

APTTUS[®]

Contract Management on Salesforce Winter 2019 SOAP API Guide

11/29/2019

Table of Contents

About this Guide	4
What's New.....	5
Overview	6
API Supported Packages.....	6
Document Setup.....	6
API Standards and Development Platforms	7
Standards	7
Development Platforms	7
Field Types	7
API Reference.....	9
Agreement Web Service.....	9
Activating an Agreement.....	10
Publishing Document Outside of Salesforce	10
Publishing the Document to Chatter.....	11
Publishing the Document to Files and Attachments.....	12
Selecting a Template for the Agreement Document	12
Terminating Agreements	13
Merge Web Service	13
Adding Text to a Header and Footer.....	15
Applying Security Parameters to a PDF.....	17
Converting a Word Document into a given output format	19
Converting a Word Document into a given output format (Deprecated)	20
Converting a Word Document into a given output format for Standard Object (Deprecated)	23
Converting a Word Document into a PDF.....	25
Converting a Word Document into a PDF for Standard Objects	26
Enabling Authoring for Agreement Documents	29
Generating a Document.....	31
Generating a Document for a Standard Object	34
Generating a Document on Submit Request.....	35

Generating a Document on Submit Request for a Standard Object	38
Generating a Supporting Document.....	40
Generating a Supporting Document on Submit Request.....	44
Merging Multiple Documents to a PDF	46
Merging Proposal Documents into a single PDF Document.....	54
Package Multiple Documents	57
Author Web Service.....	61
Getting Word Document Information for Parent with Files.....	61
Initiate Parallel Review	62
End Parallel Review	63
Callback Classes	63
IAgreementLifecycleCallback2 and IDynamicSectionCallback2.....	63
Apttus Copyright Disclaimer	68

About this Guide

Apttus provides Application Programming Interfaces (API) for you to extend the features offered by Apttus. These extensions add more functionality to the features available through configuration on Salesforce.

The Apttus Contract Management API Reference Guide describes the APIs used to publish and manipulate the document. It also provides information on creating custom logic.

What's New

The following table lists changes in the documentation to support each release.

Release	Topic	Description
Winter 2019	Merging Multiple Documents to a PDF	Modified topic.
	Terminating Agreements	New topic.
Summer 2019	Initiate Parallel Review	New topic.
	End Parallel Review	New topic.
Spring 2019	Callback Classes	Added Contract Management callback classes.
Winter 2018	No updates	No new APIs are introduced in this release. This guide is updated to reflect the product name change.

Overview

- [API Supported Packages](#)
This guide covers Contract Management and Merge Service APIs.
- [Document Setup](#)
The Apttus Contract Management API Reference Guide details the APIs that you can use to manipulate Apttus objects through API calls and passing parameters.
- [API Standards and Development Platforms](#)
Apttus APIs are based on Salesforce APIs and use the same standards and platforms.
- [Field Types](#)
Apttus APIs use a subset of the supported data and field types on Salesforce.

API Supported Packages

This guide covers Contract Management and Merge Service APIs.

Package	Latest Certified Version
Apttus Contract Management (New)	10.2.0491 10.491

Document Setup

The Apttus Contract Management API Reference Guide details the APIs that you can use to manipulate Apttus objects through API calls and passing parameters:

API Reference	The <i>API reference</i> section details the APIs that you can use to manipulate Apttus objects through API calls and passing parameters. The API reference section also includes some code samples.
-------------------------------	--

API Standards and Development Platforms

Apttus APIs are based on Salesforce APIs and use the same standards and platforms.

Standards

Name	Reference
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Development Platforms

Apttus SOAP API works with standard SOAP development environments. For a list of compatible development platforms, see [Salesforce Developer Force API](#) details.

Field Types

Apttus APIs use a subset of the supported data and field types on Salesforce.

The following table lists the APIs that Apttus provides. For a comprehensive list of all field types supported by Salesforce, see [Salesforce Data Types](#).

Type	Description
Boolean	The Boolean field has a <code>true</code> (or 1) or <code>false</code> (or 0) value.
Data object	The Data Object field is an ID type and is represented by <code>CPQ.nndoin</code> in this document.
Date	The Date field contains date values only and do not contain relevant time values. Time in a date field is always set to midnight in the UTC time zone. If you want a timestamp you must use a dateTime field.
Decimal	The Decimal field provides an exact numeric value and you can arbitrarily size the precision and scale of the value.

Type	Description
ID	<p>The ID field is an alphanumeric field that acts like the primary key for a specific record associated with an object. The ID value includes a three-character code that identifies which object the record is associated with. The ID for a specific record does not change.</p> <p>For some objects, this field may also be a <code>referenceType</code> value, which contains the ID value for a related record. They are identified by field names ending in 'Id', such as <code>priceListId</code>. The ID field acts like foreign keys and their values can be changed using an <code>update()</code> call.</p>
Integer	<p>The Integer field contains whole numbers only. There are no digits after the decimal.</p>
List	<p>The List field includes a fixed set of values from which you must select a single value. Picklists are available as drop-down lists. If a picklist is unrestricted, the API does not limit entries to only currently active values.</p>
String	<p>The String field contains text and may have differing length restrictions based on the data you store in the specific field. For instance, <code>city</code> may be limited to 50 characters, while <code>addressLine1</code> is limited to 255 characters.</p>

API Reference

- [Agreement Web Service](#)
The Agreement Web Service APIs are used for activating, selecting and publishing agreements. Once an agreement has been activated and associated with a template it can be published to [Salesforce.com](#) – either to Chatter files or Documents - or published to an external repository.
- [Merge Web Service](#)
The Merge Web Service APIs are used for generating and manipulating documents. Both agreements and supporting documents can be generated, while text can be added to document headers and footers. Word documents can be converted into PDF documents and security rules can be applied to PDF documents.
- [Author Web Service](#)
The Author Web Service API contains all the services related to files or documents and can be used with various operations related to X-Author.

Agreement Web Service

The Agreement Web Service APIs are used for activating, selecting and publishing agreements. Once an agreement has been activated and associated with a template it can be published to [Salesforce.com](#) – either to Chatter files or Documents - or published to an external repository.

The APIs enable you to do the following:

Activating an Agreement

This activates an agreement and can also be used to delete related draft versions. Once activated an agreement document can be published to multiple channels.

Selecting a Template for the Agreement Document

This associates a template with the agreement, enabling the agreement document to be published.

Publishing the Document to Chatter

This publishes the agreement document for the activated agreement to Chatter > Files for the Salesforce object the agreement is associated with.

Publishing the Document to Files and Attachments

This publishes the agreement document for the activated agreement to Files and Attachments for the Salesforce object the agreement is associated with.

Publishing Document Outside of Salesforce

This publishes the agreement document for the activated agreement to an external location. This method will write the data into an integration object. A trigger should be written to send the document to the external location.

Activating an Agreement

This activates an agreement and can also be used to delete related draft versions. Once activated, an agreement document can be published to multiple channels.

activateAgreement

Parameters		
Name	Type	Description
agreementId	ID	The id of the agreement.
selectedDocIds	List<ID>	List of fully signed document ids.
removableDocIds	List<ID>	List of draft document ids that can be deleted.

Response		
Name	Type	Description
result	Boolean	Indicates whether the agreement was activated successfully.

Code Sample

```

1  Id agreementId = agreement.Id;
2  String[] activateDocIds = new String[1];
3  String[] remDocIds = new String[]{};
4  Boolean response = Apttus.AgreementWebService.activateAgreement(agreementId,
    activateDocIds, remDocIds);

```

Publishing Document Outside of Salesforce

This publishes the agreement document for the activated agreement to an external location. This method will write the data into an integration object. A trigger should be written to send the document to the external location.

publishToExternalRepository

Parameters		
Name	Type	Description
agreementId	ID	The id of the agreement
attachmentId	ID	The id of the document that is to be published.

Response		
Name	Type	Description
result	Boolean	Indicates whether the document was published successfully.

Publishing the Document to Chatter

This publishes the agreement document for the activated agreement to *Chatter > Files* for the Salesforce object the agreement is associated with.

publishToChatterRepository

Parameters		
Name	Type	Description
agreementId	ID	The id of the agreement
attachmentId	ID	The id of the document that is to be published and uploaded to Chatter > Files.

