect_location=' +
encodeURIComponent(retUrl); } var windows = []; var rowsProcessed = 0; var batchCount =
Math.ceil(context.selectedRows.length / 10); var timer = setInterval(function () {Monitor()},
500); var queue = []; for (var i = 0; i < batchCount; i++) { var tUrl = window.location; var
batch = Between(rowsProcessed, context.selectedRows); batch.map(function(row) { if(row !=
undefined){ tUrl = makeURL(row, tUrl); } }); rowsProcessed += batch.length;
queue.push(tUrl); }; function ProcessUrl(url){ windows.push(window.open(url));
windows[windows.length - 1].addEventListener('load', function() { windows[window.length -
1].close(); }, false); } function Between(start, array){ var end = start + 10 var list = []; for (var i
= start; i <= end - 1; i++) { if(i > array.length - 1){ break; } else{ list.push(array[i]); } } return
list; } function Monitor(){ if(queue.length > 0){ if(windows.length < 5)
{ PushBatch(undefined); } }else if(queue.length == 0){ clearTimeout(timer); alert("Sync to
Pardot has finished! Please wait for the remaining windows to close automatically :)"); } //
Some Loading bar code here... } function BatchComplete(index){ windows[index].close();
PushBatch(index); } function PushBatch(index){ if(index == undefined)
{ windows.push(window.open(queue.shift())); var index = windows.length - 1;
windows[index].addEventListener('load', function() { BatchComplete(index); }, false); }
else{ if(queue.length > 0){ windows[index] = window.open(queue.shift());
windows[index].addEventListener('load', function() { BatchComplete(index); },
false); } } } }});///Old// crmc.addCustomAction({// "itemID": "SendToPardot",// "isAvailable":
function (context){return
this.featureSecurity.getSetting(context.objectDescribe.name,this.itemID)!==!1},// "getLabel":
function (context){return"Send to Pardot"},// "createSubmenuItems": function (context)
{return[]},// "click": function (context){// var makeURL = function(row, retUrl) {// email =
encodeURIComponent(row.Email);// id = encodeURIComponent(row.id);// var partnerSrv =
encodeURIComponent(context.actionGrid.settings.Partner_Server_URL_290);// var session =
encodeURIComponent(sforce.connection.sessionId);// // console.log('Email: ' + row.Email + ',
ID: ' + row.id);// // console.log('Session: ' + session);// // console.log('Partner Server: ' +
partnerSrv);// return 'https://pi.pardot.com/prospect/sync/email/'+ email +'?contact_id='+ id
+'&sessionid='+ session +'&serverurl='+ partnerSrv +'&redirect_location=' +
encodeURIComponent(retUrl);// }// var url = window.location;//
if(context.selectedRows.length > 10){// console.log("Sorry, 10 records or less :(");// alert("Sorry,
10 records or less :(");// }// else{// context.selectedRows.map(function(row) {// url =
makeURL(row, url);// });// window.location = url;// }// }// });
```

Accounting - Accounts Payable in Conga Grid

Use these code examples to create custom actions for managing accounts payable in Conga Grid℠.

- Accounts Payable: Pay
- Accounts Payable: Post Account
- Accounts Payable: Unpost Account
- Expense Report: Create Accounts Payable
- Pay From Account Payable
- Pay From Account Payable Line
- Post From Account Payable
- Post From Account Payable Line
- Recurring Accounts Payable: Create Accounts Payable
- Unpost From Accounts Payable
- Unpost From Accounts Payable Line

Accounts Payable: Pay

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_AccountPayable_PayBatch", "isAvailable": function (context) { // This function is
called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Account_Payable__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Pay Batch"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
```

domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { // Couldn't load app script, assume not installed popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name: "ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed", "AccountPayableBatchPost", parameters, selectedIds); } else { popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});

Accounts Payable: Post Account

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) {
  crmc.addCustomAction({
    "itemID": "AcctSeed_AccountPayable_PostAccountPayables",
    "isAvailable": function (context) {
      // This function is called before the action item is displayed and returns a boolean if the item should be
displayed
      // By default determine availability based on Feature Security for this action
       var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false;
        // Only allow batch adding from Accounts object for now
        var isProject = context.objectDescribe.name == "AcctSeed__Account_Payable__c";
        var multipleSelected = context.selectedRows && context.selectedRows.length > 0;
        return isProject && isEnabled && multipleSelected;
    },
    "getLabel": function (context) {
      // This function returns the display label of the action item and is called before the item is shown
      return "Post Accounts Payable";
    },
    "createSubmenuItems": function (context) {
      // If this function returns additional action item objects, they will appear as submenu items
      return [];
    },
    "click": function (context) {
        // Include static resource from managed namespace pre-creates session and allows us to safely POST
data without redirect issue
        var head = document.getElementsByTagName('HEAD').item(0);
        var script = document.createElement("script");
        script.type = "text/javascript";
        head.appendChild(script);
        var domains = window.location.hostname.split(".");
        if (domains.length == 3) {
          instance = domains[0];
        }
        else {
          instance = domains[1];
        }
```

```
          script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" +
new Date().getTime();
          script.onerror = function() {
            // Couldn't load app script, assume not installed
            popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app installed?");
          };
          script.onload = function() {
            if (context.selectedRows.length > 0) {
              var selectedIds = [];
              Object.each(context.selectedRows, function (item) { selectedIds.push({name: "ids", value: item["Id"]}); });
              var parameters = {
                retURL: context.actionGrid.settings.pageURL,
              };
                sfdc.postToVFPage("acctseed", "AccountPayableBatchPost", parameters, selectedIds);
            }
            else {
                popup.popup('Please select records', 'Select at least one record to process.');
            }
          };
        }
    });
});
```

Accounts Payable: Unpost Account

```
 crmc.require([ 'sfdc' ,  'KendoPopup' ], function (sfdc, popup) {
     crmc.addCustomAction({
          "itemID" :  "AcctSeed_AccountPayable_UnpostAccountPayables" ,
          "isAvailable" : function (context) {
          // This function is called before the action item is displayed and
returns a boolean if the item should be displayed
          // By default determine availability based on Feature Security for
this action
             var isEnabled =  this .featureSecurity.getSetting(context.object
Describe.name,  this .itemID) !==  false ;

 // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Account_Payable__c"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0;
return isProject && isEnabled && multipleSelected; }, "getLabel": function
(context) { // This function returns the display label of the action item
and is called before the item is shown return "UnPost Accounts Payable"; },
"createSubmenuItems": function (context) { // If this function returns
additional action item objects, they will appear as submenu items return
[]; }, "click": function (context) { //Include static resource from managed
namespace pre-creates session and allows us to safely POST data without
redirect issue var head = document.getElementsByTagName('HEAD').item(0); var
script = document.createElement("script"); script.type = "text/javascript";
head.appendChild(script); var domains = window.location.hostname.split(".");
if (domains.length == 3) { instance = domains[0]; } else { instance =
domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new
Date().getTime(); script.onerror = function() { // Couldn't load app script,
assume not installed popup.popup("Could not obtain session for Accounting
Seed Financial Suite is the app installed?"); }; script.onload = function()
{ if (context.selectedRows.length > 0) { var selectedIds = [];
Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL:
context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed",
"AccountPayableBatchUnpost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to
process.'); } }; } });});
```

Expense Report: Create Accounts Payable

crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID": "AcctSeed_ExpenseReport_CreateAccountsPayable", "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject = context.objectDescribe.name == "AcctSeed__Expense_Report__c"; var multipleSelected = context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the display label of the action item and is called before the item is shown return "Create Accounts Payable"; }, "createSubmenuItems": function (context) { // If this function returns additional action item objects, they will appear as submenu items return []; }, "click": function (context) { // Include static resource from managed namespace pre-creates session and allows us to safely POST data without redirect issue var head = document.getElementsByTagName('HEAD').item(0); var script = document.createElement("script"); script.type = "text/javascript"; head.appendChild(script); var domains = window.location.hostname.split("."); if (domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { // Couldn't load app script, assume not installed

```
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "AccountPayableFromExpenseReport", parameters,
selectedIds); } else { popup.popup('Please select records', 'Select at least one record to
process.'); } }; } });});
```

Pay From Account Payable

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"PayBatchAccountsPayable", "isAvailable": function (context) { // This function is called
before the action item is displayed and returns a boolean if the item should be displayed //
By default determine availability based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isVisible =
context.objectDescribe.name == ("AcctSeed__Account_Payable__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isVisible && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Pay Batch"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { //https://
acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wra
pMassAction=1&scontrolCaching=1 // Include static resource from managed namespace
pre-creates session and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "AccountPayablePayBatch", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to add to a post to
accounts payable.'); } }; } });});
```

Pay From Account Payable Line

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"PayBatch_From_Account_Payable_Line", "isAvailable": function (context) { // This function is
called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isVisible =
context.objectDescribe.name == ("AcctSeed__Account_Payable_Line__c"); var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isVisible && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"Pay Batch"; }, "createSubmenuItems": function (context) { // If this function returns
additional action item objects, they will appear as submenu items return []; }, "click":
function (context) { //https://acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wrapMassAction=1&scontrolCaching=1 // Include static resource from managed namespace
pre-creates session and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { var id =
item["AcctSeed__Account_Payable__c"]; if(id === undefined || id === null){ //Do nothing if
empty. } else if(!checkIds(id, selectedIds)){ selectedIds.push({name: "ids", value: id}); } });
if(selectedIds.length > 0) { var parameters = { retURL:
context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed",
"AccountPayablePayBatch", parameters, selectedIds); } else { popup.popup('No IDs to POST',
'Make sure that Accounts Payable is loaded on the grid and that the fields are not
empty'); } } else { popup.popup('Please select records', 'Select at least one record to add to a
post to accounts payable.'); } }; //Does the ID already exist within the array? function
checkIds(id, arr){ for (var i = 0; i < arr.length; i++) { if(arr[i].value === id){ return true; } }; return
false; } } });});
```

Post From Account Payable

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"PayBatchAccountsPayable", "isAvailable": function (context) { // This function is called
before the action item is displayed and returns a boolean if the item should be displayed //
By default determine availability based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isVisible =
context.objectDescribe.name == ("AcctSeed__Account_Payable__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isVisible && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Pay Batch"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { //https://
acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wra
pMassAction=1&scontrolCaching=1 // Include static resource from managed namespace
pre-creates session and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "AccountPayablePayBatch", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to add to a post to
accounts payable.'); } }; } });});
```

Post From Account Payable Line

Use this code template to create a custom action for Post from Account Payable Line.

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"PayBatch_From_Account_Payable_Line", "isAvailable": function (context) { // This function is
called before the action item is displayed and returns a boolean if the item should be
```

displayed // By default determine availability based on Feature Security for this action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only allow batch adding from Accounts object for now var isVisible = context.objectDescribe.name == ("AcctSeed__Account_Payable_Line__c"); var multipleSelected = context.selectedRows && context.selectedRows.length > 0; return isVisible && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the display label of the action item and is called before the item is shown return "Pay Batch"; }, "createSubmenuItems": function (context) { // If this function returns additional action item objects, they will appear as submenu items return []; }, "click": function (context) { //https://acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wrapMassAction=1&scontrolCaching=1 // Include static resource from managed namespace pre-creates session and allows us to safely POST data without redirect issue var head = document.getElementsByTagName('HEAD').item(0); var script = document.createElement("script"); script.type = "text/javascript"; head.appendChild(script); var domains = window.location.hostname.split("."); if (domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { // Couldn't load app script, assume not installed popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var selectedIds = []; Object.each(context.selectedRows, function (item) { var id = item["AcctSeed__Account_Payable__c"]; if(id === undefined || id === null){ //Do nothing if empty. } else if(!checkIds(id, selectedIds)){ selectedIds.push({name: "ids", value: id}); } }); if(selectedIds.length > 0) { var parameters = { retURL: context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed", "AccountPayablePayBatch", parameters, selectedIds); } else { popup.popup('No IDs to POST', 'Make sure that Accounts Payable is loaded on the grid and that the fields are not empty'); } } else { popup.popup('Please select records', 'Select at least one record to add to a post to accounts payable.'); } }; //Does the ID already exist within the array? function checkIds(id, arr){ for (var i = 0; i < arr.length; i++) { if(arr[i].value === id){ return true; } }; return false; } } });});

Recurring Accounts Payable: Create Accounts Payable

crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID": "AcctSeed_RecurringAccountPayable_CreateAccountsPayable", "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this action var isEnabled =

