



Expere® Document Services

# Requirements Editor

February 2016



## Preface

This publication was written for Expere® Document Services

**Publication Information / Version**  
Document Title: Requirements Editor  
Release Date: February 2016

### **Distributed Subject to Terms of a License or other Agreement**

The contents of this publication, including its appendices, exhibits, and other attachments, as updated or revised, are highly confidential and proprietary to Wolters Kluwer Financial Services, Inc. or its subsidiaries or affiliates (“Wolters Kluwer Financial Services”). This publication is distributed pursuant to a Non-Disclosure Agreement, Evaluation Agreement, License Agreement and/or other similar agreement(s) with Wolters Kluwer Financial Services, Inc. or its subsidiary or affiliate. Unless otherwise specifically provided in such agreement(s), the reproduction of this publication is strictly prohibited. Use and distribution of this publication are also subject to the responsibilities and obligations of such agreement(s), which require confidential treatment of this publication and its contents.

Information in this guide is subject to change without notice and does not represent a commitment on the part of Wolters Kluwer Financial Services.

### **Do Not Reproduce or Transmit**

Unless otherwise specifically authorized in the agreement or license under which this publication has been provided, no part of this publication may be posted, played, transmitted, distributed, copied or reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, or retaining on any information storage and retrieval system, without prior written permission from Wolters Kluwer Financial Services.

Requests for permission to reproduce content should be directed to Wolters Kluwer Financial Services, Inc., Corporate Legal Department, by telephone at 1-800-397-2341.

### **Not a Substitute for Legal Advice**

This publication is intended to provide accurate and authoritative information about the subject matter covered based upon information available at the time of publication. Examples given in this publication are for illustrative purposes only. Development of this publication and the software (including forms, disclosures, reports, and other documents generated by the software) or other products that it describes was based on Wolters Kluwer Financial Services' understanding of various laws, regulations and commentaries. Wolters Kluwer Financial Services cannot and does not guarantee that its understanding is correct.

This publication is not intended, and should not be used, as a substitute for legal, accounting, or other professional advice. Wolters Kluwer Financial Services is not engaged in providing legal, accounting or other professional services. If legal or other professional assistance is required, you should seek the services of a competent professional. We encourage you to seek the advice of your own attorney concerning all legal issues involving the use of this publication and any products described in this publication. If your interpretations or your counsel's interpretations are contrary to those expressed in this publication, you should of course, follow your/your counsel's interpretations.

The following notice is required by law:

**Wolters Kluwer Financial Services' PRODUCTS AND SERVICES ARE NOT A SUBSTITUTE FOR THE ADVICE OF AN ATTORNEY.**

### **Warranty Disclaimer**

Except only for the warranties (if any) expressly set forth in the agreement(s) under which this publication is provided (i.e., your agreement or license for the described product), this publication is provided “as is”, and Wolters Kluwer Financial Services makes no warranty, express, implied, by description, by sample or otherwise, and in particular and without limitation, makes no implied warranties of merchantability or fitness for purpose. No modifications to this Warranty Disclaimer are authorized unless in writing and signed by the President or a Vice President of the Wolters Kluwer Financial Services entity licensing the product described in this publication.

### **Attributions and Acknowledgements**

All trademarks are the property of their respective owners.

### **Copyright Information**

©2014 Wolters Kluwer Financial Services, St. Cloud, Minnesota

This publication is the confidential information of Wolters Kluwer Financial Services. Distribution of this publication is subject to restrictions in the license or agreement under which this publication is provided to authorized Wolters Kluwer Financial Institution customers.

All rights reserved.

## Table of Contents

<b>Introduction.....</b>	<b>1</b>
What's New.....	1
Documentation and Online Help.....	1
Accessibility in Tagged PDF Implementation Guide.....	2
Technical Support.....	2
<b>Getting Around the Requirements Editor.....</b>	<b>3</b>
About the Interface.....	3
Menus.....	11
Customization Options.....	15
Error Log and the Application Data Folder.....	18
Authoring Perspectives.....	19
Outline Editor.....	20
Form View (Static Documents).....	21
Field Grid.....	22
Package Rules.....	24
Notes.....	25
PDF Outline.....	26
Preview.....	27
Instance Rules Tab.....	28
Instance Display Rules (Static Documents).....	29
Arranging and Docking Windows.....	30
Floating Windows.....	30
Docking Windows.....	31
Hiding/Minimizing Windows.....	33
Resetting Window Layouts.....	33
View Windows.....	34
Attributes.....	34
Cell Contents.....	35
Dynamic Help.....	37
Active Conditions.....	38
PTR Editor.....	39
Precedence Status.....	40
Build Errors.....	41
Content Errors.....	42
Messages.....	43
<b>Creating Requirements Files (REQ Files).....</b>	<b>45</b>
Authoring Dynamic REQ Files.....	46
Create a New Dynamic Requirements File.....	47
Authoring Static REQ Files.....	49
Create a New Static Requirement File.....	50
Authoring Inherited REQ Files.....	52
Create a New Inherited Requirement File.....	52
Authoring EPX Files.....	53

Creating a Requirements File from an Existing File (Save As).....	53
Create a New Requirements File From an Existing File.....	53
<b>Setup and Configuration Features.....</b>	<b>55</b>
Configuring for Expere Content Repositories.....	55
Configuring for E-Forms Rendition Integration.....	57
Configuring for DocViewer Integration.....	57
Working with Stylesheets (StyleGuides).....	58
About StyleGuides.....	58
Updating Style Sheets.....	60
Stylesheet Options and Override Details.....	60
Working with Schema Files.....	61
Managing Fonts.....	62
<b>Understanding Key Concepts.....</b>	<b>65</b>
About Business Logic.....	65
About Precedence.....	65
About Instancing.....	66
Instance Formatting Metadata.....	67
Duplicate an Instance.....	69
About References.....	70
About Content References.....	70
Document References.....	71
About Internal References (XREF).....	71
Product Technical Rule (PTR) Reference.....	73
<b>Assembling Documents.....</b>	<b>76</b>
About Document Assembly.....	76
Assembling (Building) a Document.....	76
Executing a Document.....	77
Executing Document Package Rules.....	79
<b>Working with Transactions.....</b>	<b>81</b>
About Transactions and Transaction Data.....	81
Authoring Transaction Data.....	82
Creating Schema-based Transactions.....	83
Creating Transactions Associated with a Document.....	83
Creating Document Rule-based Transactions.....	84
Creating Package Rule-based Transactions.....	84
Importing Transactions.....	85
Saving Transactions.....	85
Renaming Transactions.....	85
Validating Transactions.....	86
<b>Working with the DocViewer.....</b>	<b>88</b>
<b>Working with the Rendition Plug-In.....</b>	<b>89</b>
Using Overrides.....	89
Creating and Editing Instances.....	91
Creating a Rendition.....	93
Using the Rendition Toolbar.....	94
Using Text Boxes (Boxing).....	95
<b>Working with the Document Editor.....</b>	<b>96</b>