Response		
Name	Type	Description
result	Boolean	Indicates whether the document was published to Chatter successfully.

Code Sample

```

1 Id agreementId = agreement.Id;
2 Id publishDocId= AttachmentId;
3 Boolean response = Apttus.AgreementWebService.publishToChatterRepository
  (agreementId, publishDocId);

```

Publishing the Document to Files and Attachments

This publishes the agreement document for the activated agreement to *Files and Attachments* for the Salesforce object the agreement is associated with.

publishToDocumentRepository

Parameters		
Name	Type	Description
agreementId	ID	The id of the agreement
attachmentId	ID	The id of the document that is to be published and uploaded to Files and Attachments.

Response		
Name	Type	Description
result	Boolean	Indicates whether the document was published to Files and Attachments successfully.

Selecting a Template for the Agreement Document

This associates a template with the agreement, enabling the agreement document to be published.

selectFirstTemplateIdForAgreement

Parameters		
Name	Type	Description
agreementId	ID	The id of the agreement

Response		
Name	Type	Description
result	ID	The id of the template that is used to generate the agreement document

Terminating Agreements

The `terminateAgreementsAsync` API allows you to bulk terminate agreements based on the agreement status.

terminateAgreementsAsync

Parameters			
Name	Type	Description	Required?
agreementIds	List<ID>	List of agreement Ids	Yes

Code Sample

```
List<ID> agreementIdList = new List<ID> ();
Boolean response = Apttus.AgreementWebService.terminateAgreementsAsync(agreementIdList);
```

Use Case

Bulk terminate agreements and their related agreements with the same agreement number.

Order of Execution

1. Check permissions.
2. Get the agreement object.
3. Register the terminate indicator.
4. Terminate the agreement.
5. Save the terminated agreement.
6. Reset the terminate indicator.
7. Refresh the original agreement.
8. Post-termination updates.

Merge Web Service

Converting a Word Document into a given output format for Standard Object (Deprecated)The Merge Web Service APIs are used for generating and manipulating documents. Both agreements and supporting documents can be generated, while text can be added to document headers and footers. Word documents can be converted into PDF documents and security rules can be applied to PDF documents.

The APIs enable you to do the following:

Generating a Document

Generates a Word agreement document by using the template specified by the *templateId* parameter and the agreement indicated by the *agreementId*. The document is added to the Files and Attachments section for the Agreement object.

Generating a Document for a Standard Object

Generates a Word agreement document for a standard object, using the template selected by the *templateId* parameter and the object indicated by the *sObjectId* and *sObjectType*.

Generating a Supporting Document

Generates a supporting document, such as Contract Summary, Renewal Letter, and Termination Letter, using the template specified by the *templateId* and is associated with the agreement record represented by the *agreementId*.

Adding Text to a Header and Footer

Appends the given text to the header and footer for a given document.

Converting a Word Document into a given output format

This API converts an agreement Word document, specified in *docFileName* into a specific output format using the parameters, such as *agreementId* and *docFormat*. This API internally checks if the header and footer are added to the generated document.

Converting a Word Document into a given output format for Standard Object (Deprecated)

For any standard object, this API converts a generated Word document into a specific output format, specified in *docFormat* using the parameters, such as, *docId*, *docFileName*. This API is deprecated.

Converting a Word Document into a given output format (Deprecated)

This API converts an agreement Word document into a specific output format using the parameters, such as *agreementId* and *docFormat*. This API is deprecated.

Converting a Word Document into a PDF

This converts a Word document into a PDF document. It can be any Word document and does not have to be created using *generateDoc* or *generateSupportingDoc*. The protection level applicable on the Word document is not applicable to the PDF document.

Converting a Word Document into a PDF for Standard Objects

This converts a Word document into a PDF document for a standard object. It can be any Word document and does not have to be created using *generateDoc* or *generateSupportingDoc*. The protection level applicable on the Word document is not applicable to the PDF document.

Applying Security Parameters to a PDF

This applies Adobe PDF security parameters to the PDF document. Ensure that you have set an admin entry with the PDFSecurityDefaulttags. For more information, refer *Contract Management Administrator Guide*.

Enabling Authoring for Agreement Documents

This API enables redlining capability for a list of documents, specified in the *docIds*, by using the parameters, such as *sObjectID* and *pLevel*.

Generating a Document on Submit Request

This API submits the generate document action using the given template from *templateId* and agreement from *agreementId*. The output format of the generated document is of the type specified in the *docFormat* parameter.

Generating a Document on Submit Request for a Standard Object

For a standard object, this API submits the generate document action using the given template from *templateId* and for a sObject from *sObjectId* and *sObjectType*. The output format of the generated document is of the type specified in the *docFormat* parameter.

Generating a Supporting Document on Submit Request

This API submits the generate supporting document action using the given template from *templateId* and for an agreement from *agreementId*. The output format of the generated supporting document is of the type specified in the *docFormat* parameter.

Packaging Multiple Documents

This API packages multiple documents (such as a Quote and Agreement document) together into a single merged DOCX document. Implementation requires creating a custom Visualforce page to use the API for document packaging purposes.

Merging Multiple Documents to a PDF

Apttus provides three APIs to assist you in merging multiple documents into a single PDF. Choose the API that best suits your needs.

Merging Proposal Documents into a single PDF Document

This API packages multiple proposal documents together into a single merged PDF document. Implementation requires creating a custom Visualforce page to use the API for document packaging purposes.

Adding Text to a Header and Footer

Append the given text to the header and footer for a given document.

appendToHeaderFooter

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the document which will have header and footer content added to it.	Yes

Parameters			
Name	Type	Description	Required?
headerText	String	The header text to be appended to the document.	Yes
footerText	String	The footer text to be appended to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server Url.	Yes

Response		
Name	Type	Description
result	Boolean	Indicates whether the header and footer are added successfully.

Code Sample

```

ID docId= AttachmentId;
String headerText = 'This is Header';
String footerText = 'This is Footer';
String apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() + '/services/Soap/u/14.0/' +
UserInfo.getOrganizationId();
String SessionId = sessionId; //This is the session Id.
String generatedDocId = Apttus_Proposal.MergeWebService.appendToHeaderFooter(docId,
headerText, footerText, sessionId, serverUrl);
    
```

Note `appendToHeaderFooter` does not support protection levels for FX1 documents (FX2 documents do support document protection). Document protection levels are reset when header and footer are added. To retain protection levels for FX1 documents, use `appendToHeaderFooter2` instead.

appendToHeaderFooter2

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the document which will have header and/or footer content add to it.	Yes
headerText	String	The header text to append to the document.	Yes
footerText	String	The footer text to append to the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server Url.	Yes

Response

Name	Type	Description
result	Boolean	Indicates whether the header and footer were added successfully.

Code Sample

```

ID docId = 'a0Y61000002K6IR';
String headerText = 'Agreement Id: 0123456789';
String footerText = 'Date Created: 11/27/2018 7:23:15 PM';
String pLevel = 'Full access';
String sessionId = sessionId; //This is the session Id.
String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/services/Soap/u/14.0/' + UserInfo.getOrganizationId();
Boolean result = Apttus.MergeWebService.appendToHeaderFooter2(docId, headerText, footerText, pLevel, sessionId, serverUrl);

```

Applying Security Parameters to a PDF

This applies Adobe PDF security parameters to the PDF document. Ensure that you have set an admin entry with the **PDFSecurityDefault** tags. For more information, refer *Contract Management Admin Guide*.

The `applyPDFSecurity()` function provides a great way to handle document security. Using this function is fairly complex, and what we want is a simple way to quickly apply security. The function provides us with just such an option. The actual parameters in the policy could be `docId`, `sessionId`, `serverUrl` and so on. This API controls the security properties of a PDF document, such as, Content Copying, Filling of form fields, and Printing.

applyPDFSecurity

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the PDF document to apply security parameters to.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server Url.	Yes

Response		
Name	Type	Description
result	Boolean	Indicates if the security parameters were applied successfully to the PDF.

Code Sample

```

1  ID docId = 'a0Y61000002K6IR';
2  String sessionId = sessionId; //This is the session Id.
3  String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
4  Boolean result =
  Apttus.MergeWebService.applyPDFSecurity(docId,sessionId,serverUrl);

```

Converting a Word Document into a given output format

This API converts an agreement Word document, specified in docFileName into a specific output format using the parameters, such as agreementId and docFormat. This API internally checks if the header and footer are added to the generated document.