```
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Recurring_Account_Payable__c"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"Create Accounts Payable"; }, "createSubmenuItems": function (context) { // If this function
returns additional action item objects, they will appear as submenu items return []; }, "click":
function (context) { // Include static resource from managed namespace pre-creates session
and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "AccountsPayableFromRecrAccountsPayable", parameters,
selectedIds); } else { popup.popup('Please select records', 'Select at least one record to
process.'); } }; } });});
```

Unpost From Accounts Payable

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"UnPost_AP_Batch_From_Account_Payable_Line", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isVisible
= context.objectDescribe.name == ("AcctSeed__Account_Payable_Line__c"); var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isVisible && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"UnPost Accounts Payable"; }, "createSubmenuItems": function (context) { // If this function
returns additional action item objects, they will appear as submenu items return []; }, "click":
function (context) { //https://acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wrapMassAction=1&scontrolCaching=1 // Include static resource from managed namespace pre-creates session and allows us to safely POST data without redirect issue var head = document.getElementsByTagName('HEAD').item(0); var script = document.createElement("script"); script.type = "text/javascript"; head.appendChild(script); var domains = window.location.hostname.split("."); if (domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { // Couldn't load app script, assume not installed popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var selectedIds = []; Object.each(context.selectedRows, function (item) { var id = item["AcctSeed__Account_Payable__c"]; if(id === undefined || id === null){ //Do nothing if empty. } else if(!checkIds(id, selectedIds)){ selectedIds.push({name: "ids", value: id}); } }); if(selectedIds.length > 0) { var parameters = { retURL: context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed", "AccountPayableBatchUnpost", parameters, selectedIds); } else { popup.popup('No IDs to POST', 'Make sure that Accounts Payable is loaded on the grid and that the fields are not empty'); } } else { popup.popup('Please select records', 'Select at least one record to add to a post to accounts payable.'); } }; //Does the ID already exist within the array? function checkIds(id, arr){ for (var i = 0; i < arr.length; i++) { if(arr[i].value === id){ return true; } }; return false; } } });});
```

Unpost From Accounts Payable Line

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID": "UnPost_AP_Batch_From_Account_Payable_Line", "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only allow batch adding from Accounts object for now var isVisible = context.objectDescribe.name == ("AcctSeed__Account_Payable_Line__c"); var multipleSelected = context.selectedRows && context.selectedRows.length > 0; return isVisible && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the display label of the action item and is called before the item is shown return "UnPost Accounts Payable"; }, "createSubmenuItems": function (context) { // If this function returns additional action item objects, they will appear as submenu items return []; }, "click": function (context) { //https://acctseed.cs7.visual.force.com/apex/AccountPayablePayBatch?
retURL=%2Fa0H%3Ffcf%3D00BF0000006rihU%26rolodexIndex%3D-1%26page%3D1&wra
pMassAction=1&scontrolCaching=1 // Include static resource from managed namespace
pre-creates session and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript";
head.appendChild(script); var domains = window.location.hostname.split("."); if
(domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; }
script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/
AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { //
Couldn't load app script, assume not installed popup.popup("Could not obtain session for
Accounting Seed Financial Suite is the app installed?"); }; script.onload = function() { if
(context.selectedRows.length > 0) { var selectedIds = [];
Object.each(context.selectedRows, function (item) { var id =
item["AcctSeed__Account_Payable__c"]; if(id === undefined || id === null){ //Do nothing if
empty } else if(!checkIds(id, selectedIds)){ selectedIds.push({name: "ids", value: id}); }
}); if(selectedIds.length > 0) { var parameters = { retURL:
context.actionGrid.settings.pageURL }; sfdc.postToVFPage("acctseed",
"AccountPayableBatchUnpost", parameters, selectedIds); } else { popup.popup('No IDs to
POST', 'Make sure that Accounts Payable is loaded on the grid and that the fields are not
empty'); } } else { popup.popup('Please select records', 'Select at least one record to
add to a post to accounts payable.'); } }; //Does the ID already exist within the
array? function checkIds(id, arr){ for (var i = 0; i < arr.length; i++) { if(arr[i].value === id)
{ return true; } } return false; } } });});
```

Accounting - Billing in Conga Grid

Use these code examples to create custom actions for billing in Conga Grid℠.

- Billing
- Billing: Unpost Billings
- Create Billing
- Expense Report: Create Billings
- Opportunity: Post Billing
- Post Billings
- Unpost Billings

Billing

```
// {!REQUIRESCRIPT("https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js")} //
jQuery.noConflict();// (function($, buttonObj) {// var pageName = "BillingBatchPost";// var
disabledLabel = "Posting...";// var packageName = (buttonObj.name.indexOf("acctseed") >=
0 ? "acctseed__" : "");// var submitUrl = "{!URLFOR('/apex/" + packageName + pageName +
"')}";// if (buttonObj.name.indexOf("acctseed") >= 0) {submitUrl = submitUrl.replace("--c","--
acctseed").replace("//c","//acctseed");}// var submitForm = buttonObj.form;//
submitForm.action = submitUrl + "?retURL=" + encodeURIComponent(location.pathname +
location.search) + "&wrapMassAction=1&scontrolCaching=1";// submitForm.onsubmit =
function() { return true }; // $
(".listButtons .btn").not(".refreshListButton").toggleClass("btnDisabled").val(disabledLabel).pr
op("disabled", true);// submitForm.submit();// })(jQuery,this);crmc.require(['sfdc', 'KendoPopup'],
function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"Batch_Post_Billings_From_Billings", "isAvailable": function (context) { // This function is
called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Billing__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Post Billings"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "BillingBatchPost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to add to a post
Billings.'); } }; } });});
```

Billing: Unpost Billings

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_Billing_UnpostBillings", "isAvailable": function (context) { // This function is called
before the action item is displayed and returns a boolean if the item should be displayed //
By default determine availability based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Billing__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "UnPost Billings"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "BillingBatchUnpost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Create Billing

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"createOrdersFromProjects", "isBatchAddItem": true, "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var
isOpportunity = context.objectDescribe.name == "Opportunity"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isOpportunity &&
isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the
display label of the action item and is called before the item is shown return "Create
Billing"; }, "createSubmenuItems": function (context) { // If this function returns additional
action item objects, they will appear as submenu items return []; }, "click": function (context)
{ if(context.selectedRows.length <= 5){ var ids = []; var submitUrl = "/apex/
acctseed__BillingFromOpportunity?id=" context.selectedRows.map(function(row)
{ ids.push(row.Id); }); for (var i = 0; i < ids.length; i++) { window.open(submitUrl + ids[i]); }; } else
{ popup.popup("Record Selection", "Currently, there is only support for 5 max records per
batch."); } } });});
```

Expense Report: Create Billings

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_ExpenseReport_CreateBillings", "isAvailable": function (context) { // This function
is called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Expense_Report__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Create Billings"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
```

sfdc.postToVFPage("acctseed", "BillingFromExpenseReport", parameters, selectedIds); } else { popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});

Opportunity: Post Billing

crmc.require(['sfdc', 'KendoPopup', 'KendoEntry'], function (sfdc, popup, entry) { crmc.addCustomAction({ "itemID": "AcctSeed_Opportunity_PostBilling", "isBatchAddItem": true, "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only allow batch adding from Accounts object for now var isOpportunity = context.objectDescribe.name == "Opportunity"; var multipleSelected = context.selectedRows && context.selectedRows.length > 0; return isOpportunity && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the display label of the action item and is called before the item is shown return "Create Billing"; }, "createSubmenuItems": function (context) { // If this function returns additional action item objects, they will appear as submenu items return []; }, "click": function (context) { var ids = []; if(context.selectedRows.length <= 5){ var submitUrl = "/apex/acctseed__BillingFromOpportunity?id=" context.selectedRows.map(function(row) { ids.push(row.Id); }); for (var i = 0; i < ids.length; i++) { window.open(submitUrl + ids[i]); }; afterUpdate(); } else { popup.popup("Record Selection", "Currently, there is only support for 5 max records per batch."); } function afterUpdate(){ // Prompt to navigate the user to the results var buttons = [{ label: "Yes", click: function() { var billings = sfdc.query("SELECT Name, Id FROM AcctSeed__Billing__c WHERE AcctSeed__Opportunity__c IN('" + ids.join("','") + "') AND isdeleted = false"); form(billings); } }, { label: "No" }]; popup.popupWithButtons("Finished reviewing tabs?", "Would you like to choose the Posting Status value?", buttons); function form(ids){ var objectDescribe = sfdc.getSObjectDescribe("AcctSeed__Billing__c"); var picklistValues = []; Object.each(objectDescribe.fields, function(field){ if (field.name == "AcctSeed__Status__c"){ picklistValues = field.picklistValues; } }); entry.entry("Posting Status", [{name: "AcctSeed__Status__c", label: 'Select Status', type: 'picklist', values: picklistValues, }], null, null, function(selectedValues) { var selectedStatus = selectedValues["AcctSeed__Status__c"].value; update(ids, selectedStatus); }); } function update(ids, status){ var pass = 0; var fail = 0 for (var i = 0; i < ids.length; i++) { //create an account var record = new sforce.SObject("AcctSeed__Billing__c"); //update that Billing record. record.id = ids[i].Id; record.AcctSeed__Status__c = status; result = sforce.connection.update([record]); if (result[0].getBoolean("success")) { pass++; } else { fail++; } }; viewResults(pass, fail, ids); } function viewResults(pass, fail, ids){ // Prompt to navigate the user to the results var message = "" + pass + " have been updated" + ", would you like to

```
Conga Grid the results?" var buttons = [{ label: "Yes", click: function() { urlPrams = []; for (var i
= 0; i < ids.length; i++) { urlPrams.push(ids[i].Id); }; // Navigate to a list of just these recordIds
window.open(kendo.format("/apex/CRMC_PP__crmc_grid?object={0}&Ids={1}",
"AcctSeed__Billing__c", urlPrams.join())); } }, { label: "No" }];
popup.popupWithButtons("Updated Billings", message, buttons); } } } });});
```

Post Billings

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_Billing_PostBillings", "isAvailable": function (context) { // This function is called
before the action item is displayed and returns a boolean if the item should be displayed //
By default determine availability based on Feature Security for this action var isEnabled =
this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only
allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Billing__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Post Billings"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "BillingBatchPost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Unpost Billings

```
// {!REQUIRESCRIPT('https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js')} //
jQuery.noConflict();// (function($, buttonObj) {// var pageName = "BillingBatchUnpost";// var
disabledLabel = "Unposting...";// var packageName = (buttonObj.name.indexOf("acctseed")
>= 0 ? "acctseed__" : "");// var submitUrl = "{!URLFOR('/apex/" + packageName + pageName +
"')}";// if (buttonObj.name.indexOf("acctseed") >= 0) {submitUrl = submitUrl.replace("--c","--
acctseed").replace("//c","//acctseed");}// var submitForm = buttonObj.form;//
submitForm.action = submitUrl + "?retURL=" + encodeURIComponent(location.pathname +
location.search) + "&wrapMassAction=1&scontrolCaching=1";// submitForm.onsubmit =
function() { return true }; // $
(".listButtons .btn").not(".refreshListButton").toggleClass("btnDisabled").val(disabledLabel).pr
op("disabled", true);// submitForm.submit();// })(jQuery,this);crmc.require(['sfdc', 'KendoPopup'],
function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"Batch_UnPost_Billings_From_Billings", "isAvailable": function (context) { // This function is
called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Billing__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "UnPost Billings"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "BillingBatchUnpost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to unpost
Billings.'); } }; } });});
```

Accounting - Miscellaneous in Conga Grid

Use these code examples to create custom actions for other common tasks in Conga Grid℠.

- Mass Delete Financial Report Results
- Post Cash Receipts
- Post Journal Entries
- Post Scheduled Revenue Expense
- Process Orders From Projects
- Unpost Cash Receipts
- Unpost Journal Entries
- Unpost Scheduled Revenue Expense

Mass Delete Financial Report Results

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_FinancialReportResult_MassDelete", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Financial_Report_Result__c"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"Post Accounts Payable"; }, "createSubmenuItems": function (context) { // If this function
returns additional action item objects, they will appear as submenu items return []; }, "click":
function (context) { // Include static resource from managed namespace pre-creates session
and allows us to safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "FinancialReportResultDelete", parameters, selectedIds); }
else { popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Post Cash Receipts

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_CashReceipt_PostCashReceipts", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Cash_Receipt__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Pay Batch"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "CashReceiptBatchPost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Post Journal Entries

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_JournalEntry_PostJournalEntries", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Account_Payable__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Post Journal Entries"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "JournalEntryBatchPost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Post Scheduled Revenue Expense

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_ScheduledRevenueExpense_Post", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Scheduled_Revenue_Expense__c"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"Post"; }, "createSubmenuItems": function (context) { // If this function returns additional
action item objects, they will appear as submenu items return []; }, "click": function (context)
{ // Include static resource from managed namespace pre-creates session and allows us to
safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
```

var domains = window.location.hostname.split("."); if (domains.length == 3) { instance = domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance + ".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime(); script.onerror = function() { // Couldn't load app script, assume not installed popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name: "ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, }; sfdc.postToVFPage("acctseed", "ScheduledRevenueExpenseBatchPost", parameters, selectedIds); } else { popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});