Using the Document Editor.....	97
Document Editor Toolbar Features OPEN.....	98
<b>Working with Dynamic Requirements Documents.....</b>	<b>99</b>
About Dynamic Document Structure.....	99
Iterators.....	101
Tagline.....	101
Metadata.....	102
Prolog.....	104
Body (Section).....	104
Section.....	105
Viewing Dynamic Document Source XML.....	107
Working with Elements/Attributes.....	107
Moving items using Drag/Drop.....	107
Cut/Copy/Paste.....	107
Undo and Redo.....	108
Child Elements.....	108
Attributes.....	108
Inserting Elements Before/After.....	108
Using Rules.....	109
Viewing Rules.....	109
Notes .....	109
Defined Terms.....	109
Emphasis.....	110
Data.....	111
Emphasis Elements and Tags.....	115
Implementing Barcodes as an Ancillary Output Option.....	119
Modifying Space Between Columns.....	120
<b>Working with Static Requirements Documents.....</b>	<b>122</b>
Exporting an XTemplate.....	122
Export an XTemplate.....	122
Importing an XTemplate.....	122
Import an XTemplate.....	122
Adding Continuation Pages to Static Documents.....	123
<b>Working with Merge Sessions.....</b>	<b>124</b>
Merge Session Windows.....	124
New Merge Session.....	126
Merge Session Window.....	127
Content Item Diffs Window.....	129
Shared Content Diffs Window.....	130
REQ Diffs Window.....	131
REQ Diff Editor.....	133
Merge Notes Window.....	133
Merge Window Filters.....	134
Merge Toolbars.....	136
Context Menu Items.....	138
REQ Diff Editor.....	141
Outline Editor Tab.....	141
Field Grid Tab.....	142
Package Rules Tab.....	144

Notes Tab.....	145
Preview Tab.....	146
PDF Outline Tab.....	147
Instance Rules Tab.....	148
Instance Display Rules Tab.....	148
Working with Merge Session Documents.....	149
Compare with Common Base.....	149
Compare with Incoming.....	150
Three Way Merge.....	151
Accepting and Rejecting Changes.....	151
Merge Menu.....	153
<b>General Authoring.....</b>	<b>159</b>
Working with Tables.....	159
Create a Table.....	159
Table Elements and Attributes.....	160
Customizing the Header, Rows, and Borders within a Table.....	166
Changing Border Colors in a Table.....	167
Defining the Number of Rows (SortByWrapper Element).....	167
Removing Padding between Table Headings.....	171
Signature Areas.....	172
SignatureSection Element.....	176
Editing a ContentReferenced Signature Section.....	176
Editing Signature Height and Width.....	177
Add the HighlightSignatureIndicator to the Transaction.....	179
Adding Features to a Signature Area.....	180
Inserting a Signature Image.....	183
Working with eSignatures.....	184
Signature Guidelines for Static Documents.....	184
Working with Images.....	188
Adding an Image File.....	188
Adding an Image Reference to File System.....	188
Adding an Image Reference to Transaction Data.....	188
Finding Rules Across a Collection.....	189
Exporting Field Grid Information.....	189
Output Formats.....	189
Output Format Reference.....	190
Using the Field Grid to Navigate to a Data Point Location.....	195
Defining Measurement Values.....	196
Using Text Formatting Tags within DataPoints.....	197
Using Spell Check within Dynamic Documents.....	197
Author Fillable Field or Fillable Checkbox.....	198
<b>Appendix.....</b>	<b>206</b>
Icons.....	206



## Introduction

The Requirements Editor is a content authoring and document development application using structured mark-up (XML) for content editing and validation. The Requirements Editor is the primary authoring component of the Expere Authoring Environment. The application is used to produce an REQ file that combines the content of one or more documents and the rules for producing a unique document given a particular transaction data set. Authored REQ files can be submitted for assembly and generated as output documents for non-production uses such as testing, troubleshooting, and other review activities.

This guide provides practical information on using the Requirements Editor to author new content, edit existing content, and assemble documents for non-production output to PDF. It also describes the basic process of authoring structured mark-up documents and their transformation to targeted outputs.

**Note:**

While the content is XML, the Editor has specific functionality that cannot be met by commercial XML editors. Many XML editors insert special characters and comments that are not allowed in the Editor; it is important that REQ files are only edited in the Editor. It is also important to note that successive versions of the Editor introduce new features that are then used to author Expere Knowledge Base content; content authored in a newer version of the Editor may not work in an older version so it is important to make sure that the Editor and the released content are compatible.

## What's New

This section summarizes key enhancements and important changes made to the Requirements Editor, including changes made to the documentation. See the *Release Notes* for a detailed description of new features, enhancements, and defect corrections introduced in each Requirements Editor release.

The documentation was last updated on June 3, 2014.

Release	Date	Module	Description
April 2015	04/29/2015	Documentation	Added section <a href="#">Implementing Barcodes as an Ancillary Output Option</a>
BETA	06/03/2014	BETA Release	Placeholder for release summary description.

## Documentation and Online Help

The Requirements Editor includes online help available from within the application through the **Help** menu.

- To access the online help, go to the **Help** menu and click **Requirements Editor Help**.
- To access the online help outside of the application, go to the [Help Center](#).

The **Help Center** provides the online help as well as a PDF version of the help and additional documentation supporting the Expere system.

- The Requirements and Content Editor documentation collection includes: *Requirements Editor Online Help*, *Requirements Editor User Guide* (a PDF version of the online help), and *Requirements Editor Installation Guide*.

The name and version of applications, libraries and frameworks used by the Requirements and Content Editor are listed in the **About** dialog box. In the Help menu, click **About Requirements Editor**. The **About** dialog box displays the installed versions of each installed component and system properties. This information is particularly useful when contacting customer support.

## Accessibility in Tagged PDF Implementation Guide

If you have purchased the Tagged PDF Outputs add-on license, refer to the [Accessibility in Tagged PDF Outputs Implementation Guide](#) for authoring details.

## Technical Support

If you have any questions regarding the Requirements and Content Editor, you may contact the Wolters Kluwer Financial Services support team by phone or email. Support staff is available Monday through Friday, 7 A.M. to 7 P.M. CST. Voice mail is available after hours and on weekends.

- Expere Support is available at:
  - Tel: 1-800-274-2711, ext. 1126618
  - E-mail: [ExpereIESupport@wolterskluwer.com](mailto:ExpereIESupport@wolterskluwer.com)

When contacting the support team, it is helpful to provide the following information:

- The version/build number (from the Help > About dialog box).
- Any error message.
- The steps take to reproduce the error.

**Attention:**

There may be situations where you provide or are asked to provide data to Wolters Kluwer Financial Services for assistance in troubleshooting, implementation, or the delivery of professional services. When doing so, do not provide actual live data of your customers. Provide sample, fake data that is representative of the issue you are experiencing.