Note

- If the output document format is DOC or DOCX, the Id returned is that of the updated input document.
- If the output document format is PDF, the Id returned is that of the newly created document.

convertWordDocForAgreement

Parameters			
Name	Type	Description	Required?
agreementId	ID	The id of the agreement providing the content of the document.	Yes
docId	ID	The id of the agreement document to convert in a different output format.	Yes
docFileName	String	The file name for the Word document.	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>RTF</i> • <i>PDF</i> 	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
addWatermark	Boolean	Indicates whether a draft watermark should be added to the output document.	Yes
removeWatermark	Boolean	Indicates whether a draft watermark on the source word document should be removed from the resulting document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server URL.	Yes

Response		
Name	Type	Description
docId	ID	The id of the converted document.

Code Sample

```
String apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() + '/services/Soap/u/14.0/' + UserInfo.getOrganizationId();
ID attachmentID = 'a036F00000urlTX';
ID agreementId = 'a016F00001V5HZA';
String pLevel = 'Read only';
String docFormat = 'PDF';
Boolean addWatermark = true;
Boolean removeWatermark = true;
ID docId = Apttus.MergeWebService.convertWordDocForAgreement(agreementId, attachmentID, null, docFormat, pLevel, addWatermark, removeWatermark, ServerInfo.getSessionId(), apiServerURL);
```

Converting a Word Document into a given output format (Deprecated)

This API converts an agreement Word document into a specific output format using the parameters, such as *agreementId* and *docFormat*. This API is deprecated.

convertAgreementWordDoc

Parameters			
Name	Type	Description	Required?
agreementId	ID	The id of the agreement providing the content for the document.	Yes
docId	ID	The id of the agreement document to convert in a different output format.	Yes
docFileName	String	The file name for the Word document.	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>RTF</i> 	Yes

Parameters			
Name	Type	Description	Required?
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
addWatermark	Boolean	Indicates whether a draft watermark should be added to the output document.	Yes
removeWatermark	Boolean	Indicates whether a draft watermark on the source word document should be removed from the resulting document.	Yes
agreementNbr	String	The agreement number of the document, upserted in the header.	Yes
dateFormat	String	The date format for the footer of the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server URL.	Yes

Response		
Name	Type	Description
result	ID	The id of the converted document.

Code Sample

```
1 ID agreementId = 'a016100000Che3D';
2 ID docId = 'a0Y61000002K6IR';
3 String docFileName = 'Agreement';
4 String docFormat = 'PDF';
5 String pLevel = 'Full access';
6 Boolean addWatermark = true;
7 Boolean removeWatermark = false;
8 String agreementNbr = '1234';
9 String dateFormat = 'M/d/yyyy HH:mm:ss';
10 String sessionId = sessionId; //This is the session Id.
11 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
services/Soap/u/14.0/' + UserInfo.getOrganizationId();
12 ID ConvertedDocId =
Apttus.MergeWebService.convertAgreementWordDoc(agreementId,docId,docFileName,
docFormat,pLevel,addWatermark,removeWatermark,agreementNbr,dateFormat,session
Id,serverUrl);
```

Converting a Word Document into a given output format for Standard Object (Deprecated)

For any standard object, this API converts a generated Word document into a specific output format, specified in *docFormat* using the parameters, such as, *docId*, *docFileName*. This API is deprecated.

convertWordDoc

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the agreement document to be converted in a different output format.	Yes
docFileName	String	The file name for the Word document.	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>RTF</i> 	Yes
sObjectId	ID	The sObject id for which you want to update the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
addWatermark	Boolean	Indicates whether a draft watermark should be added to the output document.	Yes
removeWatermark	Boolean	Indicates whether a draft watermark on the source word document should be removed from the resulting document.	Yes
headerText	String	The header text to be appended to the document.	Yes
footerText	String	The footer text to be appended to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server URL.	Yes

Response		
Name	Type	Description
result	ID	The id of the converted document.

Code Sample

```

1  ID docId = 'a0Y61000002K6IR';
2  String docFileName = 'Agreement';
3  String docFormat = 'PDF';
4  ID sObjectId = '0016100000KvBHi';
5  String pLevel = 'Full access';
6  Boolean addWatermark = true;
7  Boolean removeWatermark = false;
8  String headerText = 'Agreement Id: 0123456789';
9  String footerText = 'Date Created: 11/27/2018 7:23:15 PM';
10 String sessionId = sessionId; //This is the session Id.
11 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
    services/Soap/u/14.0/' + UserInfo.getOrganizationId();
12
13 ID ConvertedDocId=
    Apttus.MergeSupport.convertWordDoc(docId,docFileName,docFormat,sObjectId,pLevel,
    addWatermark,removeWatermark,headerText,footerText,sessionId,serverUrl) ;

```


Converting a Word Document into a PDF

This converts a Word document into a PDF document. It can be any Word document and does not have to be created using generateDoc or generateSupportingDoc. The protection level applicable to the Word document is not applicable to the PDF document.

convertWordDocToPDF

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the word document to be converted to PDF.	Yes
docFileName	String	The filename for the newly created PDF.	Yes
addWatermark	Boolean	Indicates whether a draft watermark should be added to the PDF.	Yes
removeWatermark	Boolean	Indicates whether a draft watermark on the source word document should be removed from the resulting PDF.	Yes
headerText	String	The header text to append to the document.	Yes
footerText	String	The footer text to append to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server URL.	Yes

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
Id docId = attachmentId;
String filename = 'newPDFFile.pdf';
Boolean addWatermark = 'true';
Boolean removeWatermark = 'false';
String headerText = 'This is Header';
String footerText = 'This is Footer';
String apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() + '/services/
Soap/u/14.0/' + UserInfo.getOrganizationId();
String sessionId = sessionId; //This is the session Id.
Id convertedDocId = Apttus.MergeWebService.convertWordDocToPDF(docId, fileName,
addWatermark, removeWatermark, headerText, footerText, apiSessionId, apiServerURL);
```

Converting a Word Document into a PDF for Standard Objects

This converts a Word document into a PDF document for a standard object. It can be any Word document and does not have to be created using *generateDoc* or *generateSupportingDoc*. The protection level applicable on the Word document is not applicable to the PDF document.

convertWordDocToPDF2

Parameters			
Name	Type	Description	Required?
docId	ID	The id of the word document to be converted to PDF.	Yes
docFileName	String	The filename for the newly created PDF.	Yes
sObjectId	Id	The id of the standard object.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
addWatermark	Boolean	Indicates whether a draft watermark should be added to the PDF.	Yes
removeWatermark	Boolean	Indicates whether a draft watermark on the source word document should be removed from the resulting PDF.	Yes
headerText	String	The header text to append to the document.	Yes
footerText	String	The footer text to append to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server URL.	Yes

Response

Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
1 ID docId = 'a0Y61000002K6IR';
2 String docFileName = 'Agreement';
3 ID sObjectId = null;
4 String pLevel = null;
5 Boolean addWatermark = true;
6 Boolean removeWatermark = false;
7 String headerText = 'Agreement Id: 0123456789';
8 String footerText = 'Date Created: 11/27/2018 7:23:15 PM';
9 String sessionId = sessionId; //This is the session Id.
10 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
    services/Soap/u/14.0/' + UserInfo.getOrganizationId();
11
12 ID ConvertedDocId =
    Apttus.MergeWebService.convertWordDocToPDF2(docId,docFileName,sObjectId,pLevel,
    addWatermark,removeWatermark,headerText,footerText,sessionId,serverUrl);
```

Enabling Authoring for Agreement Documents

This API enables redlining capability for a list of documents, specified in the *docIds*, by using the parameters, such as *sObjectID* and *pLevel*.

enableAuthoring

Parameters			
Name	Type	Description	Required?
sObjectID	ID	The ID of the standard object for which authoring is to be enabled.	Yes
pLevel	String	The protection level to apply to the document. Valid values are <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
docIds	List	The ids of the documents for which you want to enable authoring. If the value is null, all the documents associated with the sObject are enabled. You can list comma separated docIds.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
result	Boolean	Indicates if the authoring was enabled successfully for the listed documents.