Process Orders From Projects

crmc.addCustomAction({ "itemID": "Batch_Process_Orders_From_Projects", "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject = context.objectDescribe.name == "crmc_Project__c"; var multipleSelected = context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function returns the display label of the action item and is called before the item is shown return "Create Order(s)"; }, "createSubmenuItems": function (context) { // If this function returns additional action item objects, they will appear as submenu items return []; }, "click": function (context) { // This function is what is executed when the action item is clicked // Context object has this format: / { selectedRows: rows selected in AG loadedRows: all rows loaded in AG objectDescribe: metadata for current custom object kendoGrid: kendoGrid object } */ var ids = []; // Show a prompt that contains the fields in FIELDSETNAME // For every selected row context.selectedRows.map(function(row) { ids.push(row.Id); }); window.open("/apex/Project_List_Create_Order?projectRecordIds=" + ids.join('%2c')); }});*

Unpost Cash Receipts

crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID": "AcctSeed_CashReceipt_UnpostCashReceipts", "isAvailable": function (context) { // This function is called before the action item is displayed and returns a boolean if the item should be displayed // By default determine availability based on Feature Security for this

```
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Cash_Receipt__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Pay Batch"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "CashReceiptBatchUnpost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Unpost Journal Entries

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_Batch_Unpost_Journal_Entries", "isAvailable": function (context) { // This function
is called before the action item is displayed and returns a boolean if the item should be
displayed // By default determine availability based on Feature Security for this action var
isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name, this.itemID) !==
false; // Only allow batch adding from Accounts object for now var isProject =
context.objectDescribe.name == "AcctSeed__Account_Payable__c"; var multipleSelected =
context.selectedRows && context.selectedRows.length > 0; return isProject && isEnabled &&
multipleSelected; }, "getLabel": function (context) { // This function returns the display label
of the action item and is called before the item is shown return "Unpost Journal Entries"; },
"createSubmenuItems": function (context) { // If this function returns additional action item
objects, they will appear as submenu items return []; }, "click": function (context) { // Include
static resource from managed namespace pre-creates session and allows us to safely
POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "JournalEntryBatchUnpost", parameters, selectedIds); } else
{ popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});
```

Unpost Scheduled Revenue Expense

```
crmc.require(['sfdc', 'KendoPopup'], function (sfdc, popup) { crmc.addCustomAction({ "itemID":
"AcctSeed_ScheduledRevenueExpense_Unpost", "isAvailable": function (context) { // This
function is called before the action item is displayed and returns a boolean if the item
should be displayed // By default determine availability based on Feature Security for this
action var isEnabled = this.featureSecurity.getSetting(context.objectDescribe.name,
this.itemID) !== false; // Only allow batch adding from Accounts object for now var isProject
= context.objectDescribe.name == "AcctSeed__Scheduled_Revenue_Expense__c"; var
multipleSelected = context.selectedRows && context.selectedRows.length > 0; return
isProject && isEnabled && multipleSelected; }, "getLabel": function (context) { // This function
returns the display label of the action item and is called before the item is shown return
"Unpost"; }, "createSubmenuItems": function (context) { // If this function returns additional
action item objects, they will appear as submenu items return []; }, "click": function (context)
{ // Include static resource from managed namespace pre-creates session and allows us to
safely POST data without redirect issue var head =
document.getElementsByTagName('HEAD').item(0); var script =
document.createElement("script"); script.type = "text/javascript"; head.appendChild(script);
var domains = window.location.hostname.split("."); if (domains.length == 3) { instance =
domains[0]; } else { instance = domains[1]; } script.src = "https://AcctSeed." + instance +
".visual.force.com/resource/AcctSeed__ButtonJSFunctions?ts=" + new Date().getTime();
script.onerror = function() { // Couldn't load app script, assume not installed
popup.popup("Could not obtain session for Accounting Seed Financial Suite is the app
installed?"); }; script.onload = function() { if (context.selectedRows.length > 0) { var
selectedIds = []; Object.each(context.selectedRows, function (item) { selectedIds.push({name:
"ids", value: item["Id"]}); }); var parameters = { retURL: context.actionGrid.settings.pageURL, };
sfdc.postToVFPage("acctseed", "ScheduledRevenueExpenseBatchUnpost", parameters,
```

selectedIds); } else { popup.popup('Please select records', 'Select at least one record to process.'); } }; } });});

Secure Mass Update

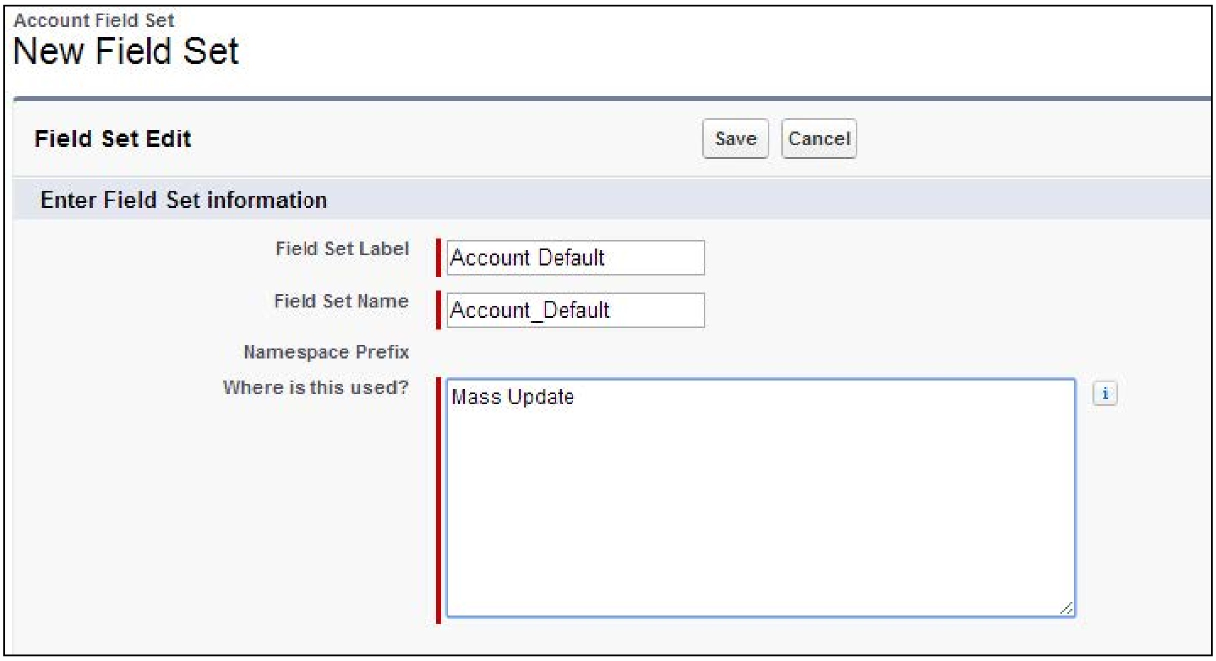
Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.

The Secure Mass Update Tool updates multiple fields on multiple records, all native within Salesforce. Configuration is tightly controlled by Salesforce Field Sets and security. This functionality and more are available within Conga Grid.

Secure Mass Update Setup and Administration

> **Warning**
>
> **Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.**

1. Go to the Mass Update App, and select the Mass Update Settings Tab.
   If you are using Salesforce Professional Edition and below, Secure Mass Update Setup and Administration.
2. Create field sets for the objects you would like to batch edit. Do this by going to the Setup → Customize → (Object) → Field Sets.



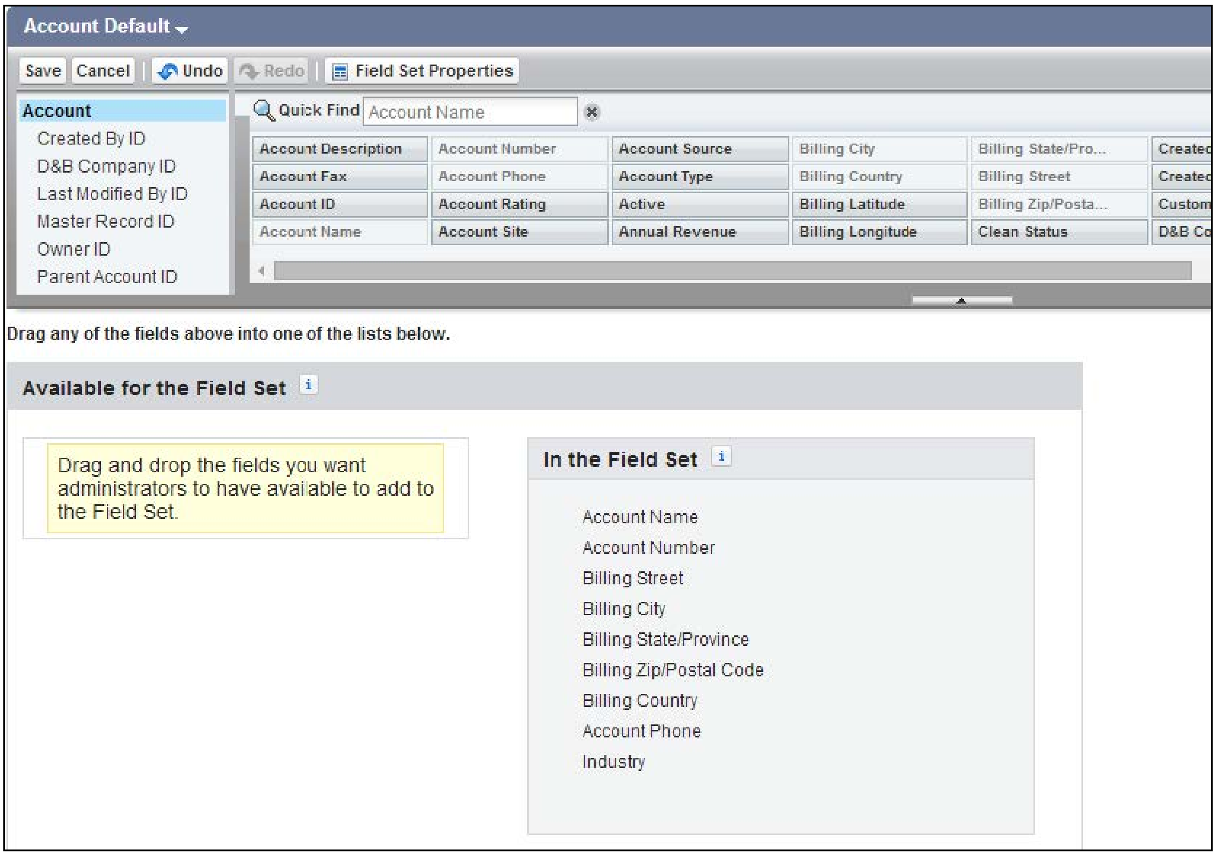
Account Field Set

New Field Set

Field Set Edit

Enter Field Set information

Field Set Label	Account Default
Field Set Name	Account_Default
Namespace Prefix	
Where is this used?	Mass Update



Account Default

Account
Created By ID
D&B Company ID
Last Modified By ID
Master Record ID
Owner ID
Parent Account ID

Quick Find: Account Name

Account Description	Account Number	Account Source	Billing City	Billing State/Pro...	Createc
Account Fax	Account Phone	Account Type	Billing Country	Billing Street	Createc
Account ID	Account Rating	Active	Billing Latitude	Billing Zip/Posta...	Custom
Account Name	Account Site	Annual Revenue	Billing Longitude	Clean Status	D&B Co

Drag any of the fields above into one of the lists below.

Available for the Field Set

Drag and drop the fields you want administrators to have available to add to the Field Set.