## Getting Around the Requirements Editor

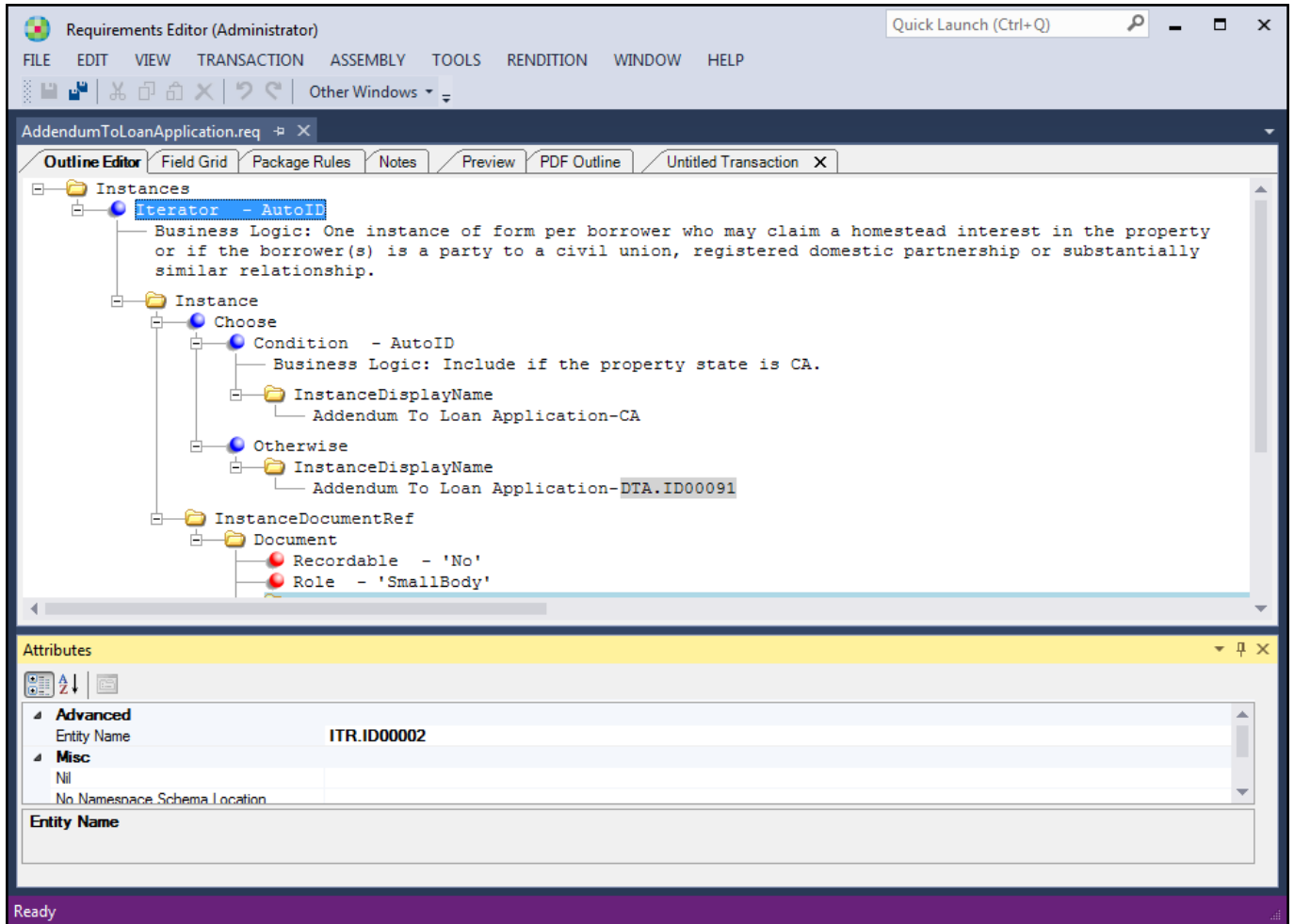
This section provides a quick look at the Requirements Editor interface in terms of how authoring features and information about a requirements file, or an REQ-based document, is presented in the application.

- [About the Interface](#)
- [Customization Options](#)
- [Error Log and the Application Data Folder](#)
- [Authoring Perspectives](#)
- [View Windows](#)

### About the Interface

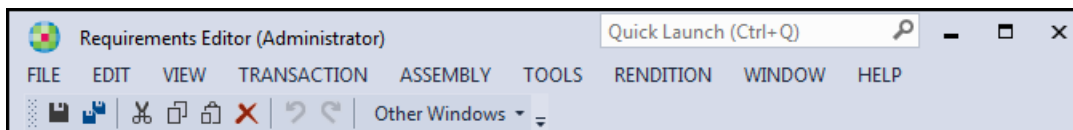
The Requirements Editor is built on a Visual Studio framework and incorporates many of the navigation features available in Visual Studio. The visual experience is highly customizable but centers on a group of menu items, toolbars, and windows.

This example displays a default view with a document opened in the Outline Editor.

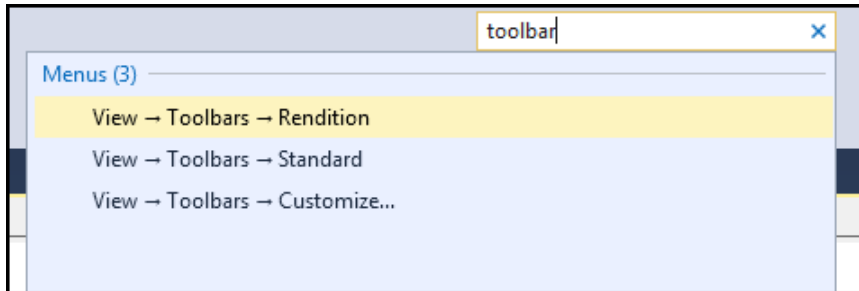


## Quick Launch

At the top of the screen is a Quick Launch feature. You can enter an action, file name, command, menu item, toolbar, or any variety of criteria and a list of related actions, documents, or other items is presented for selection.