Code Sample

```
1 ID sObjectId = 'a016100000Che3D';
2 String pLevel = 'Full access';
3 ID[] docIds = new ID[]
4 {'a0Y61000002K6IR','a0Y61000003K7jR'};
5 String sessionId = sessionId; //This is the session Id.
6 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
services/Soap/u/14.0/' + UserInfo.getOrganizationId();
7
8 Boolean success =
Apttus.MergeWebService.enableAuthoring(sObjectId,pLevel,docIds,sessionId,serverUrl);
```

Generating a Document

Generates a Word agreement document by using the template specified by the `templateId` parameter and the agreement indicated by the `agreementId`. The document is added to the Files and Attachments section for the Agreement object.

The document can subsequently be converted to a PDF, using [convertWordDocToPDF](#).

generateDoc

Parameters			
Name	Type	Description	Required?
<code>templateId</code>	ID	The id of the template used to generate the word document.	Yes
<code>agreementId</code>	ID	The id of the agreement providing the content for the document.	Yes
<code>pLevel</code>	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
<code>docFormat</code>	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
<code>sessionId</code>	String	The Salesforce session id.	Yes
<code>serverUrl</code>	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
<code>generatedDocId</code>	ID	The id of the generated document.

Code Sample

```

1 ID templateId= 'a036F00000urlTX';
2 Id agreementId = 'a016F00001V5HZA';
3 String pLevel = 'Read only';
4 String docFormat = 'DOC';
5 String apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() + '/'
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
6 String sessionId = UserInfo.getSessionId(); //This is the session Id.
7 String generatedDocId = Apttus.MergeWebService.generateDoc(templateId,
  agreementId, pLevel, docFormat, sessionId, apiServerURL);

```

generateDoc2

This API is the same as the **generateDoc** API and additionally enables you to add a draft watermark to the generated document.

Parameters			
Name	Type	Description	Required?
templateId	ID	The id of the template used to generate the word doc.	Yes
agreementId	ID	The id of the agreement providing the content for the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
isDraft	Boolean	Indicates whether a draft watermark should be added to the document.	Yes
sessionId	String	The Salesforce session id.	Yes

Parameters			
Name	Type	Description	Required?
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
generatedDocId	ID	The id of the generated document.

Code Sample

```

1 ID templateId= 'a036F00000urlTX';
2 Id agreementId = 'a016F00001V5K6C';
3 String pLevel = 'Read only';
4 String docFormat = 'DOC';
5 Boolean isDraft = true;
6 String apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
7 String sessionId = UserInfo.getSessionId(); //This is the session Id.
8 String generatedDocId = Apttus.MergeWebService.generateDoc2(templateId,
  agreementId, pLevel, docFormat, isDraft, sessionId, apiServerURL);

```

Generating a Document for a Standard Object

Generates a Word agreement document for a standard object, using the template selected by the `templateId` parameter and the object indicated by `sObjectId` and `sObjectType`.

`generateCustomDoc`

Parameters			
Name	Type	Description	Required?
<code>templateId</code>	ID	The id of the template used to generate the word doc.	Yes
<code>sObjectId</code>	ID	The id of the sObject holding the data for the merge fields.	Yes
<code>sObjectType</code>	String	The type of the sObject.	Yes
<code>pLevel</code>	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
<code>docFormat</code>	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
<code>isDraft</code>	Boolean	Indicates whether a draft watermark should be added to the document.	Yes
<code>sessionId</code>	String	The Salesforce session id.	Yes
<code>serverUrl</code>	String	The Salesforce server id.	Yes
Response			
Name	Type	Description	
<code>docId</code>	ID	The id of the generated document.	

Code Sample

```

ID templateId = 'a036100000Fha5D';
ID sObjectId = '0016100000KvBHi';
String sObjectType = 'Account';
String pLevel = 'Full access';
String docFormat = 'DOCX';
Boolean isDraft = true;
String sessionId = sessionId; //This is the session Id.
String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/services/Soap/
u/14.0/' + UserInfo.getOrganizationId();
ID docId =
Apttus.MergeWebService.generateCustomDoc(templateId,sObjectId,sObjectType,pLevel,docForm
at,isDraft,sessionId,serverUrl);

```

Generating a Document on Submit Request

This API submits the generate document action using the given template from *templateId* and agreement from *agreementId*. The output format of the generated document is of the type specified in the *docFormat* parameter.

submitGenerateDoc

Parameters			
Name	Type	Description	Required?
templateId	ID	The id of the template used to generate the word doc.	Yes
agreementId	ID	The id of the agreement providing the content for the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
isDraft	Boolean	Indicates whether a draft watermark should be added to the document.	Yes

Parameters			
Name	Type	Description	Required?
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
1 ID templateId = 'a036100000Fha5D';
2 ID agreementId = 'a016100000Che3D';
3 String pLevel = 'Full access';
4 String docFormat = 'DOC';
5 Boolean isDraft = false;
6 String sessionId = sessionId; //This is the session Id.
7 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
8
9 ID callId =
  Apttus.MergeWebService.submitGenerateDoc(templateId,agreementId,pLevel,docFor
  mat,isDraft,sessionId,serverUrl);
```

Generating a Document on Submit Request for a Standard Object

For a standard object, this API submits the generate document action using the given template from *templateId* and for a sObject from *sObjectId* and *sObjectType*. The output format of the generated document is of the type specified in the *docFormat* parameter.

submitGenerateCustomDoc

Parameters			
Name	Type	Description	Required?
templateId	ID	The id of the template used to generate the word doc.	Yes
sObjectId	ID	The sObject id which comprises the data for the merge fields.	Yes
sObjectType	String	The type of the sObject.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
isDraft	Boolean	Indicates whether a draft watermark should be added to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```

1  ID templateId = 'a036100000Fha5D'
2  ID sObjectId = '0016100000KvBHi';
3  String sObjectType = 'Account';
4  String pLevel = 'Full access';
5  String docFormat = 'DOC';
6  Boolean isDraft= false;
7  String sessionId = sessionId; //This is the session Id.
8  String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/'
   services/Soap/u/14.0/' + UserInfo.getOrganizationId();
9
10 ID callId =
   Apttus.MergeWebService.submitGenerateCustomDoc(templateId,sObjectId,sObjectTy
   pe,pLevel,docFormat,isDraft,sessionId,serverUrl);

```

Generating a Supporting Document

Generates a supporting document, such as Contract Summary, Renewal Letter, and Termination Letter, using the template specified by the *templateId* and is associated with the agreement record represented by the *agreementId*.

generateSupportingDoc

Parameters			
Name	Type	Description	Required?
templateId	ID	The id of the template used to generate the word document.	Yes
agreementId	ID	The id of the agreement providing the content for the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i> 	Yes
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i> 	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
1 ID templateId = 'a036100000Fha5D';
2 ID agreementId = 'a016100000Che3D';
3 String pLevel = 'Full access';
4 String docFormat = 'DOC';
5 String sessionId = sessionId; //This is the session Id.
6 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
7 ID docId =
  Apttus.MergeWebService.generateSupportingDoc(templateId,agreementId,pLevel,do
  cFormat,sessionId,serverUrl);
```

generateSupportingDoc2

This API is the same as the **generateSupportingDoc** API and additionally enables you to add a draft watermark to the generated document.

Parameters			
Name	Type	Description	Required?
templateId	ID	The id of the template used to generate the word doc.	Yes
agreementId	ID	The id of the agreement providing the content for the document.	Yes
pLevel	String	The protection level to apply to the document. Valid values are: 'Full access', 'Insert comments only', 'Insert comments and tracked changes only', 'Read only', and 'Fill in form fields only'.	Yes
docFormat	String	Indicates whether the document will be 'DOC', 'DOCX', 'PDF', or 'RTF'.	Yes
isDraft	Boolean	Indicates whether a draft watermark should be added to the document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
1 ID templateId = 'a036100000Fha5D';
2 ID agreementId = 'a016100000Che3D';
3 String pLevel = 'Full access';
4 String docFormat = 'DOC';
5 isDraft= true;
6 String sessionId = sessionId; //This is the session Id.
7 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
8 ID docId =
  Apttus.MergeWebService.generateSupportingDoc2(templateId,agreementId,pLevel,d
  ocFormat,isDraft,sessionId,serverUrl);
```

Generating a Supporting Document on Submit Request

This API submits the generate supporting document action using the given template from *templateId* and for an agreement from *agreementId*. The output format of the generated supporting document is of the type specified in the *docFormat* parameter.

submitGenerateSupportingDoc

Parameters		
Name	Type	Description
templateId	ID	The id of the template used to generate the word document.
agreementId	ID	The id of the agreement providing the content for the document.
pLevel	String	The protection level to apply to the document. Valid values are: <ul style="list-style-type: none"> • <i>Full access</i> • <i>Insert comments only</i> • <i>Insert comments and tracked changes only</i> • <i>Read only</i> • <i>Fill in form fields only</i>
docFormat	String	Indicates whether the document will be: <ul style="list-style-type: none"> • <i>DOC</i> • <i>DOCX</i> • <i>PDF</i> • <i>RTF</i>
isDraft	Boolean	Indicates whether a draft watermark should be added to the document.
sessionId	String	The Salesforce session id.
serverUrl	String	The Salesforce server id.