In the Field Set

Account Name
Account Number
Billing Street
Billing City
Billing State/Province
Billing Zip/Postal Code
Billing Country
Account Phone
Industry

3. At the Mass Update Settings page, click Add New to create a new Mass Update configuration row:

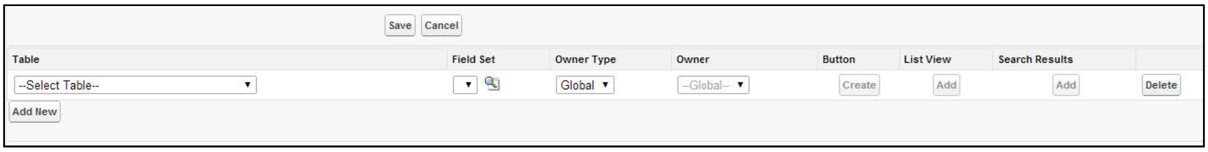
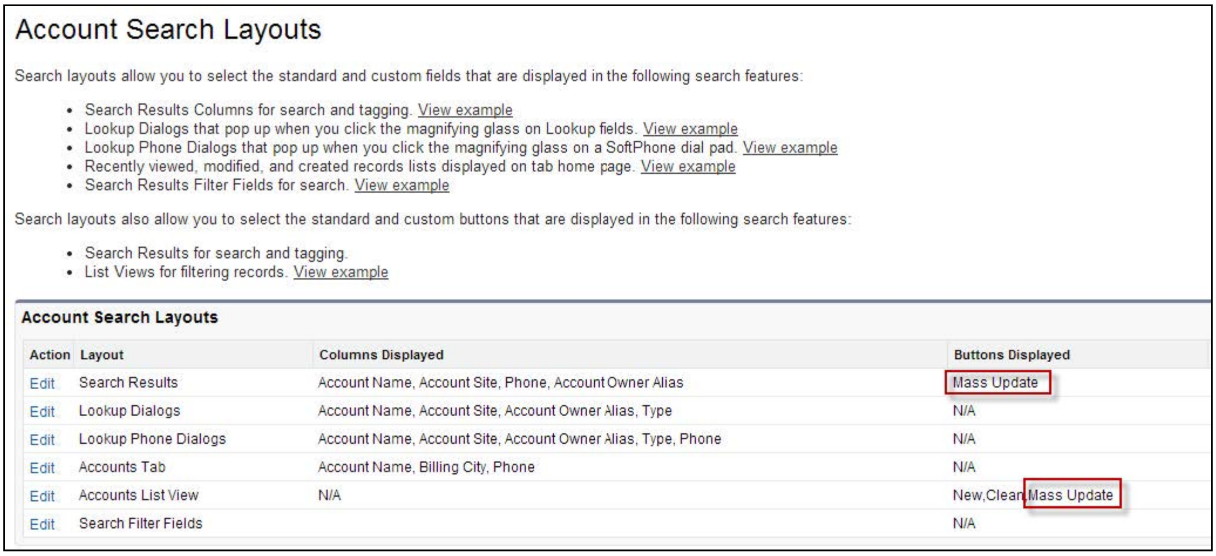


Table	Field Set	Owner Type	Owner	Button	List View	Search Results	
--Select Table--		Global	--Global--	Create	Add	Add	Delete

4. Select the Object (Table), field set, and Owner Type.
   Global allows for all users, despite profiles.
   Selecting Profile will enable the Owner dropdown list, which will allow selection of a particular Profile
   Selecting the User Owner Type will enable the Owner dropdown list, which will allow selection of any particular User that is defined in the system
5. When selecting Create, a button is defined automatically for that object. Clicking Add under List View and Search Results will automatically place that button under the <Object> List View and the Search Results for that Object. Below is a screen shot of the end result:



Account 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

Account Search Layouts

Action	Layout	Columns Displayed	Buttons Displayed
Edit	Search Results	Account Name, Account Site, Phone, Account Owner Alias	Mass Update
Edit	Lookup Dialogs	Account Name, Account Site, Account Owner Alias, Type	N/A
Edit	Lookup Phone Dialogs	Account Name, Account Site, Account Owner Alias, Type, Phone	N/A
Edit	Accounts Tab	Account Name, Billing City, Phone	N/A
Edit	Accounts List View	N/A	New,Clean,Mass Update
Edit	Search Filter Fields		N/A

6. Make sure to click Save to save your new row.
7. You optionally can create a button for Related Lists. For instructions on how to do this, refer to Applying Mass Update to Related Lists.

Additional Steps for Salesforce Professional Edition

The installation steps for Professional Edition Users and below are different. Please perform the following steps instead of using the automatic configuration tool:

1. Go to Setup
2. Go to Customize > (Object you want to configure Mass Update)
3. Click Buttons, Links, and Actions
4. lick New Button or Link

Fill out the following fields with the following information:

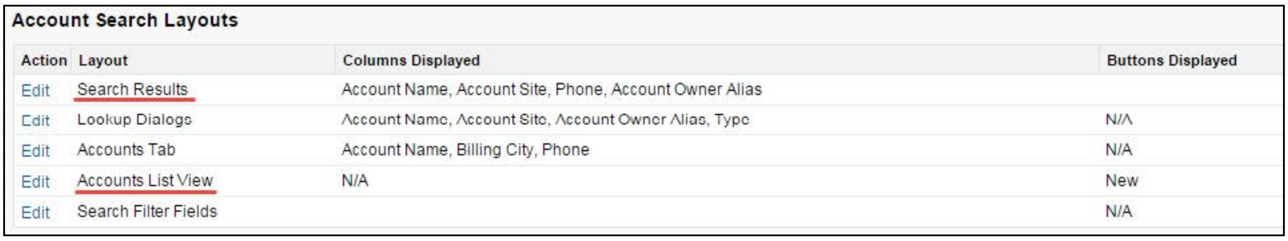
Label: “Mass Update” Name: “crmc_Mass_Update” Display Type: List Button (Check the Display checkboxes)

Behavior: Execute JavaScript Content Source: OnClick JavaScript Then, in the Required Content section, copy and paste this code:

```
(function (selectedIds, retUrl) {var head =
document.getElementsByTagName( 'HEAD' ).item( 0 );var script =
document.createElement ( "script" );script.type =  "text/javascript" ;head.a
ppendChild(script);script.onload = function ()
{MassUpdate.massUpdate(selectedIds, retUrl);};script.src =  "/resource/
CRMC_MassUpdate__MassUpdate?ts="  +  new  Date().getTime();}({!
GETRECORDIDS( $ObjectType.<objectname>)},window.top.location.href));
```

You must insert your own Object name in the highlighted area. I.E, a Mass Update button for Accounts would have to read ($ObjectType.Account).

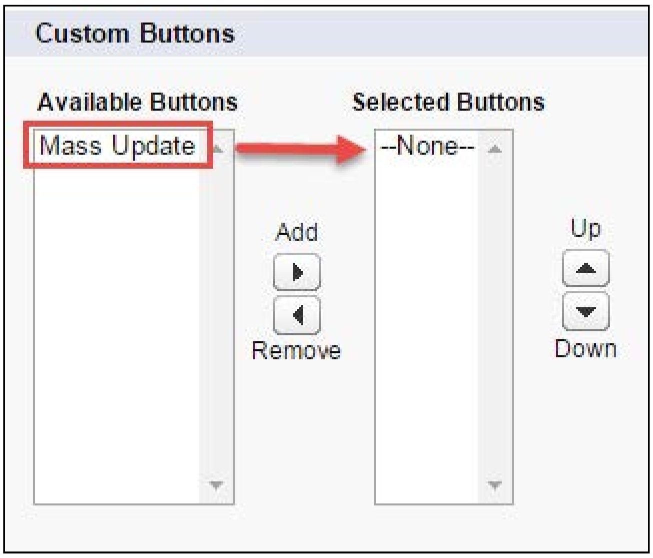
Next, we must add this button to the Search Layout. Do this by going to (Object name) > Search Layouts. Then Edit the "Search Results" and "List View."



Account Search Layouts

Action	Layout	Columns Displayed	Buttons Displayed
Edit	Search Results	Account Name, Account Site, Phone, Account Owner Alias	
Edit	Lookup Dialogs	Account Name, Account Site, Account Owner Alias, Type	N/A
Edit	Accounts Tab	Account Name, Billing City, Phone	N/A
Edit	Accounts List View	N/A	New
Edit	Search Filter Fields		N/A

Under the Custom Buttons list, "Mass Update" should appear under the Available Buttons list. Add it to the Selected Buttons list and click Save. Make sure that this is done in both the "Search Results" and "List View" Search Layouts.

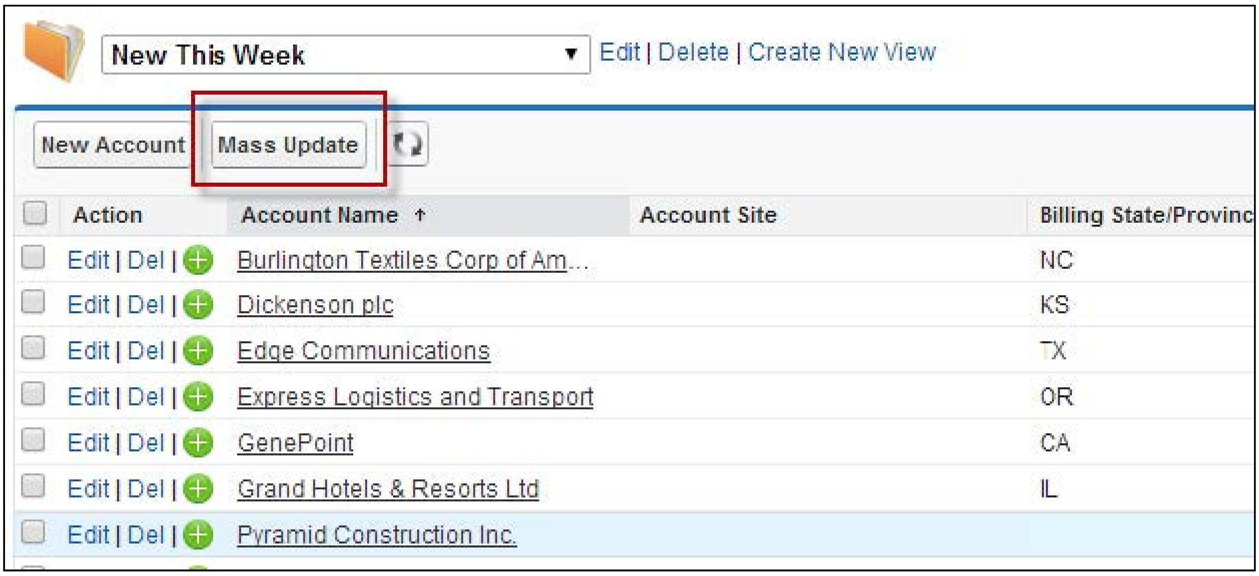


Secure Mass Update End User Experience

Warning

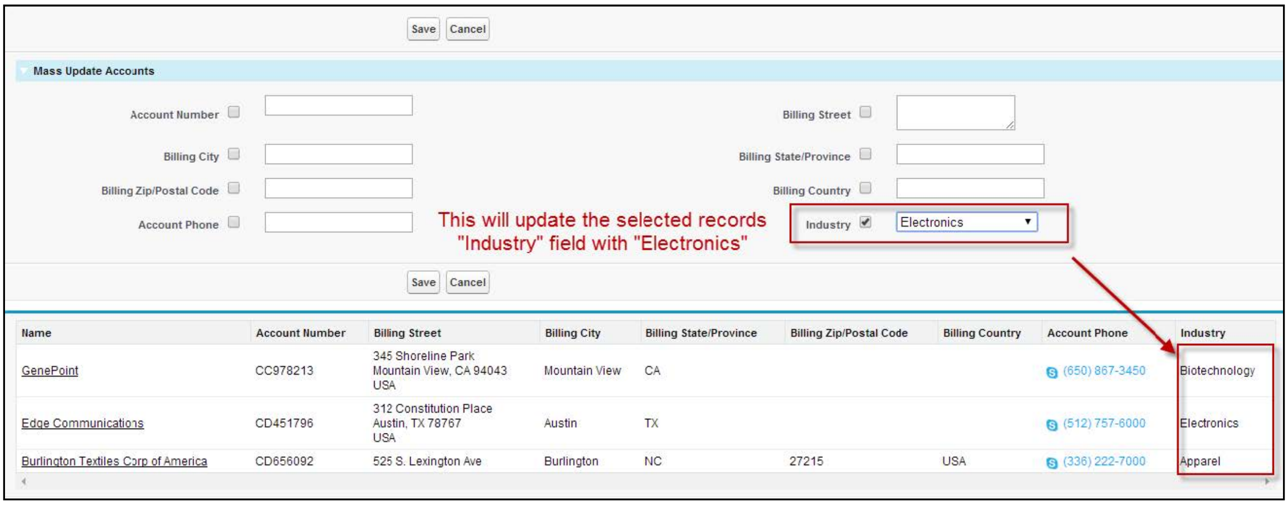
Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.