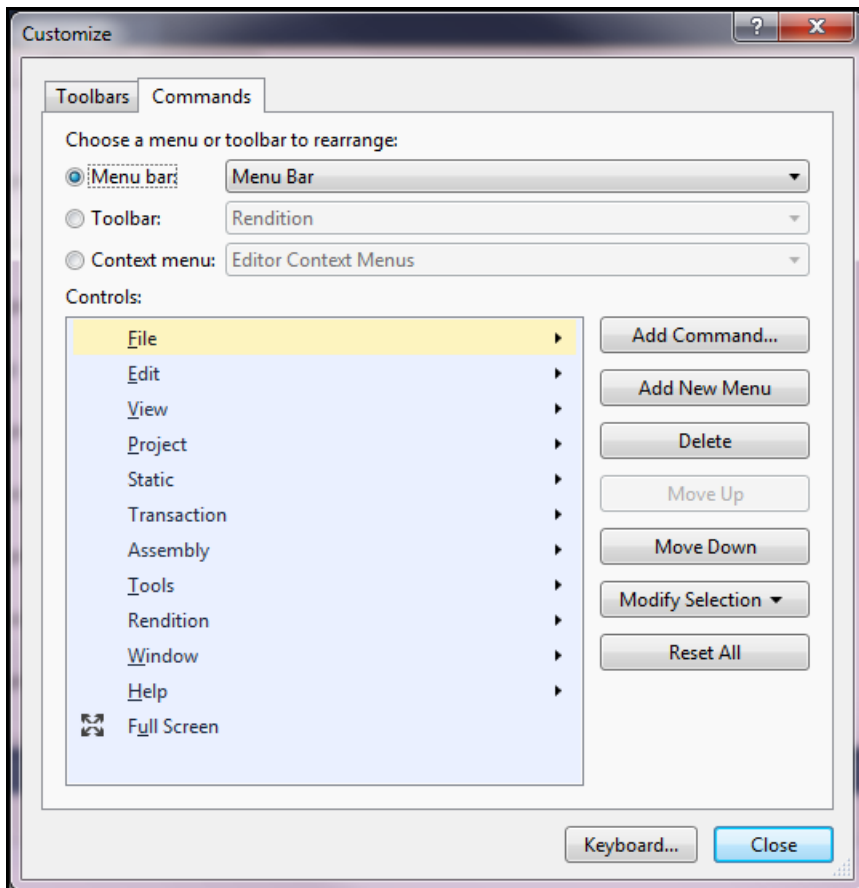
The Quick Launch feature can be disabled and customized in the **Options** dialog box (**Tools > Options**). Select **Quick Launch** in the **Environment** options section. This can be a quick way to open or launch a particular feature. In this example, we type in the text *toolbar* and the Quick Launch lists several toolbar actions.



## Menus

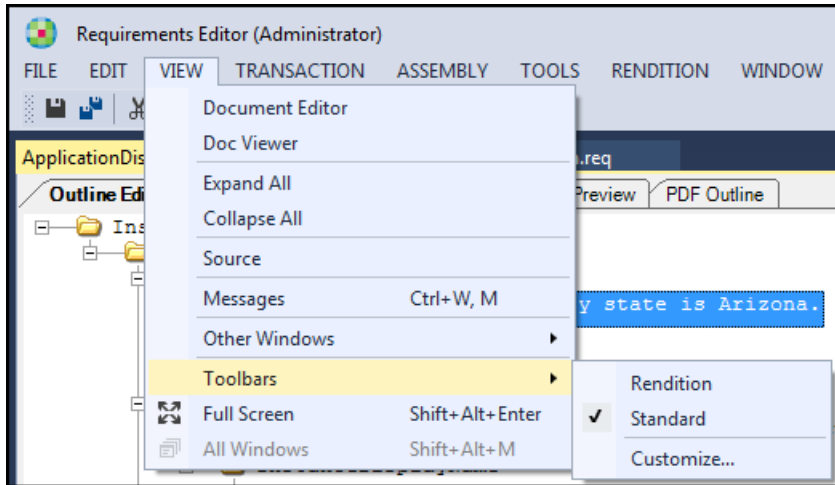
The Requirements Editor menu bar provides access to all of the features available in the application. Each menu item is context sensitive meaning that options are enabled and disabled depending on where you are in the application and the features available to you based on the Requirements Editor edition and advanced features you have installed.

You can modify the menus in the **Customize** dialog box (**Tools > Customize**). In the **Customize** dialog box, select the **Commands** tab and select **Menu bar**.