Response		
Name	Type	Description
result	ID	The id of the generated document.

Code Sample

```
1 ID templateId = 'a036100000Fha5D';
2 ID agreementId = '0016100000KvBHi';
3 String pLevel = 'Full access';
4 String docFormat = 'DOC';
5 Boolean isDraft= false;
6 String sessionId = sessionId; //This is the session Id.
7 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
8
9 ID callId =
  Apttus.MergeWebService.submitGenerateSupportingDoc(templateId,agreementId,pLe
  vel,docFormat,isDraft,sessionId,serverUrl);
```

Merging Multiple Documents to a PDF

You can use any of the three following Merge Webservice APIs to merge multiple documents into a single PDF. Choose the API that best suits your needs (note limitations for each):

- mergeDocsToPDF
- mergeDocsToPDF2
- createPDF

mergeDocsToPDF

This API merges multiple documents using the values specified in the docIds parameter into a single PDF document with the name docFileName.

Parameters			
Name	Type	Description	Required?
parentId	ID	The id of the parent object to associate with the result document.	Yes
docIds	List	A collection of the ids of the documents to merge into a PDF document.	Yes
docFileName	String	The file name for the PDF document.	Yes
sessionId	String	The Salesforce session id.	Yes
serverUrl	String	The Salesforce server id.	Yes

Note

You cannot use Apttus-generated PDFs with document protection enabled as input documents.

Response

Name	Type	Description
result	ID	The id of the generated document.

Note

Footers in the final PDF will always list the original filename of the input document for the collated page.

Code Sample

```

1 ID parentId = '0016100000KvBHi'
2 ID[] docIds = new ID[]{'a0Y61000002K6IR','a0Y61000003K7jR'};
3 String docFileName = 'Agreement-Merged.pdf';
4 String sessionId = sessionId; //This is the session Id.
5 String serverUrl = System.Url.getSalesforceBaseUrl().toExternalForm() + '/
  services/Soap/u/14.0/' + UserInfo.getOrganizationId();
6
7 ID docId =
  Apttus.MergeWebService.mergeDocsToPDF(parentId,docIds,docFileName,sessionId,s
  erverUrl);

```

mergeDocsToPDF2

The API introduced in this release is an enhancement of the existing Merge Webservice API `mergeDocsToPDF`. `mergeDocsToPDF2` merges multiple documents using the values specified in the `docIds` parameter into a single PDF document with the name `docFileName` and adds a flag to specify whether PDF document protection should be carried over to the collated PDF.

Before invoking this API, **CustomClass.ActionParams** must be instantiated by passing the parameters described below. The parameters are then passed to the API when it is invoked.

Parameters			
Name	Type	Description	Required?
parentId	Id	The Id of the parent object to associate with the result (merged) document.	Yes
docIds	List	A collection of the Ids of the documents to merge into a PDF document.	Yes
docFileName	String	The file name for the PDF document.	Yes
sessionId	String	The Salesforce session Id.	Yes
sessionUrl	String	The Salesforce server Id.	Yes
ProtectOutputDoc	Boolean	If true, protect the output document with PDF security as defined in Comply properties.	Yes

Parameters																																																	
Name	Type	Description			Required?																																												
addAutoHeaderFooter	Boolean	If true: <ul style="list-style-type: none"> The agreement number is added as the header on the generated agreement. The timestamp as defined in comply system property is added as the footer on the generated agreement. If false, you need to define the following parameters.			Yes																																												
<table border="1"> <thead> <tr> <th colspan="5">Parameters</th> </tr> <tr> <th>S. No</th> <th>Name</th> <th>Type</th> <th>Description</th> <th>Required?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>AddHeader</td> <td>Boolean</td> <td>If true, you need to define parameters from 2-6.</td> <td>Optional</td> </tr> <tr> <td>2</td> <td>Header Mode</td> <td>String</td> <td>The existing footer values are customized based on the values provided for parameters 3-6. Supported value: Overwrite</td> <td>Yes if AddHeader is true</td> </tr> <tr> <td>3</td> <td>Header Text</td> <td>String</td> <td>The text that should be displayed at the header of the generated document.</td> <td>Yes if AddHeader is true</td> </tr> <tr> <td>4</td> <td>Header Font</td> <td>String</td> <td>The system font for the header text. Sample value: Arial, Tahoma</td> <td>Yes if AddHeader is true</td> </tr> <tr> <td>5</td> <td>Header FontSize</td> <td>Integer</td> <td>The font size of the header text.</td> <td>Yes if AddHeader is true</td> </tr> <tr> <td>6</td> <td>Header Alignment</td> <td>String</td> <td>The alignment of the header text. Supported values: Left, Right, Center</td> <td>Yes if AddHeader is true</td> </tr> <tr> <td>7</td> <td>AddFooter</td> <td>Boolean</td> <td>If true, you need to define parameters from 8-12.</td> <td>Optional</td> </tr> </tbody> </table>					Parameters					S. No	Name	Type	Description	Required?	1	AddHeader	Boolean	If true, you need to define parameters from 2-6.	Optional	2	Header Mode	String	The existing footer values are customized based on the values provided for parameters 3-6. Supported value: Overwrite	Yes if AddHeader is true	3	Header Text	String	The text that should be displayed at the header of the generated document.	Yes if AddHeader is true	4	Header Font	String	The system font for the header text. Sample value: Arial, Tahoma	Yes if AddHeader is true	5	Header FontSize	Integer	The font size of the header text.	Yes if AddHeader is true	6	Header Alignment	String	The alignment of the header text. Supported values: Left, Right, Center	Yes if AddHeader is true	7	AddFooter	Boolean	If true, you need to define parameters from 8-12.	Optional
Parameters																																																	
S. No	Name	Type	Description	Required?																																													
1	AddHeader	Boolean	If true, you need to define parameters from 2-6.	Optional																																													
2	Header Mode	String	The existing footer values are customized based on the values provided for parameters 3-6. Supported value: Overwrite	Yes if AddHeader is true																																													
3	Header Text	String	The text that should be displayed at the header of the generated document.	Yes if AddHeader is true																																													
4	Header Font	String	The system font for the header text. Sample value: Arial, Tahoma	Yes if AddHeader is true																																													
5	Header FontSize	Integer	The font size of the header text.	Yes if AddHeader is true																																													
6	Header Alignment	String	The alignment of the header text. Supported values: Left, Right, Center	Yes if AddHeader is true																																													
7	AddFooter	Boolean	If true, you need to define parameters from 8-12.	Optional																																													

Parameters					
Name	Type	Description			Required?
Parameters					
S. No	Name	Type	Description	Required?	
8	Footer Mode	String	The existing footer values are customized based on the values provided for parameters 9-12. Supported value: Overwrite	Yes if AddHeader is true	
9	FooterText	String	The text that should be displayed at the footer of the generated document.	Yes if AddHeader is true	
10	FooterFont	String	The system font for the footer text. Sample value: Arial, Tahoma	Yes if AddHeader is true	
11	FooterFontSize	Integer	The font size of the footer text.	Yes if AddHeader is true	
12	FooterAlignment	String	The alignment of the footer text. Supported values: Left, Right, Center	Yes if AddHeader is true	

Response		
Name	Type	Description
docId	ID	The id of the generated PDF document.