Depending on how you've configured the Mass Update buttons, your users will see the button appear if they satisfy the security requirements. See an example below:



□	Action	Account Name ↑	Account Site	Billing State/Provinc
□	Edit | Del | ⊕	Burlington Textiles Corp of Am...		NC
□	Edit | Del | ⊕	Dickenson plc		KS
□	Edit | Del | ⊕	Edge Communications		TX
□	Edit | Del | ⊕	Express Logistics and Transport		OR
□	Edit | Del | ⊕	GenePoint		CA
□	Edit | Del | ⊕	Grand Hotels & Resorts Ltd		IL
□	Edit | Del | ⊕	Pyramid Construction Inc.		

When selecting one or multiple rows with the standard Salesforce.com checkboxes in the search grid, the Mass Update Edit page appears. The fields that appear are those selected from the defined field set:



Account Number □

Billing City □

Billing Zip/Postal Code □

Account Phone □

Billing Street □

Billing State/Province □

Billing Country □

Industry ☑ Electronics ▼

This will update the selected records "Industry" field with "Electronics"

Name	Account Number	Billing Street	Billing City	Billing State/Province	Billing Zip/Postal Code	Billing Country	Account Phone	Industry
GenePoint	CC978213	345 Shoreline Park Mountain View, CA 94043 USA	Mountain View	CA			(650) 867-3450	Biotechnology
Edge Communications	CD451796	312 Constitution Place Austin, TX 78767 USA	Austin	TX			(512) 757-6000	Electronics
Burlington Textiles Corp of America	CD656092	525 S. Lexington Ave	Burlington	NC	27215	USA	(336) 222-7000	Apparel

Whether selected in the Field Set or not, the Object Name field will always appear as the first column in the preview grid so that you understand which records your editing.

It should also be noted that by checking a box next to a field but leaving the value blank, that is instructing the Mass Update Tool that you intend to have that field be set to null.

Apply Mass Update to Activities

Warning

Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.

Here are the steps on applying Secure Mass Update to the Activities object.

Tip

There is no standard object tab for Activities, as it is a combination of Events and Tasks. Because of this, the setup for Secure Mass Update adds a few more simple steps.

1. Clicking on the “Display Button” on Search Results will produce an error. This is to be expected, as Tasks and Events do not support Search Results. It also will not function for the List View They will have to be added manually.
2. Under the Secure Mass Update app menu, select the Secure Mass Update Settings Tab. You will need to make two separate Tables, one for Tasks and one for Events. With each Table, you will need to create an individual Field Set. Click the “Create Button” to populate it automatically.
3. You only need to Click Create Button for Events or Tasks, not both.
4. To add the Buttons to the Search Results and List Views, click Setup, go under the Build Menu, and under Customize open the Drop-Down Menu for Activities.
5. Click Activity Search Layouts, and go into Edit for Search Results. Under the Custom Buttons list, Add the Mass Update button to the Selected Buttons list.
6. Repeat this process for the Activities List View.

Now the Mass Update button will appear on the List View for Activities. Now, you have the ability to Mass Update Tasks and Events simultaneously!

Apply Mass Update to a Related List

❗ **Warning**

Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.

The Secure Mass Update application has the ability to update within a Related List. Let's say, for example, you want the ability to Mass Update Cases that a particular Contact is associated with. Start by adding the Mass Update button to the Case object. Since we want to update the Cases on the Contact page layout, we need to be able to Mass Update Cases first. To do this, follow the installation instructions in the "About Secure Mass Update" section of the App.

1. Go into Setup, go under the Build Menu, and under Customize open the Drop-Down Menu for Contacts. OpenPage Layouts and Edit the Contact Layout. Under the Related Lists, open the Properties menu for Cases.
2. Click on the Drop-Down for Buttons and Add the Mass Update button to the Selected Buttons list.
3. Now the Mass Update button will appear in the Case Related list for a selected Contact.
4. Now you can Mass Update Cases within a selected Contact.

Common Secure Mass Update Configuration Issue

❗ **Warning**

Secure Mass Update is no longer supported or available for installation. The current Secure Mass Update functionality is available in the latest Conga Grid solution.

I tried to launch Secure Mass Update and I get an error.

You do not have permission to use Secure Mass Update on this object. Please contact your System Administrator. The Secure Mass Update Setting for this object has either not been created and/or assigned to your user or profile.

What To Do

This is most commonly due to a Secure Mass Update setting for the object you launched from not being created or configured. You'll need to create a Field Set with the fields you want to Mass Update, go to the Secure Mass Update Settings tab, and create a new row with the Object you want to Mass Update. Set this row to the field set you just created, and grant it to the appropriate profile or user).

Troubleshooting Conga Grid

View frequently asked questions and known issues/workarounds when using Conga Grid℠.

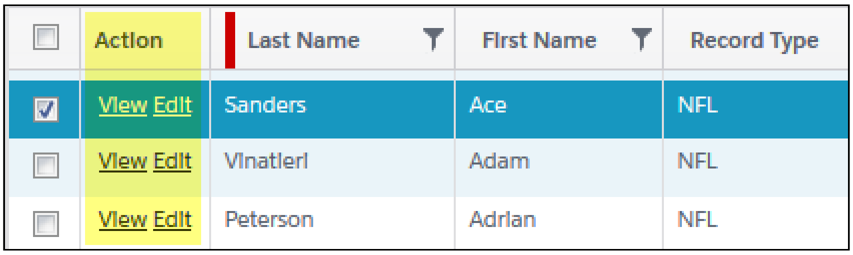
Conga Grid Frequently Asked Questions

This section lists frequently asked questions about Conga Grid℠.

FAQs on Using Conga Grid

What is the Action column, and how do I hide it?

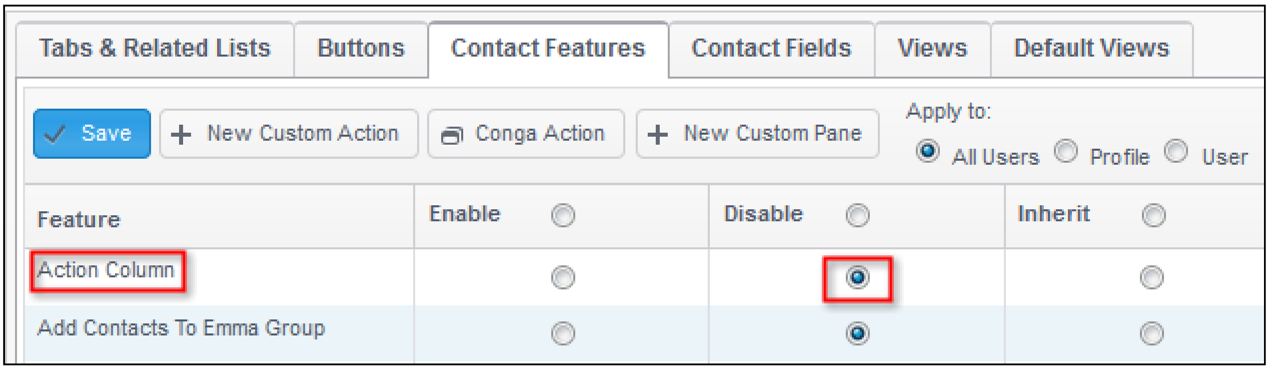
The Action column contains View and Edit hyperlinks for each record.



□	Action	Last Name	First Name	Record Type
☑	View Edit	Sanders	Ace	NFL
□	View Edit	Vinatieri	Adam	NFL
□	View Edit	Peterson	Adrian	NFL

When you select a record and click the View link, it displays a new web page with the details of that record. When you click the Edit link, it displays a web page that lets you edit the record details in Salesforce.

To hide this column, click Conga Grid Setup and click the object you are viewing. Select the object Features tab. Scroll to the Action Column feature and click the Disable option.



Apply to: ● All Users ○ Profile ○ User

Feature	Enable ○	Disable ○	Inherit ○
Action Column	○	●	○
Add Contacts To Emma Group	○	●	○

How can I tell what filters are currently applied to my record set?

In the Conga Grid Explorer, click the drop-down button next to Configure and click Filter Set. See Filter Logic for details.

If I created a filter for a column and move that header column off the Conga Grid, will my filter still be active?

Yes. You can use the Filter Logic dialog to confirm this. See the topic Can I Filter Fields that are Not Displayed? for detailed steps.

Can I batch-create cases from a list of contact search results?

Yes. Select the contact records, then right-click and select Batch Add > Cases.

Can I create a Conga Grid tab for any object?

Yes. See Single Conga Grid Tab for instructions.

How does Conga Grid manage two or more users working on a record at the same time?

Conga Grid behaves the same way that Salesforce does when two people edit the same record. If User A edits the record while User B also edits but saves first, then User A receives a warning that his changes will be lost. Two people can edit the same Conga Grid with no problem, but they must coordinate when working on the same records.

Does Conga Grid respond to duplicate rules?

Some Salesforce users have a rule that stops a user from creating a new Contact already in the system. Conga Grid honors duplicate checking rules, triggers, validation rules, and others. Conga Grid will display the same error as Salesforce if this is attempted.

Known Issue: Conga Grid displays with oversized icons in Service Console (Classic)

When using the Service Console in Salesforce Classic, some users viewing embedded Conga Grid pages may see oversized icons. To resolve this, set the showHeader attribute in the Visualforce <apex:page> containing Conga Grid Visualforce component to false (i.e., showHeader="false"). The page should display normally after applying this setting.

Can I Filter Fields that are Not Displayed?

A common misconception is that you must display all fields in the Conga Grid for which you have created a filter. Sometimes you just want to filter on a given field but not necessarily display the field because of limited screen space.

Conga Grid was designed for you to create as many filters as you need but not necessarily display the associated fields. Here is an example scenario: You are viewing a grid of Opportunities and you want to only see the records with a percentage value greater than 60%. You can temporarily add the Percentage (%) field to the grid, set the filter, then remove the field from the grid.

Follow these steps:

1. Click Opportunity in the object list.
2. Click the drop-down arrow next to Configure and click Fields.
3. Select the Probability (%) check box. You will remove it later. For now, you must display it in order to create a filter.
4. Click the Filter icon in the Probability (%) column header.
5. Click Basic Filter.
6. Select Is greater or equal to, and enter a value of 60.
7. Click Filter. An orange Filter icon indicates that a filter has been applied to the field.
8. Click and drag the Probability (%) field off the grid to hide it.
9. The entire list view of Opportunities is still filtered by a probability of 60% or greater even though the Probability (%) field is now hidden. To verify this, click the drop-down arrow next to Configure and click Filter Logic. The Filter Logic screen shows all of the current filters.

General FAQs

1. **What is Conga Grid?**
   Conga Grid is a revolutionary platform designed to improve the way every user in your organization works with Salesforce list views and related lists by offering a modern grid that resembles Microsoft Excel®.
2. **Which editions of Salesforce does Conga Grid support?**
   Conga Grid works with the following Salesforce editions: Enterprise, Unlimited, Developer, and Performance. Conga Grid also supports Professional Edition with only minor limitations when performing certain configuration tasks.
3. **Is Conga Grid Lightning ready?**
   Yes, Conga Grid was one of the first ten applications to be Lightning Certified.
4. **Does Conga Grid work with Salesforce1?**
   Yes, Conga Grid is Salesforce1 Mobile certified.
5. **Does Conga Grid work with Custom Objects?**
   Yes, the Salesforce API allows you to interact with any object and its relationships.
6. **How do I receive Conga Grid updates?**
   Conga Grid administrators are notified via e-mail about upcoming changes, as well as any necessary installation and configuration steps for that particular upgrade.
7. **What is the cost of Conga Grid?**
   Please use our Contact form to ask about specific pricing for your organization.
8. **Can I use a trial version of Conga Grid before purchasing it?**
   There are several ways to use a trial version of Conga Grid, available from

the AppExchange. Download and install Conga Grid to your production or sandbox org. Then take Conga Grid for a test drive! Everything is pre-configured.

9. **Do you have volume discounting, and can I pay month-by-month?**
   Please use our Contact form to ask about specific pricing for your organization.
10. **Who do I contact if I have an issue?**
    Please visit the Contact Support web page.
11. **How do I provide comments and feedback?**
    We love feedback! Our product roadmap depends on your ideas. Please visit the Review section for Conga Grid on the AppExchange to leave a product review. Or Contact Support.