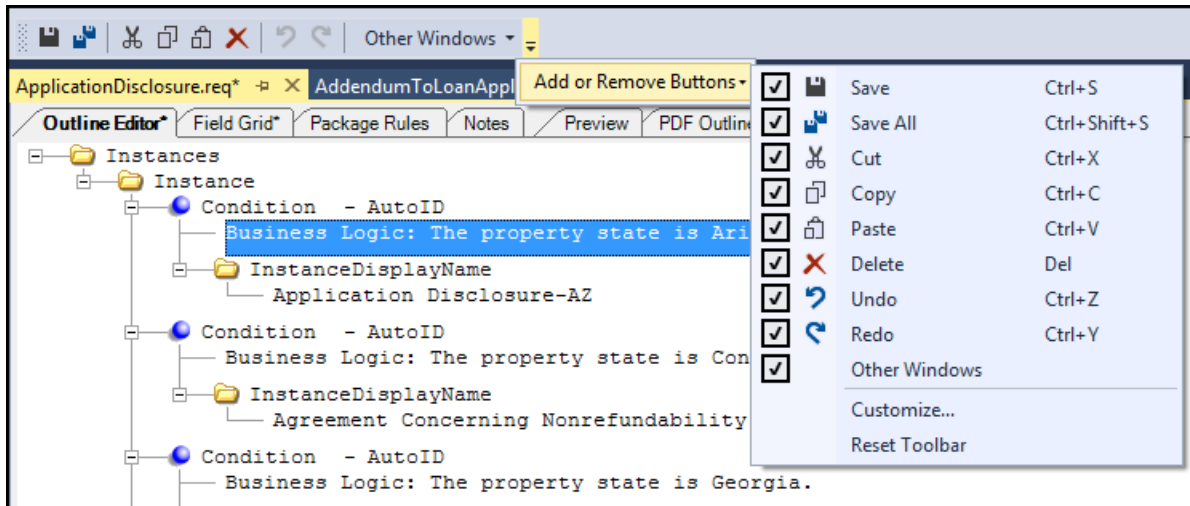


## Toolbars

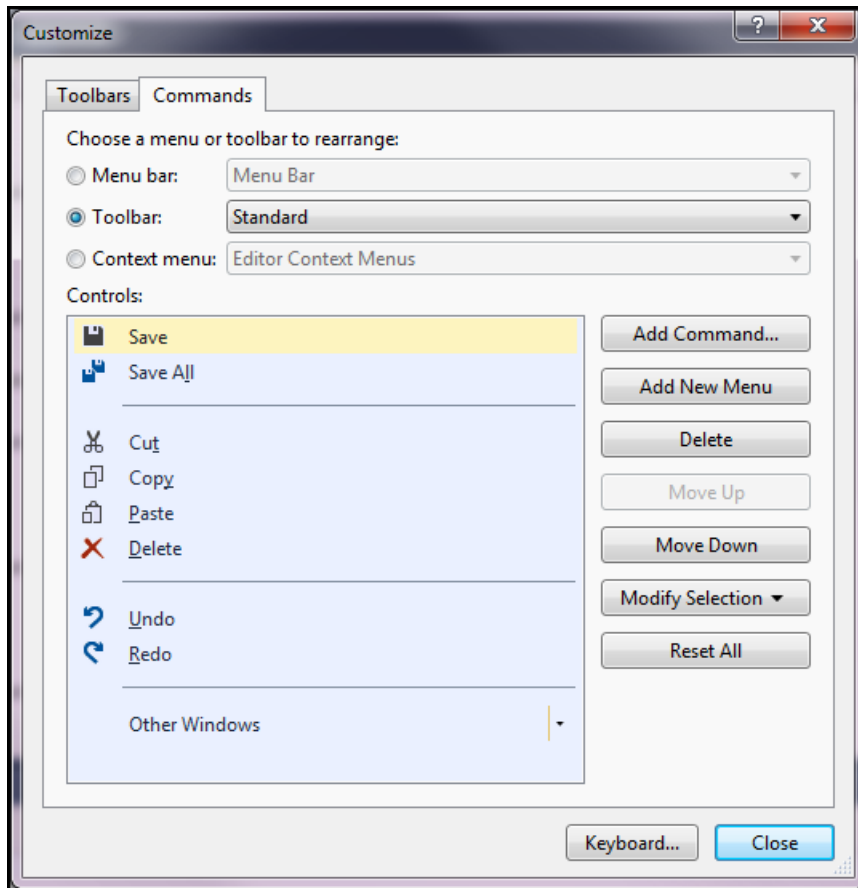
The Requirements Editor provides a standard toolbar and additional toolbars specific to features available to you based on the Requirements Editor edition and advanced features you have installed. The standard toolbar is displayed by default, to show or hide toolbars select **Toolbars** in the **View** menu.



The standard toolbar provides quick options to save, cut/copy/paste, and select various windows to include on the screen. You can add and remove toolbar button by clicking the Standard Toolbar Options arrow at the right-side of the toolbar.

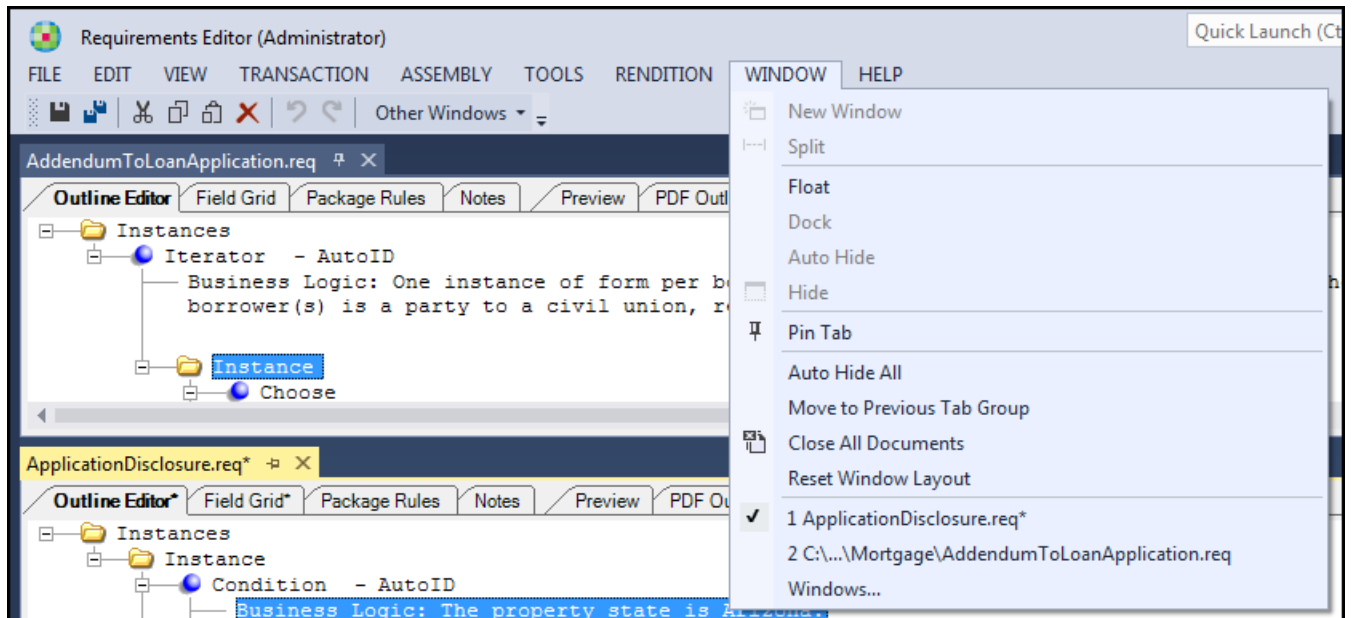


You can modify the toolbars in the **Customize** dialog box (**Tools > Customize** or click **Customize** in the **Add or Remove Buttons** drop-down box). In the **Customize** dialog box, select the **Commands** tab and select **Toolbar**.