Code Sample

```

1  public String docFormat { get; set; }
2  public String outputDocFileBasename { get; set; }
3  private Boolean protectOutputDocStatus;
4  public String protectOutputDocFlag { get; set; }
5  private Boolean preserveRedlinesStatus;
6  public String preserveRedlinesFlag { get; set; }
7  public List<SelectOption> listOfDocxAttachmentsRelated2Agreement_Selected
   { get; set; }
8  public List<SelectOption> listOfPDFAttachmentsRelated2Agreement_Selected
   { get; set; }
9  public String apiServerURL = '';
10 private String servicesSOAPInfo = '/services/Soap/u/36.0/';
11
12 ...
13 ...
14
15 private void init() {
16
17     docFormat = 'PDF';
18     protectOutputDocFlag = 'True';
19     protectOutputDocStatus = true;
20     preserveRedlinesFlag = 'True';
21     preserveRedlinesStatus = true;
22     outputDocFileBasename = 'Agreement_outputDocFilename_mergeDocsToPDF2';
23     sessionId = UserInfo.getSessionId();
24     apiServerURL = System.Url.getSalesforceBaseUrl().toExternalForm() +
   servicesSOAPInfo + UserInfo.getOrganizationId();
25 }
26
27
28 public void mergeDocsToPDF2_ControllerAction() {
29
30     String outputDocFilename = outputDocFileBasename + '.' + docFormat;
31
32     List<Apttus.CustomClass.AttachmentInfo> attachmentsInfoList = new
   List<Apttus.CustomClass.AttachmentInfo>();
33     List<Id> attachmentsIdList = new List<Id>();
34
35     // Attachments: Docx
36     if (listOfDocxAttachmentsRelated2Agreement_Selected != null) {
37
38         if (!listOfDocxAttachmentsRelated2Agreement_Selected.isEmpty()) {
39
40             for (selectoption so : listOfDocxAttachmentsRelated2Agreement_Selected) {
41
42                 Apttus.CustomClass.AttachmentInfo attachmentInfo = new
   Apttus.CustomClass.AttachmentInfo();
43
44                 Attachment attachmentDetails = getAttachmentDetails(so.getValue());

```

```

45 attachmentInfo.Title = attachmentDetails.Name;
46 attachmentInfo.RecordId = attachmentDetails.Id;
47 attachmentsInfoList.add(attachmentInfo);
48 }
49 }
50 }
51
52 // Attachments: PDF
53 if (listOfPDFAttachmentsRelated2Agreement_Selected != null) {
54
55     if (!listOfPDFAttachmentsRelated2Agreement_Selected.isEmpty()) {
56
57         for (selectoption so : listOfPDFAttachmentsRelated2Agreement_Selected) {
58
59             Apttus.CustomClass.AttachmentInfo attachmentInfo = new
                Apttus.CustomClass.AttachmentInfo();
60
61             Attachment attachmentDetails = getAttachmentDetails(so.getValue());
62             attachmentInfo.Title = attachmentDetails.Name;
63             attachmentInfo.RecordId = attachmentDetails.Id;
64             attachmentsInfoList.add(attachmentInfo);
65         }
66     }
67 }
68
69 //
70 // Attachments
71 // DocIds is all that is needed by this API - mergeDocsToPDF2
72 //
73 for (Apttus.CustomClass.AttachmentInfo info : attachmentsInfoList) {
74
75     attachmentsIdList.add(info.RecordId);
76 }
77
78 if (getAgreement!= null &&
79     !string.isBlank(outputDocFilename) &&
80     (docFormat.equalsIgnoreCase('PDF')) &&
81     (attachmentsInfoList.size() != 0) &&
82     apiServerURL != null && apiServerURL != '' &&
83     sessionId != null && sessionId != '') {
84
85     //Agreement__c is self lookup field on Agreement
86     System.debug(apiName + ': getAgreement.Agreement__c: ' +
        getAgreement.Agreement__c);
87     Apttus__APTS_Agreement__c myAgreement = [SELECT Id, Name FROM
        Apttus__APTS_Agreement__c WHERE Id = :getAgreement.Agreement__c];
88
89     // API Parameters
90     Apttus.CustomClass.ActionParams apiParameters = new
        Apttus.CustomClass.ActionParams();
91
92     apiParameters.ParentId = myAgreement.Id;

```

```

93  apiParameters.DocIds = attachmentsIdList;
94  apiParameters.DocFileName = outputDocFilename;
95  apiParameters.PreserveRedlines = getBooleanStatus(preserveRedlinesFlag);
96  apiParameters.ProtectOutputDoc = getBooleanStatus(protectOutputDocFlag);
97  apiParameters.SessionId = sessionId;
98  apiParameters.SessionUrl = apiServerURL;
99
100 apiParameters.addAutoHeaderFooter = false;
101 apiParameters.AddHeader = true;
102 apiParameters.HeaderMode = 'Overwrite';
103 apiParameters.HeaderText= 'Contract Management';
104 apiParameters.HeaderFont = 'Arial';
105 apiParameters.HeaderFontSize = '12';
106 apiParameters.HeaderAlignment = 'Left';
107 apiParameters.AddFooter = false;
108 apiParameters.FooterMode = 'Overwrite';
109 apiParameters.FooterText= 'Apttus';
110 apiParameters.FooterFont = 'Arial';
111 apiParameters.FooterFontSize = '12';
112 apiParameters.FooterAlignment = 'Left';
113
114
115 System.debug(apiParameters);
116
117 try {
118
119 //submit request
120 Id myDocId = Apttus.MergeWebService.mergeDocsToPDF2(apiParameters);
121
122 if (myDocId != null) {
123
124 // Get details of the corresponding word document.
125 String pdfFilename = '%' + outputDocFileBasename + '.pdf';
126 System.debug(' ' + apiName + ': Querying for: ' + pdfFilename);
127 List<Attachment> myDocIds = [SELECT Id, Name, CreatedDate FROM Attachment
WHERE ParentID = :myAgreement.Id AND Name like :pdfFilename ORDER BY
CreatedDate DESC];
128 Attachment myAttachment = myDocIds.get(0);
129 getGeneratedDocumentDetails(myAttachment.Id, 'PDF');
130
131 } else {
132
133 ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'API
mergeDocsToPDF2() returned null. Please try again.));
134 }
135
136 } catch (Exception ex) {
137
138 System.debug('Apttus.MergeWebService.mergeDocsToPDF2: Oops!!! Caught an
Exception 1: ' + ex.getMessage());

```

```

139     }
140
141     } else {
142     ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR,
143     'Argument(s) was not set. Please try again'));
144     }
145
146
147     private Boolean getBooleanStatus(String flagValue) {
148
149     If ( flagValue.equalsIgnoreCase('True')) {
150     return true;
151     } else {
152     return false;
153     }
154     }

```

Note

ProtectOutputDoc only enforces document protection in the target PDF when Enable PDF Security was properly configured at the time of PDF generated in the Org corresponding to the input document. Refer to Enabling PDF Security for Agreement Documents in Contract Management Administrator Guide for more information.
All redlines that are present in input documents are accepted prior to collation.

createPDF

This API merges multiple documents using the values specified in the docIds parameter into a single PDF document with the name docFileName. This API enhances mergeDocsToPDF by providing the ability to automatically display or hide the footer in the merged PDF.

Before invoking this API, CustomClass.ActionParams must be instantiated by passing the parameters described below. The parameters are then passed to the API when it is invoked.

Parameters		
Name	Type	Description
parentId	Id	The Id of the parent object to associate with the result (merged) document.
docIds	List	A collection of the Ids of the documents to merge into a PDF document.
docFileName	String	The file name for the PDF document.
sessionId	String	The Salesforce session Id.
sessionUrl	String	The Salesforce server Id.
addFooter	Boolean	Display/hide the footer in the merged PDF.



Note

Footers in the final PDF will always list the original filename of the input document for the collated page.
All redlines that are present in input documents are accepted prior to collation.

Response

Name	Type	Description
result	Id	The Id of the generated (merged) PDF.