Grid Visualforce Page vs Grid View

Grid as a Visualforce Page

- The Grid operates through a Visualforce page. For more information about the elements of a Grid Visualforce Page, see Manually Updating a Grid VisualForce page.
- Grid Visualforce pages are where component values are set. For example, if a System Administrator wants to disable the View Drop Down or the Edit/Save button on a Grid for all users, it is done in the Visualforce page. Grid Views
- The Grid View is a custom Object created in Salesforce and determines what fields, filters, Conditional Formatting Rules, and other information appear on the Grid. Whenever a Grid View is created, a new record is created under the Grid View object. Users can take this View ID and add it to the ViewID section of the Visualforce page to have that Grid automatically load that specific view as a default. For more information, see Hard-Code a Default View using a View ID.
- The Grid View object is accessible in Salesforce, or through Grid Explorer, and all Views created are visible based on Salesforce Object Permissions.
- Grid views contain a Privacy field that determines what users can access the View. The default Privacy value is Private, but users can set it as Public or Public Read-Only.
- Grid Views set to Private are typically only visible to the User that created the view. However, this is overridden by any User that has Shared Visibility to the View or has access granted via Role Hierarchy. See the Sharing Settings for Conga Grid View to learn more.
- Grid Views contain a Context field which determines if the view is available for a Visualforce Page and the Grid Explorer, or if the view is only available for a related list Grid on a Page Layout. A view with the Context "Page Layout" is only available for Related List Grids, not the Grid Explorer.

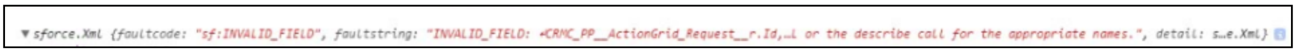
Conga Grid Known Issues and Workarounds

This section lists known issues, error messages, and workarounds when using Conga Grid℠.

Accessing Your Grid Failed due to Permission Set Issue

Issue

Users cannot access a Grid or the Conga Grid Setup object. Users may also see a perpetual loading icon. If users see the spinner and open the browser console, they may see an "INVALID_FIELD" error like the following example:



```
▼sforce.Xml {faultcode: "sf:INVALID_FIELD", faultstring: "INVALID_FIELD: ←CRMC_PP__ActionGrid_Request__r.Id,…l or the describe call for the appropriate names.", detail: s…e.Xml}
```

Cause

Salesforce is deploying a critical update that requires users to have the Customize Application permission enabled in order to obtain read access for custom settings. To prevent access issues in the future, Conga now uses a custom object instead of custom settings.

For more information see, Require Customize Application Permission for Direct Read Access to Custom Settings.

Resolution

- Ensure that Conga Grid users are assigned the Conga Grid User Permission Set or a custom Permission Set with the required permissions.
- Ensure that the Conga Grid permission set's field level security for the Conga Grid Settings object is enabled and have the required Read/Edit access.

To enable Read/Edit access to the Conga Grid Settings object:

1. Navigate to the Salesforce Setup.
2. Under the Users section, click Permission Sets.
3. Click the specific Permission Set used for Conga Grid users.
4. Click Object Settings.
5. Click Conga Grid Settings.
6. Click Edit and enable the Read permission under Object Permissions. Additionally, enable Read Access for all objects under Field Permissions.

7. Click Save.

Browsers and Kendo limitations

Browser and Kendo known limitations

Mozilla Firefox and Safari users on Mac machines cannot select default checkboxes in Conga Grid. If you select the default checkbox next to any public view in Firefox or Safari on a Mac machine nothing happens.

To avoid the issue, use the Google Chrome browser.

Import and export inherit setting issue

Import and Export setting issue

Imported data does not overwrite existing settings in a file. The export feature does not export Conga Grid's inherit settings.

Conga Grid Blank Account Values

When Conga Grid calculates column aggregates, blank values are skipped.

If you have three accounts but want the average of their Annual Revenue, and one of the Accounts has a blank value in the Annual Revenue field, the grid will display the average across only two of the accounts and not include the blank value in the calculation.

Conga Grid Encrypted Fields Permissions

Cloning Encrypted Fields

Cloning Encrypted Fields will result in error unless users have the View Encrypted Data permission enabled. The workaround for standard users is to uncheck the encrypted field when they select the record fields to be cloned.

Encrypted Fields in Custom Actions

Cloning Encrypted Fields will result in error unless users have the View Encrypted Data permission enabled. The only way to create records with encrypted fields is to have the View Encrypted Data permission enabled.

Conga Grid Error Message: Unable to Connect to the Server (Transaction Aborted: Timeout)

ERROR MESSAGE

Unable to connect to the server (transaction aborted: timeout)

CAUSE

The error can occur with objects that have a large number of custom fields that might not have proper database indexes. A timeout indicates that the query is taking too long and is encountering a Salesforce timeout limit. See the list of possible Salesforce limits. This error is not specific to Conga Grid; it also occurs if you run a similar SOQL query in Salesforce.

WHAT TO DO

Potential workarounds are available when working in Conga Grid. However, if you see a performance issue with custom or standard objects with custom fields, you should first determine if you need to add custom indexes to fields that are being sorted or filtered upon. Consider contacting Salesforce support for help with the following:

- Troubleshooting query performance problems.
- Adding a custom index to certain field types. See Improve performance with custom indexes.

To improve overall performance in Conga Grid:

1. Clear any sorting you have applied. By default, Conga Grid ships with no sorting applied to objects. This step alleviates some timeout issues. To remove a sort, click a column header that is sorted until you no longer see the arrow.
2. Ensure that the Salesforce Pod the running user is on matches the country they are in. If a user is attempting to load resources from an instance of Salesforce in another region, performance degradation can occur.
3. Reduce the Conga Grid page size to 50 items per page. This is the smallest option available.
4. Apply a filter on a known indexed field to reduce the total number of records being evaluated. If the field is not indexed, it could slow down performance. If smart filters are also timing out, apply a basic filter. To do this, click the filter icon on any column, then switch to Basic Filter.

Once the result set is less than 10,000, a timeout error should not occur. You can attempt to sort the records now. Smart filters should also work again once the dataset is filtered, as they only show results that match within the current dataset.

Performance issues can also be intermittent with your Salesforce instance. Refer to trust.salesforce.com to check for any known performance degradation. If performance is still unacceptable, refer to the resources below to determine if you need to add custom indexes to problem fields. If you suspect the performance problem is specific to Conga Grid, Contact Support. If you still cannot resolve the issue, we recommend submitting a case to Salesforce support to ask for a query performance evaluation. (Conga Grid can provide an example SOQL query to help diagnose the issue.)

To file a support case:

1. Review the knowledge article from Salesforce on custom indexes.
2. Review the checklist for Salesforce requests.
3. File a case with Salesforce support.

Conga Grid Error Message: Unable to Create/Update Fields (AccountID) or Name ID Value of Incorrect Type

ERROR MESSAGE

Unable to create/update fields: AccountID. Please check the security settings of this field and verify that it is read/write for your profile or permission set.

Or:

Name ID: id value of incorrect type.

CAUSE

When attempting to create tasks inline in a related list Conga Grid, the grid may refresh and you can lose data that you entered. You may receive the error messages listed above. These errors and loss of data are caused by an issue with Task Who IDs and What IDs. When Person Accounts are active, tasks can treat the Foreign Key lookup from Accounts to Tasks as a ***`WhoID`***.

WHAT TO DO

To modify the Visualforce Page Markup of the related list to use a ***`WhatID`***:

1. In Salesforce, click Setup.
2. In the App Setup area, click Develop and click VisualForce Pages.
3. Locate the VisualForce page and click its Label link.
4. Click Edit.
5. Click the Visualforce Markup tab.
    - For a single related list Visualforce page:
        - Find the value ***`FKName="WhoId"`*** or ***`FKName="<ObjectId>"`***.
        - Change the value to ***`FKName="WhatId"`***.

- For a MultiTab Visualforce page:
    - Find the Tasks tab in the `<li>...</li>` block of tags, and note the position in the list.
    - Find the same position in the `<div>...</div>` block of tags.
    - In that DIV section, replace `FKName = “AccountId”` or `FKName = “WhoId”` with `FKName=”WhatId”`.

6. Click Save.

Conga Grid Scope for "My Account Team and My Accounts" does not retreive Records

Issue

The Conga Grid filter for "My Account Team and My Accounts" does not show the proper records when the Scope is set.

Cause

Salesforce SOQL queries do not support "My Account Team and My Accounts." SOQL queries are designed for single scope queries. For more information, see: Using Scope

Workaround

Use traditional Conga Grid filters to retrieve the desired record set.

Conga Grid Multi-Currency Parenthesis Issue

When an Org has Multi-Currency enabled, users will see formatting that resembles the following example:

USD 350 (GBP 263.98)

The first value shows the currency that matches the Currency Code of the record, and the second value in parenthesis shows the converted value for the default currency of the Org.

This matches the default behavior seen in Salesforce Orgs with Multi-Currency enabled.

Issue

Due to Salesforce limitations, Conga Grid does not support Advanced Currency settings or the disablement of parenthetical values. There is no workaround at this time.

Activity History and Open Activities Implementation Restriction

PROBLEM

There is an implementation restriction on Activity History (or Open Activities). When you query the relationship, security evaluation is implemented for users who don't have administrator permissions, and only a single parent record might be evaluated.

CAUSE

Salesforce restricts the usage of Activities to users who do not have the View All Data permission. The restrictions are as follows:

- In the main clause of the relationship query, you can reference only one record. For example, you can't filter on all records where the account name starts with "A." Instead, you must reference a single account record.
- In the inner clause of the query, you can't use WHERE.
- In the inner clause of the query, you must specify a limit of 500 or fewer on the number of rows that are returned in the list.
- In the inner clause of the query, you must sort on ActivityDate in ascending order and LastModifiedDate in descending order. You can optionally display nulls last. For example: ORDER BY ActivityDate ASC NULLS LAST, LastModifiedDate DESC.

WHAT TO DO

There are two workarounds:

1. Give the users who need to see Activity History or Open Activities in the View all Data permission in their profile. See more here.
2. Change all references to Activities to Task and/or Events. You can do this by removing Activity History and Open Activities from related list and related columns and add Tasks and Event instead.

Browser Settings Prevent Conga Grid from Displaying

Problem

White space is displayed while the browser tries to load indefinitely.

Workaround

Steps for correcting the issue vary for each web browser:

- Google Chrome
- Mozilla Firefox
- Internet Explorer

Browser Settings Prevent Conga Grid from Displaying - Google

Browser settings in Chrome interfere with Grid loading in Related Lists in Page Layouts. White space is displayed while the browser tries to load indefinitely.

1. Click the button in the upper-right corner of the browser, and click Settings.
2. Click the Show Advanced Settings link at the bottom of the Settings page.
3. In the Privacy section, click Content Settings.
4. In the Cookies section, clear the Block third-party cookies and site data check box.
5. Click Done.

Grid will load properly in a Related List on a Salesforce Page Layout.

Browser Settings Prevent Conga Grid from Displaying - Firefox

Browser settings in Firefox interfere with Grid loading in Related Lists in Page Layouts. White space is displayed while the browser tries to load indefinitely.

1. Click the button in the upper-right corner of the browser, and click Options.
2. Click the Privacy tab.
3. In the History section, select Use custom settings for history from the drop-down menu.
4. Click Accept cookies from sites.
5. Make sure that the Accept third-party cookies option is not set to Never.
6. Close the Options page.

Grid will load properly in a Related List on a Salesforce Page Layout.

Browser Settings Prevent Conga Grid from Displaying - Internet Explorer

Browser settings in Explorer interfere with Grid loading in Related Lists in Page Layouts. White space is displayed while the browser tries to load indefinitely.

1. Click the Tools icon in the upper-right corner of the browser, and click Internet options.
2. Click the Privacy tab.
3. Click Advanced. The Advanced Privacy Settings screen appears.
4. Select the Override automatic cookie handling check box.
5. Click Accept in First-party Cookies and Third-party Cookies.
6. Click OK, then OK again to return to your browser.

Grid will load properly in a Related List on a Salesforce Page Layout.

Bulk Salesforce Triggers and Mass Update

PROBLEM

A Mass Update Results screen appears with the message: Mass Update Unsuccessful! The following errors occurred.

CAUSE

Apex CPU time limit exceeded

This is common when a trigger uses multiple FOR loops and Mass Update runs on a large amount of data. A workaround is to remove some of the FOR loops in the trigger and to try to map your data differently.

Insufficient access rights on object id

This error can occur when a trigger is not properly using a List. If the DML is called and triggers a secondary DML that modifies a parent record that is shared between two records being mass updated, only the initial DML triggers will work. The rest will show the error message.