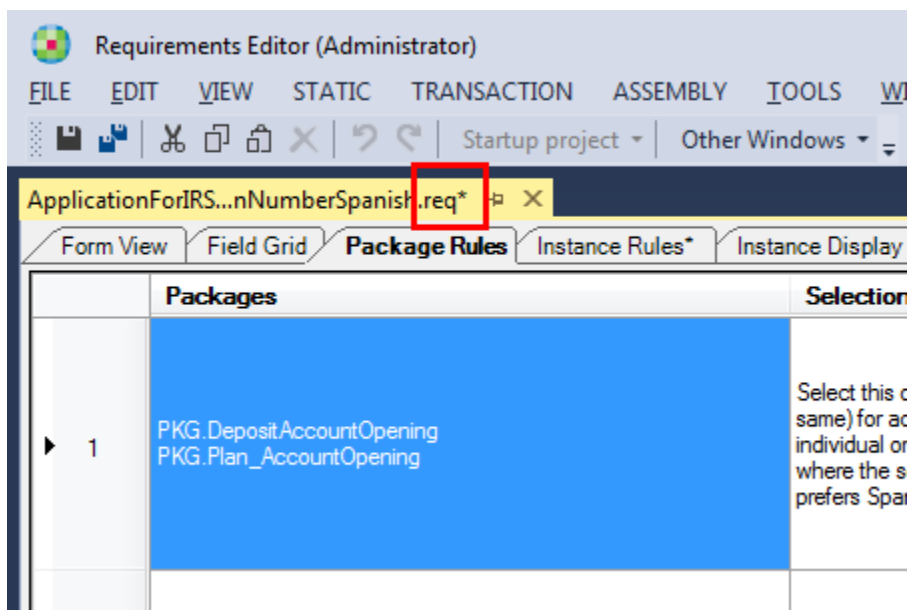


## Document Pane

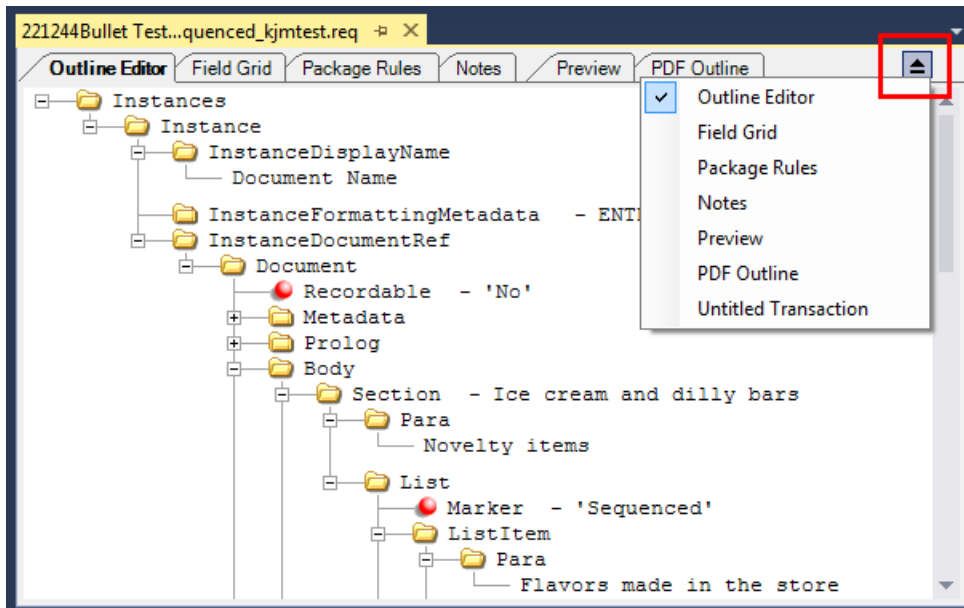
The Document pane displays a file or multiple files (and REQ file) for authoring. When documents are opened, each is docked in the pane in a tab group. You can change how documents are presented in the document pane by selecting from the options available in the Window menu or by simply selecting a document tab and dragging it (moving it) to a new position. Some of the presentation options include: docked, floating, and horizontal or vertical placement within the group of open documents.



When a document has been modified, Requirements Editor provides a visual indicator by placing an asterisk after the filename of the document, as shown in the following example:

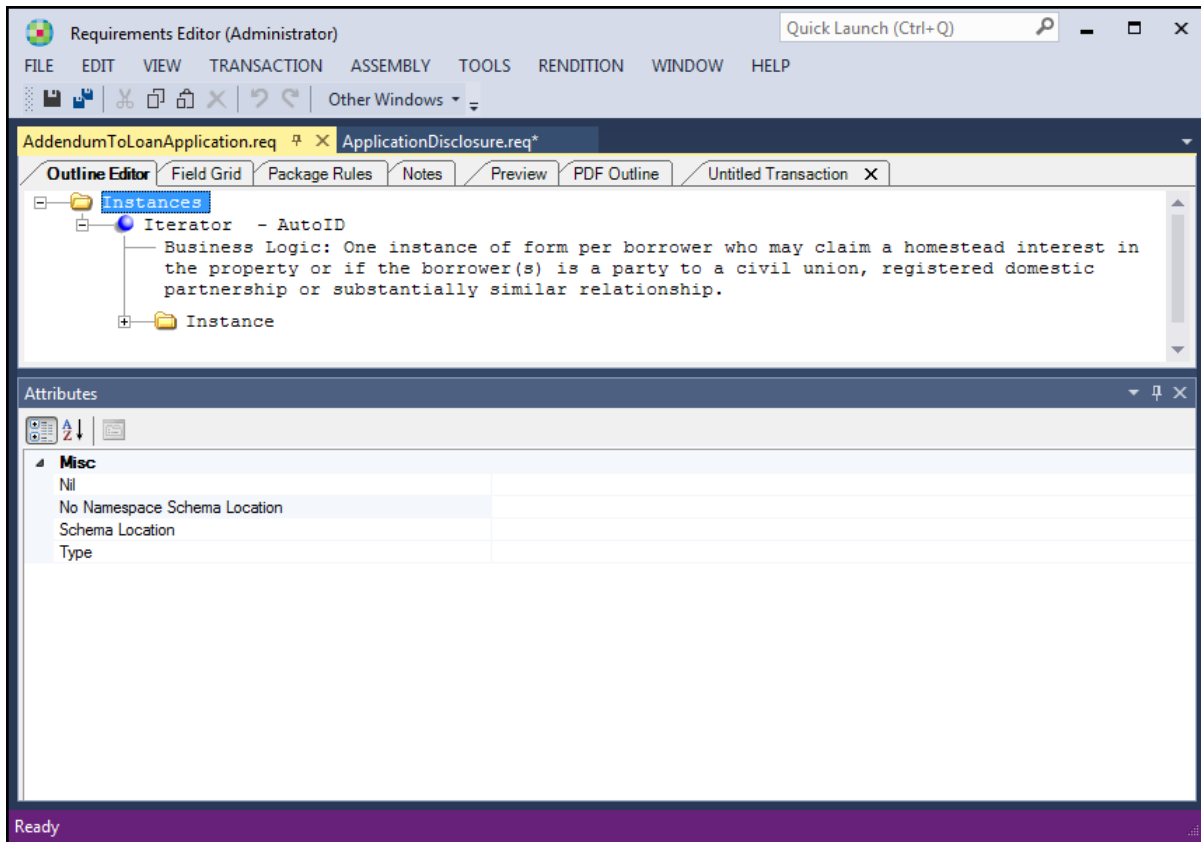


If there are more tabs than can be displayed, a drop list icon is presented, allowing you to select from a list of all tabs.