Code Sample

```

1 CustomClass.ActionParams objParam = new CustomClass.ActionParams();
2 objParam.SessionId = ServerInfo.getSessionId();
3 objParam.SessionUrl = ServerInfo.getServerUrl();
4 objParam.ParentId = '{parentID}';
5 objParam.DocIds = new ID[]{ID.valueOf('{docId}'),
6 ID.valueOf('{docId}')};
7 objParam.DocFileName = 'My_Merged_PDF.pdf';
8 objParam.AddFooter = true;
9 ID mergedPDF = MergeWebService.createPDF(objParam);
    
```



Note

Values inside '{ }' in the above code sample denote an Id is required.

Merging Proposal Documents into a single PDF Document

createPDFPackage

Use this API to package multiple proposal documents together into a singled merged PDF document. Implementation requires creation of a custom Visualforce page to make use of the API for document packaging purposes.

Parameters		
Name	Type	Description
parentId	ID	The ID of the parent object associated with the result.

Parameters		
Name	Type	Description
attachments	List<Apttus.CustomClass.AttachmentInfo>	List of all documents in the order to be packaged.
externalLinks	List<Apttus.CustomClass.ExternalLink>	List of external links to be embedded in document.
docFileName	String	Name of the packaged document file (PDF).
sessionId	ID	The Session ID to use for callback.
sessionURL	String	The Server URL to use for callback.

Response		
Name	Type	Description
result	ID	Returns the ID of the merged PDF document.

Code Sample

```

1      // Attachments
2          //
3          List<Apttus_Proposal.CustomClass.AttachmentInfo> attachmentsInfoList
= new List<Apttus_Proposal.CustomClass.AttachmentInfo>();
4          // Attachment: Primary
5          Apttus_Proposal.CustomClass.AttachmentInfo attachmentInfo = new
Apttus_Proposal.CustomClass.AttachmentInfo();
6          ...
7          ...
8          attachmentInfo.Title          = attachmentDetails.Name;
9          attachmentInfo.RecordId       = attachmentDetails.Id;
10         attachmentsInfoList.add(attachmentInfo);
11
12         // Attachments: Docx
13         for (selectoption so :
listOfDocxAttachmentsRelated2Proposal_Selected) {
14
15             Apttus_Proposal.CustomClass.AttachmentInfo attachmentInfo = new
Apttus_Proposal.CustomClass.AttachmentInfo();
16             ...
17             ...
18             attachmentInfo.Title          = attachmentDetails.Name;
19             attachmentInfo.RecordId       = attachmentDetails.Id;
20             attachmentsInfoList.add(attachmentInfo);
21
22         }
23         // Attachments: PDFs
24         for (selectoption so : listOfPDFAttachmentsRelated2Proposal_Selected)
{
25
26             Apttus_Proposal.CustomClass.AttachmentInfo attachmentInfo = new
Apttus_Proposal.CustomClass.AttachmentInfo();
27             ...
28             ...
29             attachmentInfo.Title          = attachmentDetails.Name;
30             attachmentInfo.RecordId       = attachmentDetails.Id;
31             attachmentsInfoList.add(attachmentInfo);
32
33         }
34         //
35         // External Links
36         //
37         List<Apttus_Proposal.CustomClass.ExternalLink> externalLinksInfoList
= new List<Apttus_Proposal.CustomClass.ExternalLink>();
38         if ( !string.isBlank(externalLinks) ) {
39
40             String[] arrayOfURLs = parseURLsCSV(externalLinks);
41
42             for (String str : arrayOfURLs) {

```



```

43
44         String url      = str.trim();
45         String filename = getFilenameFromURL(url);
46
47         Apttus_Proposal.CustomClass.ExternalLink externalLinkInfo =
new Apttus_Proposal.CustomClass.ExternalLink();
48         externalLinkInfo.Title = filename;
49         externalLinkInfo.Url   = url;
50         externalLinksInfoList.add(externalLinkInfo);
51     }
52 }
53 // API Parameters
54 Apttus_Proposal.CustomClass.ActionParams apiParameters = new
Apttus_Proposal.CustomClass.ActionParams();
55 apiParameters.ParentId      = myProposal.Id;
56 apiParameters.Attachments  = attachmentsInfoList;
57 apiParameters.ExternalLinks = externalLinksInfoList;
58 apiParameters.DocFileName  = 'outputCollatedPDFFilename.pdf';
59 apiParameters.SessionId    = sessionId;
60 apiParameters.SessionUrl   = apiServerURL;
61 try {
62
63     //submit request
64     Id myDocId =
Apttus_Proposal.MergeWebService.createPDFPackage(apiParameters);
65
66 } catch (Exception ex) {
67
68     System.debug('Apttus.MergeWebService.createPDFPackage: Oops!!!
Caught an Exception: ' + ex);
69 }

```

Package Multiple Documents

This API can be used to package multiple documents (such as a Quote and Agreement document) together into a single merged DOCX document. Implementation requires creation of a custom Visualforce page to make use of the API for document packing purposes.

createWordPackage

Parameters			
Name	Type	Description	Required?
ParentId	ID	The ID of the parent object associated with the result.	Yes
Attachments	List<Apttus.CustomClass.AttachmentInfo>	List of all documents in the order to be packaged.	Yes
ExternalLinks	List<Apttus.CustomClass.ExternalLink>	List of external links to be embedded in document. (<i>not currently supported</i>)	Yes
DocFileName	String	Name of the packaged document file (.doc or .docx).	Yes
ProtectionLevel	String	Protection level of the document generated.	Yes
AddWatermark	Boolean	Indicates whether to add a watermark to the document.	Yes
RemoveWatermark	Boolean	Indicates whether to remove a watermark from the document.	Yes
AddAutoHeaderFooter	Boolean	Indicates whether to use the configured header and footer format located in Comply System Properties. <ul style="list-style-type: none"> • If AddAutoHeaderFooter=true, the API uses the Comply System Properties defined for header and footer data. • If AddAutoHeaderFooter=false, the API uses values from the <i>headerText</i> and <i>footerText</i> parameters. 	Yes
HeaderText	String	Text to include in the document header.	Yes
FooterText	String	Text to include in the document footer.	Yes
PreserveRedlines	Boolean	Indicates whether the redlines in the input document will be preserved/ accepted in the packaged document. The default value is false.	Yes
SessionId	ID	The Session ID to use for callback.	Yes
SessionUrl	String	The Server URL to use for callback.	Yes

i Note: header text will always be right-aligned and footer text will always be left-aligned by default. Also, any page numbers in the input documents will not be present in the final combined document.

Response		
Name	Type	Description
result	ID	Returns the ID of the merged DOCX document.

Code Sample

```
1 // attachments
2 List<Apttus.CustomClass.AttachmentInfo> attachments = new
  List<Apttus.CustomClass.AttachmentInfo>();
3 Apttus.CustomClass.AttachmentInfo attInfo = new
  Apttus.CustomClass.AttachmentInfo();
4 attInfo.Title = 'Agreement-1';
5 attInfo.RecordId = attS0.Id;
6
7 attachments.add(attInfo);
8
9 Apttus.CustomClass.AttachmentInfo attInfo2 = new
  Apttus.CustomClass.AttachmentInfo();
10 attInfo2.Title = 'Agreement-2';
11 attInfo2.RecordId = attS02.Id;
12
13 attachments.add(attInfo);
14
15 // external links (not supported)
16 List<Apttus.CustomClass.ExternalLink> links = new
  List<Apttus.CustomClass.ExternalLink>();
17
18 Apttus.CustomClass.ExternalLink linkInfo = new
  Apttus.CustomClass.ExternalLink();
19 linkInfo.Title = 'Drawing-1';
20 linkInfo.Url = 'https://drive.google.com/doc1.docx';
21
22 links.add(linkInfo);
23
24 Apttus.CustomClass.ExternalLink linkInfo2 = new
  Apttus.CustomClass.ExternalLink();
25 linkInfo2.Title = 'Drawing-2';
26 linkInfo2.Url = 'https://drive.google.com/doc2.docx';
27
28 links.add(linkInfo2);
29
30 // create action parameters
31 Apttus.CustomClass.ActionParams params = new
  Apttus.CustomClass.ActionParams();
32
33 params.ParentId = agreementS0.Id;
34 params.Attachments = attachments;
35 params.ExternalLinks = links;
36 params.AddAutoHeaderFooter = true;
37 params.ProtectionLevel = AgreementProtection.LEVEL_FULL_ACCESS;
38 params.DocFileName = 'Agreement-Merged.docx';
39 params.AddWatermark = true;
40 params.RemoveWatermark = false;
41 params.HeaderText = 'Agreement Id: 0123456789';
42 params.FooterText = 'Date Created: 11/27/2011 7:23:15 PM';
```

```

43 params.PreserveRedlines = true;
44 params.SessionId = ServerInfo.getSessionId();
45 params.SessionUrl = ServerInfo.getServerUrl();
46
47 ID docId = Apttus.MergeWebService.createWordPackage(params);
48
49 System.debug('Merged Doc ID=' + docId);

```

Author Web Service

The Author Web Service API contains all the services related to files or documents and can be used with various operations related to X-Author.