DML or query limit reached

This error can occur when a trigger calls a DML, SOQL, or SOSL query in a FOR loop. Consider creating lists to replace some of these insides of the FOR loops.

Refer to the Salesforce documentation for more information on bulkification.

Campaign Member Option Appears Twice

PROBLEM

The Campaign Member option appears twice under the Batch Add action menu if:

- You have the Campaign Member Loader application installed
- You have Conga Grid version 1.49.143 or later.CAUSE

Conga Grid natively includes the Campaign Member Loader functionality, so you no longer need to install the application separately.

WORKAROUND

To resolve the issue of the Campaign Member menu option appearing twice, you can uninstall the Campaign Member Loader application or disable the feature security.

To disable the feature security for Campaign Member Loader:

1. Click Conga Grid Setup.
2. Click Global Defaults in the object list.
3. Click the Feature Security tab.
4. Click Disable next to Campaign Member Loader.

Cannot Select Lookup Values

PROBLEM

When you click the spyglass icon in a lookup cell, the Lookup window appears but you might not be able to select a lookup value.

CAUSE

This problem occurs in embedded related lists Conga Grids. This is an issue with legacy Visualforce pages created with the Conga Grid setup wizards for embedded related lists.

WORKAROUND

A solution is to change the *showHeader* attribute in the Visualforce markup code from *false* to *true*. This adds the necessary JavaScript code from Salesforce to ensure that the spyglass icon works correctly. The change does not display the Salesforce header while embedded in a page layout.

Conga Grid is using a disproportional number of API calls

Question or Issue

Conga Grid appears to be using a disproportional number of API calls.

Causes or Possible Causes

The Grid is displayed on a frequently visited paged which, by nature, is an API consumer.

Related Known Errors and Workarounds

It is highly recommended to use a layout that does not display a Grid on frequently visited pages.

Custom Objects Do Not Appear in Conga Grid Explorer

Follow these steps to correct the issue:

1. In Salesforce, click Setup.
2. Click Create and click Objects.
3. Click the Edit link next to the object in question.
4. Scroll to the bottom of the page and select the Allow Search check box.
5. Click Save, then refresh the Conga Grid Explorer. The custom object will appear in the object list.

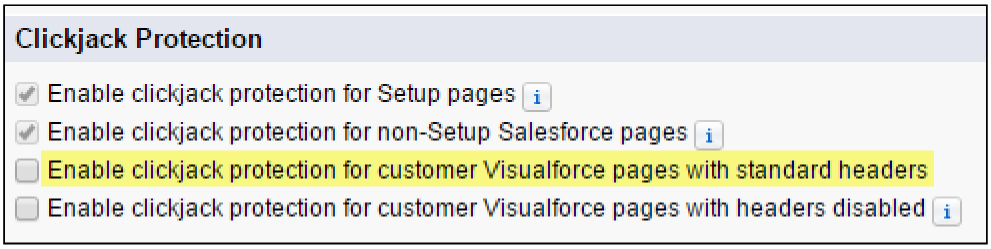
Embedded Conga Grids Do Not Display

Sometimes an embedded Conga Grid does not display in a Service or Sales Console. Or, it won't display in a page layout when the organization ("org") has a custom URL; for example, mycompany.salesforce.com instead of salesforce.com. The reason for this may be due to a Salesforce clickjacking security setting.

Clickjacking refers to a method of tricking you into clicking a button or link on a Web page that appears legitimate. Clicking it can expose your system to malicious code. Salesforce has security settings that protect your Visualforce pages against this activity. However, some of these settings can interfere with apps such as Conga Grid that use inline frames to display content.

Follow these steps:

1. In Salesforce, click Setup.
2. Click Security Controls and click Session Settings.
3. In the Clickjack Protection section, clear the third check box highlighted below: This setting could be enabled, for example, if you run a Health Check in Salesforce and choose to increase your security level. Note that the first two settings are enabled by default and cannot be modified by Salesforce end users.



Clickjack Protection

☑ Enable clickjack protection for Setup pages
☑ Enable clickjack protection for non-Setup Salesforce pages
☐ Enable clickjack protection for customer Visualforce pages with standard headers
☐ Enable clickjack protection for customer Visualforce pages with headers disabled

4. Click Save at the bottom of the Session Settings page.

Error Message Accessing Conga Grid Documentation in the App

Issue

You see the following error message when visiting the legacy documentation website.

Causes

The legacy documentation site is no longer available. The removal of the site causes old packages to display an error message where legacy articles existed.

Workarounds

You must upgrade to the latest version of Conga Grid from the AppExchange .

Error Showing when Exporting to Excel from Lightning

Question or Issue

When you export to excel in lightning, you receive the following error: The file format and extension of 'download.xls' don't match. The file could be corrupted or unsafe. Unless you trust its source, don't open it. Do you want to open it anyway?

Causes or Possible Causes

This is a Microsoft error that is caused as a result of the method used to export information from Salesforce into excel.

Related Known Errors and Workarounds

Because the error is produced by Microsoft, we cannot prevent it from firing. The workaround is to accept the warning and the document opens as expected.

Filtered Lookup Limitations in Communities

Limitation

There is a limitation with filtered lookup fields in Communities when using Conga Grid. If you attempt to edit a filtered lookup field's value in Conga Grid, nothing appears because Community users do not have access to Salesforce's Tooling API. This stems from current Community User access limits.

Cause

The Community users do not have access to the Tooling API. Access to the Tooling API is part of the Salesforce set up for Community Members.

Filtered Lookup Showing All Values for Force.com License User

PROBLEM

When a Force.com Licensed User attempts to edit a filtered lookup field, the values are not filtered correctly and all values are shown as available.

CAUSE

This problem occurs for users with the Force.com license type due to a limitation the license has accessing the Salesforce API.

WORKAROUND

A solution is to change the user license from a Force.com to a standard Salesforce license.

Getting the Best Performance from Conga Grid

When Conga Grid displays for the first time, several factors can potentially increase the page loading time:

The number of records per page. Conga Grid can display 1,000 records per page but the display time will be slower than 100 records per page. Start with 500 records per page, then decrease the number if performance becomes an issue.

Complex filtering. This is the most common reason for a delay in page loading time. If the view is so slow that you are receiving timeout errors, you may need to filter additional fields or even ask Salesforce support to set a specific index on the particular field you are filtering. For more information, see: Conga Grid Error Message: Unable to Connect to the Server (Transaction Aborted: Timeout)

Multi-field sorting. This type of sorting is done on the server side by Salesforce SOQL statements, and it can have a minor impact to your query results.

Too many fields displayed in the grid. Displaying hundreds of fields in the grid can impact performance. Consider using the Detail Grid Tab in the Reading Pane to display details about selected records.

Parent fields. Displaying many fields with foreign keys, or fields from those parent objects, can impact performance.

Displaying the Reading Pane with a large related list as the first tab. In this case, Conga Grid needs to load the queries for the Reading Pane and the related list, which impacts performance.

Long text fields. Displaying many long text fields in the grid can cause a minor delay. Consider using the Detail Grid Tab in the Reading Pane.

Editing and Saving Data - You can only edit within the bounds of what Salesforce allows, and what has been configured in your Salesforce Org.

Conga Grid as related lists on embedded page layouts:

- Set DelayLoad="true" in the Visualforce Markup for all tabs other than the first tab in a Multi-Tabular Related List.
- Avoid multiple VisualForce pages such as one page that calls another page.

As the volume of data increases, optimized queries become more crucial for sorting and filtering of data. Conga Grid provides considerable freedom in sorting and filtering of your data. This may result in queries retrieving data in a less efficient manner. Because Conga Grid uses Salesforce SOQL queries, it is subject to the same performance limitations. Please review the resources below to ensure your data queries, through the sort and filter selection you make in Conga Grid, are optimized to return information as fast as possible.

- Write Efficient Queries (Trailhead)
- Query and Search Optimization Cheat Sheet
- SOQL Best Practices
- Large Data Volumes
- Query Plan Tool

Headers do not translate in other languages

Problem

Some headers of columns do not translate into another language or locale.

Causes

All new and saved views created or updated in Conga Grid version 1.79.9 or later receive a new setting that stores the language of the user who created the view.

If you load a view with the setting, Conga Grid translates the column headers when your language does not match the language saved in the selected view.

Important

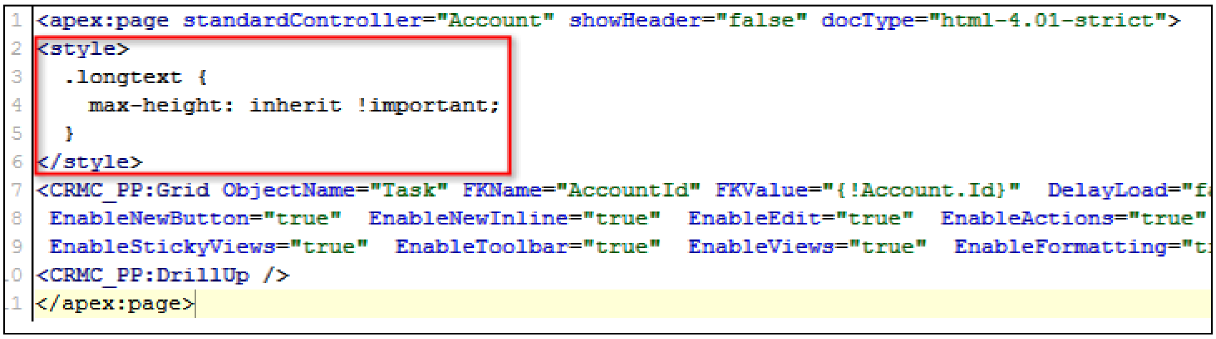
When a view is localized, Conga Grid does not save the Column Setting's re-labeled text.

If you modified a view before the view was updated by the translation language setting, the sticky view cached in your browser can prevent the text from being translated.
Click Reset View to address this issue.

Long Text Fields Only Display Two Lines of Text

Long-text fields (such as Description) only display two lines of text by default. To display more than two lines, you must modify the markup code in the associated Visualforce page. Follow these steps:

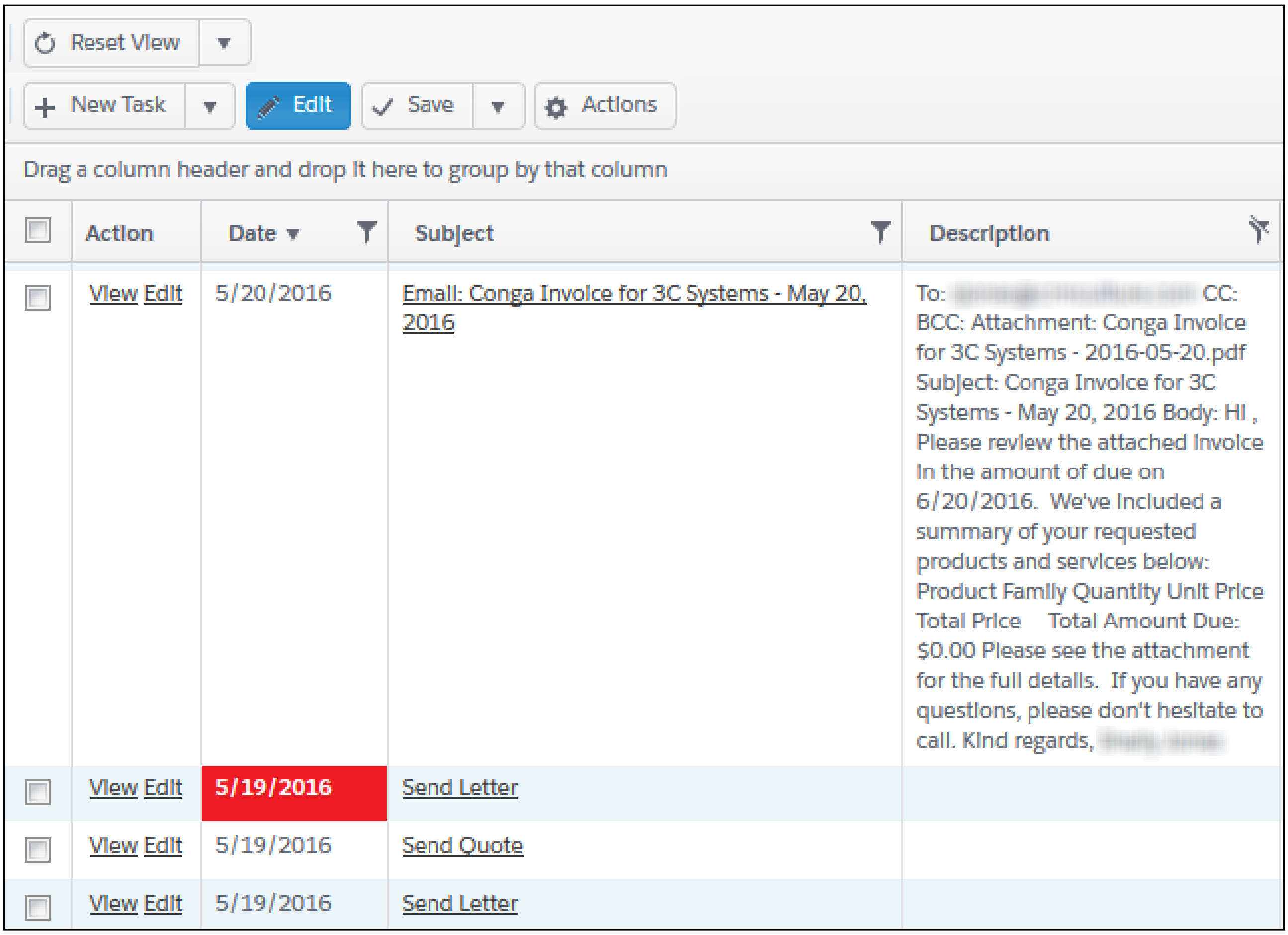
1. In Salesforce, click Setup.
2. In the App Setup area, click Develop and click Visualforce Pages.
3. Find the name of the Visualforce page associated with your view, and click its Edit link.
4. Add the following code after the starting
   <apex> tag:<style> .longtext { max-height: inherit !important; }</style>
5. For example:



```
<apex:page standardController="Account" showHeader="false" docType="html-4.01-strict">
<style>
  .longtext {
    max-height: inherit !important;
  }
</style>
<CRMC_PP:Grid ObjectName="Task" FKName="AccountId" FKValue="{!Account.Id}"  DelayLoad="f
 EnableNewButton="true"  EnableNewInline="true"  EnableEdit="true"  EnableActions="true"
 EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="t
<CRMC_PP:DrillUp />
</apex:page>
```