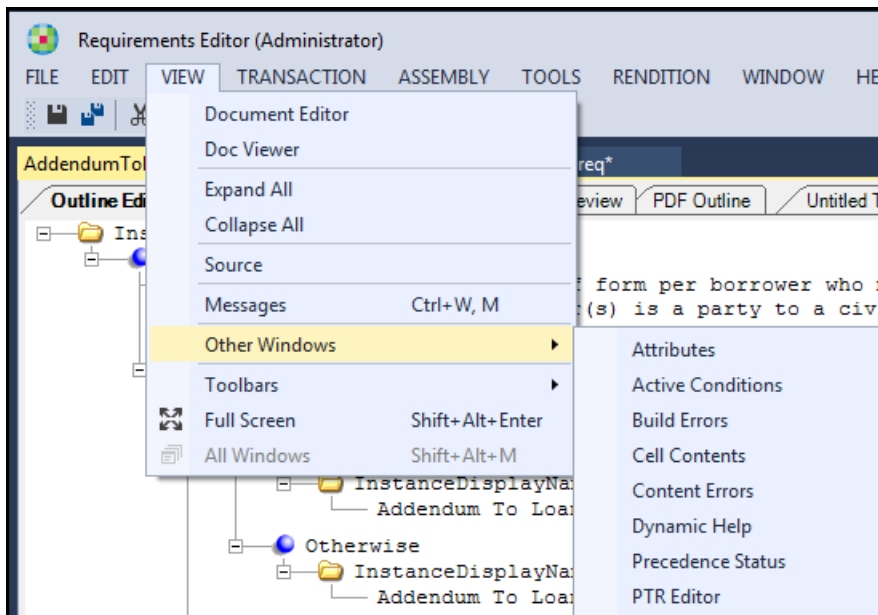


### View Windows

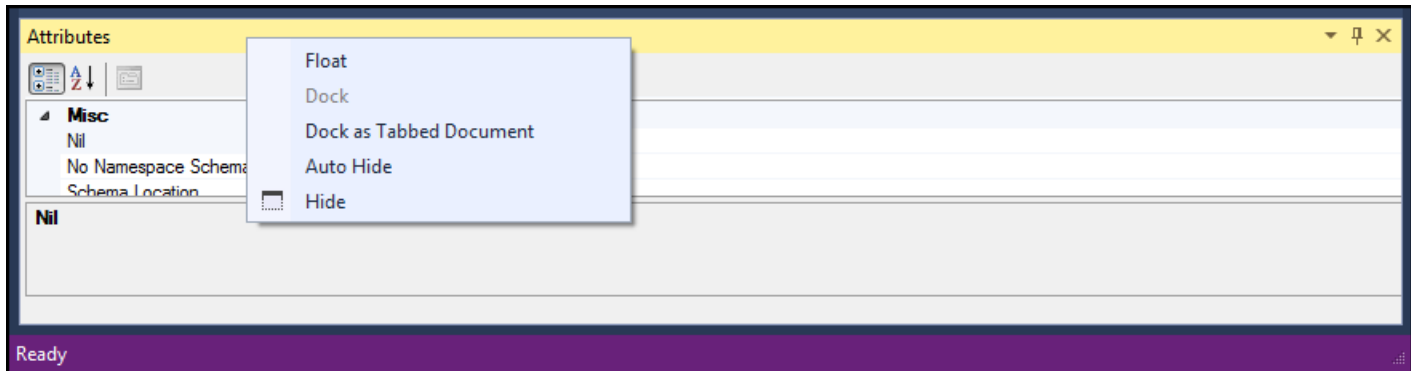
The View Windows (Windows) display specific information about the active document in the document pane. By default, these open at the bottom of the screen. In this example, the **Attributes** view window is open.



View Windows are available in the standard toolbar in the **Other Windows** drop-down list as well as in the **View** menu. In the **View** menu, select **Other Windows** and select the view you wish to open.



The View Windows can be moved in the same manner as the Document Pane windows. Select the view window by placing your cursor in the title banner to select it and drag the window to move it to another position. You can also select the drop-down arrow or right-click in the title banner to see a list of options such as float, dock, dock as a tabbed document in the document pane, and hide.



## Menus

The following sections contain detailed information on each command located within the menu bar.

### Note:

Some menu items are context sensitive, meaning that options are enabled and disabled depending on where you are in the application and the features available to you based on the Requirements Editor edition and advanced features you have installed.

### File Menu

- **New** – Mainly used for creating dynamic, inherited, or static documents. You can also select and create various types of transactions, as well as REQ Diff Editor files (for merges).
- **Open** – Used to open REQ files as well as transaction XML files.
- **Close** – Closes the file that currently has focus in the tool. This will close transaction XML or REQ files.
- **Merge Sessions (New, Open, and Close)** – These menu options are for the new merge tool functionality (refer to the Merge Tool section for details). Not included in the FOP or XEP installs.
- **Save/Save As/Save All** – Used to save changes made to the file that has focus in the tool. Save All saves all files that are open in the tool, regardless of if they have focus.
- **Properties** – Shows the different properties of the document that currently has focus.
- **Page Setup/Print Preview/Print** – All of these options relate to printing.
- **Recent Files** – This menu option allows you open files that you had worked with recently.
- **Exit** – Closes the application.

### Edit Menu

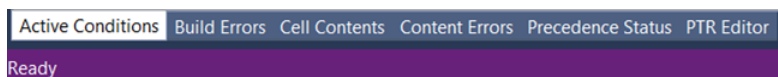
- **Undo/Redo** – Undo/Redo the last change that was made.

- **Cut/Copy/Paste** – Works like normal Cut/Copy/Paste functionality. You can use these three options on elements in the “Outline Editor” tab.
- **Duplicate** – Only usable in the “Outline Editor” tab. This creates a copy of the currently selected element (or elements). The copy is added to the tab.
- **Move Up/Down** – Only usable in the “Outline Editor” tab. This moves the currently selected element (or elements) up or down in the tree structure of elements.
- **Decrease/Increase Indent** – Only usable in the “Outline Editor” tab. This will move the currently selected element (or elements) to a shallower or deeper level in the tree structure of the elements.
- **Find** – Normal find functionality. It works in many RE 3 windows/tabs.
- **Replace** – Normal replace functionality. Users should avoid using this functionality for the time being.
- **Incremental Search** – Built in Visual Studio functionality. Currently, this does nothing. Users can ignore this for now.
- **Go To** – Built in Visual Studio functionality. Currently, this does nothing. Users can ignore this for now.
- **Navigate To** – Built in Visual Studio functionality. Currently, this does nothing. This can be ignored.

#### View Menu

- **Doc Viewer** – Allows you to see the document through a Doc Viewer web interface (a web page). Not included in the FOP or XEP installs.
- **Expand/Collapse All** – Only usable in the “Outline Editor” tab. This expands or collapses all tree nodes in the tab.
- **Messages** – Built in Visual Studio functionality. Currently, this window/tab shows nothing. Users can ignore this for now.
- **Merge Session** – Only applies to merges. Makes the merge session windows appear. Not included in the FOP or XEP installs.
- **REQs Diffs** – Only applies to merges. Makes the REQ Diffs window appear. Not included in the FOP or XEP installs.
- **Content Item Diffs** – Only applies to merges. Makes the Content Item Diffs window appear. Not included in the FOP or XEP installs.
- **Incoming REQ** – Only applies to merges. Makes the Incoming REQ window appear. Not included in the FOP or XEP installs.
- **Existing REQ** – Only applies to merges. Makes the Existing REQ window appear. Not included in the FOP or XEP installs.
- **Shared Content Diffs** – Only applies to merges. Makes the Shared Content Diffs window appear. Not included in the FOP or XEP installs.
- **Merge Notes** – Only applies to merges. Makes the Merge Notes window appear. Not included with the FOP or XEP installs.

- **Other Windows** – Clicking on each of these options will show their respective window/tab. These options (minus Dynamic Help) are also tabs on the bottom of your “Outline Editor” tab.



- **Toolbars** – Allows you change which toolbars appear. You can also customize the toolbars and menu options.
- **Full Screen** – Changes the view to a full screen one.

### Static Menu

**Note:**  
These options appear for static documents only.

- **Find Rules for this Public Name** – Used to find rules for a given public name.
- **Find Definition for this Public Name** – Used to find definitions for a given public name.
- **Find Rules and Definitions for All Public Names** – Performs a combination of the two options above.
- **Verify Revised Rules** – Verifies any revisions for rules.
- **Import/Export xTemplate** – Used to import or export xTemplates for static documents.
- **Continuation Page Setup** – Used to add continuation pages to static documents.