Getting Word Document Information for Parent with Files

This method returns the list of word documents of type files, based on parent id passed in parameter.

getWordDocumentInfosForParentWithFiles

Parameters		
Name	Type	Description
parentId	ID	The id of the parent agreement.

Response		
Name	Type	Description
List	ContentDocumentLink	Returns the list of content document link.

Code Sample

```

1 ID agreementId = 'a0Y61000002K6IR';
2 List<ContentDocumentLink> lstContentDocumentLink =
  Apttus.AuthorWebService.getWordDocumentInfosForParentWithFiles(agreementId);

```

Initiate Parallel Review

This initiates the parallel review for a document that is part of an Agreement. The document must be part of the agreement document version object and document versioning must be enabled.

initiateParallelReview

Parameters			
Name	Type	Description	Required?
attachmentId	ID	Id of the document that is to be reviewed.	Yes
agreementId	ID	Id of the agreement.	Yes
documentVersionDetailId	ID	Id of the document version.	Yes
emailSubject	String	Subject of the review email.	Yes
emailBody	String	Body of the review email.	Yes
List<ReviewerWrapper> reviewersInfo	List	List of reviewers.	Yes

Code Sample

```

Id attachmentId = '00P0S00000270MrUAI'; //attachment or fileId to sent for review
Id agreementId = 'a070S000002KpC5QAK'; //agreementId
Id documentVersionDetailId = 'a1y0S000000TsvxQAC'; // documentVersionDetailId -get the
parent Id of the attachment/file.
//set the reviewers list.
List<ReviewerWrapper> reviewersList = new List<ReviewerWrapper>();
//add the recipients -(multiple recipients can be added) can be user or contact or
email.
//'userOrContactId' is passed if it is user or contact, pass null if recipient type is
email.
//ensure recipientname and recipient email are passed.
ReviewerWrapper reviewerWrapper = new ReviewerWrapper(userOrContactId, 'recipientname',
'recipientemail@email.com', null);
reviewersList.add(reviewerWrapper);
//Invoke the method by passing above parameters along with Email Subject and Email Body.
AuthorWebService.initiateParallelReview(attachmentId, agreementId,
documentVersionDetailId, 'Email Subject', 'Email Body in HTML', reviewersList);
    
```

End Parallel Review

This ends the parallel review for a document belonging to a particular review cycle.

endParallelReview

Parameters			
Name	Type	Description	Required?
reviewCycleId	ID	Id of the review cycle.	Yes
emailId	String	Email Id of the reviewer.	Yes
documentVersionDetailId	ID	Id of the document version.	Yes
comment	String	Reviewer comment.	Optional

Code Sample

```
ID reviewCycleId = 'a2a4N000000X86gQAC';
String emailId = 'recipientemail@email.com';
ID documentVersionDetailId = 'a1y0S000000TsvxQAC' ;
String comment = 'Review done Please incorporate changes ' ;
emailId
documentVersionDetailId
documentVersionDetailId );
AuthorWebService.endParallelReview(reviewCycleId ,
```

Callback Classes

Apttus Contract Management allows you to create and register custom code in the form of callback class for the following operations:

- [IAgreementLifecycleCallback2](#) and [IDynamicSectionCallback2](#)

IAgreementLifecycleCallback2 and IDynamicSectionCallback2

Contract Management on salesforce provides the following callback interfaces:

IAgreementLifecycleCallback2

The administrator can use this callback class to handle the Agreement lifecycle specific actions like afterCreate , afterClone, afterTerminate, afterAmend, and so on.

IDynamicSectionCallback2

The administrator can use this callback class to handle the dynamic section when an agreement document is generated.

To enable the callback interface for IAgreementLifecycleCallback2

1. Write a class implementing the Interface.
2. Overwrite the methods and write what you want to add in a specific action method.
3. Go to **Setup > Custom Settings > Comply Custom Classes**.
4. Create a New Entry.
5. Execute actual Action and check the log.
6. Create Apex Class implementing the Callback Interface.

Callback Methods Invoked

S.No	Callback Method	When is the callback method invoked
01	afterActivate	After Agreement is Activated
02	afterAmend	After Agreement is amended
03	afterCancel	After Agreement is canceled
04	afterClone	After Agreement is Cloned
05	afterComplete	Called After Every Action (ActionType is Sent as Parameter to the method)
06	afterCreate	After New Agreement is created
07	afterCreateChild	After New Child Agreement is Created
08	afterCreateExecuted	After create of Store Executed Agreement

S.No	Callback Method	When is the callback method invoked
09	afterCreateFromOpportunity	After New Agreement is created for Opportunity
10	afterCreateOffline	After offline Agreement is created
11	afterExpire	After Agreement is expired
12	afterRenew	After Agreement is Renewed
13	afterTerminate	After Agreement is Terminated

IAgreementLifecycleCallback2

```

global class APTS_CustomAgreementLifecycleCallback implements
CustomClass.IAgreementLifecycleCallback2 {

    global void afterActivate(APTS_Agreement__c originalSO, CustomClass.ActivationType
activationType){
        logMessage('Action after Activate' , 'After Activate');
    }

    global void afterAmend(APTS_Agreement__c originalSO, APTS_Agreement__c amendmentSO){
        logMessage('Action after Amend' , 'After Amend');
    }
    global void afterCancel(APTS_Agreement__c agreementSO){
        logMessage('Action after Cancel' , 'After Cancel');
    }
    global void afterClone(APTS_Agreement__c originalSO, APTS_Agreement__c cloneSO){
        logMessage('Action after Clone' , 'After Clone');
    }
    global void afterComplete(APTS_Agreement__c agreementSO, CustomClass.ActionType
actionType){
        logMessage('Action after Complete' , '+actionType);
    }
    global void afterCreate(APTS_Agreement__c agreementSO){
        logMessage('Action after Create' , 'After Create');
    }
    global void afterCreateChild(APTS_Agreement__c parentSO, APTS_Agreement__c childSO){
        logMessage('Action after Create Child ' , 'After Create '+childSo);
    }
    global void afterCreateExecuted(APTS_Agreement__c agreementSO){
        logMessage('Action after Executed' , 'After Execute '+agreementSO);
    }
    global void afterCreateFromOpportunity(APTS_Agreement__c agreementSO){
        logMessage('Action after Create agreement from Opp' , 'After Create');
    }
    global void afterCreateOffline(APTS_Agreement__c agreementSO){
        logMessage('Action after Create offline' , 'After Create');
    }
    global void afterExpire(APTS_Agreement__c agreementSO){
        logMessage('Action after Expire' , 'After Expire');
    }
    global void afterRenew(APTS_Agreement__c originalSO, APTS_Agreement__c renewalSO){
        logMessage('Action after Renew' , 'After Renew');
    }
    global void afterTerminate(APTS_Agreement__c agreementSO){
        logMessage('Action after Terminate' , 'After Terminate');
    }
}

```

```
private void logMessage(String message , String action){  
    System.debug('[LifecycleCallback ('+action+')] :'+message);  
}  
}
```

Apttus Copyright Disclaimer

Copyright © 2019 Apttus Corporation (“Apttus”) and/or its affiliates. All rights reserved.

No part of this document, or any information linked to or referenced herein, may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written consent of Apttus. 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 Apttus 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: Apttus 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. Apttus and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Apttus and X-Author are registered trademarks of Apttus 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. Apttus 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. Apttus 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. Apttus 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.apttus.com>.

DOC ID: CMSFWIN19APIG20191129