6. Click Save.

Here is an example of the result:



Drag a column header and drop it here to group by that column

□	Action	Date ▾	Subject	Description
□	View Edit	5/20/2016	Email: Conga Invoice for 3C Systems - May 20, 2016	To: [redacted] CC: BCC: Attachment: Conga Invoice for 3C Systems - 2016-05-20.pdf Subject: Conga Invoice for 3C Systems - May 20, 2016 Body: Hi , Please review the attached Invoice in the amount of due on 6/20/2016. We've included a summary of your requested products and services below: Product Family Quantity Unit Price Total Price Total Amount Due: $0.00 Please see the attachment for the full details. If you have any questions, please don't hesitate to call. Kind regards, [redacted]
□	View Edit	5/19/2016	Send Letter	
□	View Edit	5/19/2016	Send Quote	
□	View Edit	5/19/2016	Send Letter	

MIME types not supported with Salesforce

PROBLEM

You might see the following error if Enable Content Sniffing protection is selected under the Content Sniffing protection section in Salesforce Setup:

Refused to execute script from 'https://www.abc.com/resource/1225399...enametoinclude' because its MIME type ('text/plain') is not executable, and strict MIME type checking is enabled.

CAUSE

When a user creates a Conga Grid custom action, a custom MIME type is automatically associated with it which is not supported in Salesforce security because the MIME type is not standard.

If a custom action is not showing under the actions menu even when it has been enabled in Conga Grid Setup, review your developer console:

- Chrome: Chrome Menu → More Tools → Developer Tools → Console or hit the F12 key.

- FireFox: FireFox Menu → Developer → Browser Console or hit Ctrl + Shift + k
- Edge and IE: Edge/IE Menu → F12 → Developer Tools or hit the F12 key

WORKAROUND

To fix the issue, go into Salesforce and select Setup - > Security Controls - > Session Settings and then uncheck the Enable Content Sniffing protection checkbox under the Content Sniffing protection section.

Mixed Case Text Fields Causing Incorrect Display in Filter Dropdown

Question or Issue

Text Fields with mixed-case values result in an incorrect value display for filter dropdowns.

If one record has a Billing Country of USA and another has usa, they are counted as 2 different values but appear in the filter dropdown as only one. This causes the total number of values and the number of checkboxes to be different.

Related Known Errors and Workarounds

Check the Select All checkbox on a mixed-case field to display the proper results, but when you open the filter again, Select All will be unchecked.

Multi-Tabular Grid Trouble Loading in Safari Mobile

Multi-Tabular Grids fail to load with Safari browsers on Mobile devices. To resolve this, the Visualforce Markup can be modified with a document ready addition.

Navigate to the Visualforce page by going to Salesforce Setup → Develop → Visualforce Pages and select your Grid MultiTab Visualforce page. Click Edit.

Scroll to the bottom of the Visualforce markup and locate the following segment:

```
<script>
$(document).ready(function() {
    $( "#tabstrip" ).kendoTabStrip({
      activate: function(e){
        setTimeout(function(){          $(window).resize();        });
      }
    });
    $( "#tabstrip" ).css( "display" ,  "block" );
});
 </script>
```

Replace the above segment with the following code:

```
<script>
setTimeout(function(){
       $( "#tabstrip" ).kendoTabStrip({
         activate: function(e){
           setTimeout(function(){          $(window).resize();        });
         }
       });
       $( "#tabstrip" ).css( "display" ,  "block" );
   },  500 )
});
 </script>
```

Once this is done, MultiTab Grids load properly on Mobile devices using the Safari browser.

Send Email from Task Object: Batch Failed Error

PROBLEM

When Sending an Email from a Task Object the user receives a "Batch Failed Error" message.

CAUSE

This is because the Task object has many types and can also be linked to Users or Contacts and the fields are linked dynamically by Salesforce.

WORKAROUND

Salesforce API documentation says "This field contains a value only if the related record is a user." Sending an Email from a Task Object cannot be supported due to Salesforce API limitations. See API Developer Guide for more information.

Style Tags for Embedded Grids

Question or Issue

You have an embedded Conga Grid through the use apex:iframe and interacting with the Grid results in the page bouncing back to the top.

Workaround

To correct this behavior, the following lines should be added using styling tags:

@media only screen and (max-device-width: 768px) { html,body { height: 100vh; overflow: auto; -webkit-overflow-scrolling: touch; } }

This will handle mobile detection and apply the proper styling for the page.

Issue with Using Conga Grid in Microsoft Edge 17

Issue

When using Conga Grid in Microsoft Edge to edit information in a text field, Conga Grid may not retain the changes after focus is shifted from the field.

Cause

This is a known issue in Microsoft Edge 17 and ensuing versions. The issue prevents text input fields from storing information before another field is accessed. For more information, see Microsoft's documented issue.

Once the referenced issue is resolved, editing text fields should work as expected in Edge once more.

Workarounds

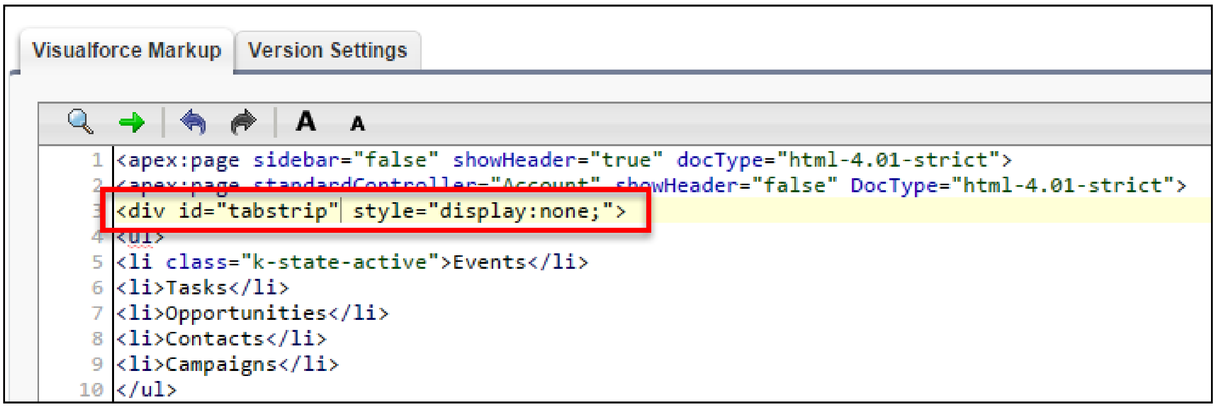
1. Use a web browser other than Microsoft Edge (Chrome, Firefox, Safari).
2. Use Microsoft Edge 16 or preceding versions.
3. After editing a text field, click outside of the grid before clicking into a new cell. This should retain the changes made.

Visualforce Page Performance

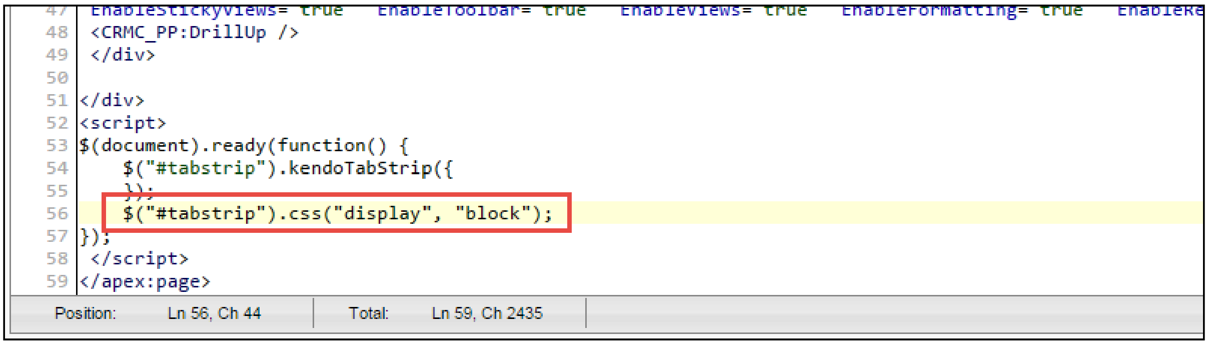
You can improve the performance of existing Visualforce pages created before Conga Grid version 1.49.61.

To improve Visualforce performance:

1. In Salesforce, click Setup.
2. In the App Setup area, click Develop → Data → Visualforce Pages
3. Find the Visualforce page you want to modify, and click its Edit link.
4. Click the Visualforce Markup tab.
5. Add the following lines of code:



```
<apex:page sidebar="false" showHeader="true" docType="html-4.01-strict">
<apex:page standardController="Account" showHeader="false" DocType="html-4.01-strict">
<div id="tabstrip" style="display:none;">
<ul>
<li class="k-state-active">Events</li>
<li>Tasks</li>
<li>Opportunities</li>
<li>Contacts</li>
<li>Campaigns</li>
</ul>
```



```
  EnableStickyViews="true"  EnableToolbar="true"  EnableViews="true"  EnableFormatting="true"  EnableRe
  <CRMC_PP:DrillUp />
  </div>

</div>
<script>
$(document).ready(function() {
    $("#tabstrip").kendoTabStrip({
    });
    $("#tabstrip").css("display", "block");
});
 </script>
</apex:page>
```

Position: Ln 56, Ch 44 Total: Ln 59, Ch 2435

6. Click Save.

Conga Grid Product Links

The following topics are all in-product links available within Conga Grid.

- Conga Grid Setup
- Conga Grid Explorer

Conga Grid Setup

Welcome to Conga Grid! This page is intended for configuring Conga Grid to show up on Tabs or embedded in Page Layouts. More on configuration can be found at Getting Started with Conga Grid.

Please refer to this video to get started with Conga Grid before you initially configure the product.

If at any point you need technical assistance, Contact Support.

To purchase Conga Grid, please contact us.

Conga Grid Explorer

Conga Grid EXPLORER

Welcome to Conga Grid!

To begin, please start by selecting an object on the left.

This page is intended to "kick the tires" of Conga Grid and see what your data looks like in this tool. It is not intended as the main entry point for most users. More on configuration can be found at Getting Started with Conga Grid.

If at any point you need technical assistance, please Contact Support.

To purchase Conga Grid, please contact us.

Grid Features by Release

Review the latest Conga Grid Features by Release document.

- Features by Release

Features by Release

This document contains an overview of features introduced in each major release of Conga Grid. For more information, see Conga Grid Features by Release.

Conga Copyright Disclaimer

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.