### Merge Menu

**Note:**  
These options appear for internal installations or Requirements Editor 3.

- **Generate Results & Update Content References** – Performs the function of the next two options.
- **Generate Results Only** – Creates all the folders and files containing the merge results.
- **Update Content References** – Updates the content references present in the merge session.
- **Expere Content Build ERL Format** – Builds the repository content in the ERL format.
- **Generate Impact Analysis Report** – Creates a webpage detailing the differences between the repositories.

### Transaction Menu

- **Create Transaction from Document Rules** – You can opt to create a transaction internally or externally. If a user opts for an external transaction, they create a transaction xml file and then build it within RE 3. The external transaction isn’t linked to the REQ, rather it’s treated as another tab for just working on that xml.
- **Import Transaction** – Used to import transaction XML.
- **Save Transaction As** – Used to save transactions.
- **Change Transaction Name** – Used to change the name of transactions.

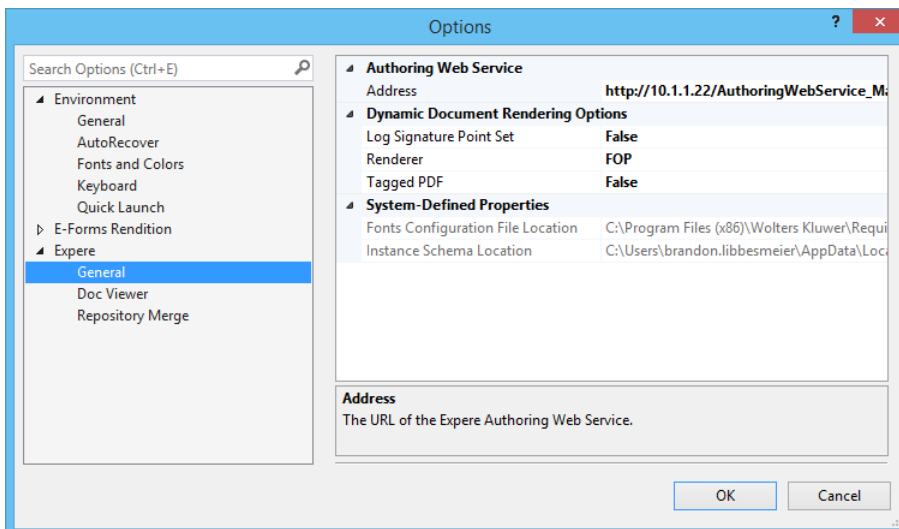
- **Validate Transaction** – Used to validate whether the transaction XML is correctly constructed.

### Assembly Menu

- **Build** – Runs a syntax check on REQs. Any incorrect structure or PTR will show build errors.
- **Execute Document** – Creates and displays a PDF version of the document, based on the transaction XML you provide.
- **Execute Package Rules** – Creates and displays a PDF which outlines each package rule and if the document was autoselected or suggested for each of them.

### Tools Menu

- **Find Forms Using this Rule** – Shows documents that use a given rule. This option only appears for static documents.
- **Export Field Grid** – Used to export the field grid contents as an Excel document.
- **Spell Check** – Performs a spell check. Currently, this has limited use.
- **Update Style Sheets** – Gets updated versions of style sheets used for authoring.
- **Add Custom Fonts** – Allows a user to add custom fonts for their content.
- **Customize** – Lets you customize your menu options and toolbars. Users shouldn't need to use this.
- **Options** – Displays an options window where various settings for RE 3 reside.



### Rendition Menu

**Note:**

These options are not included in the FOP or XEP versions of Requirements Editor 3.

- **Create Renditions** – Used to create different file format versions of the REQ file.
- **Edit Instances** – Used for EPM documents. Allows you to edit the different instances for a document.

## Window Menu

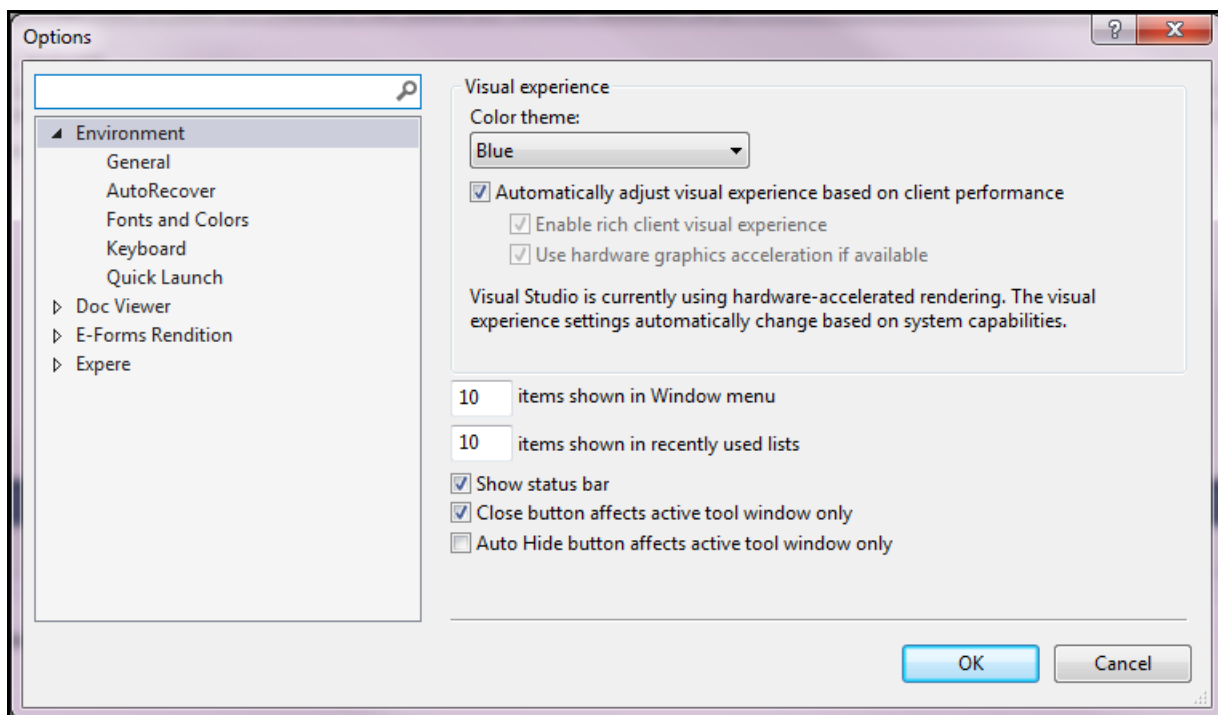
All of the menu options are used to select and position the windows and tabs.

## Help Menu

- **Requirements Editor Help** – Opens the online help webpage.
- **About Requirements Editor** – Displays version information and installed packages.

## Customization Options

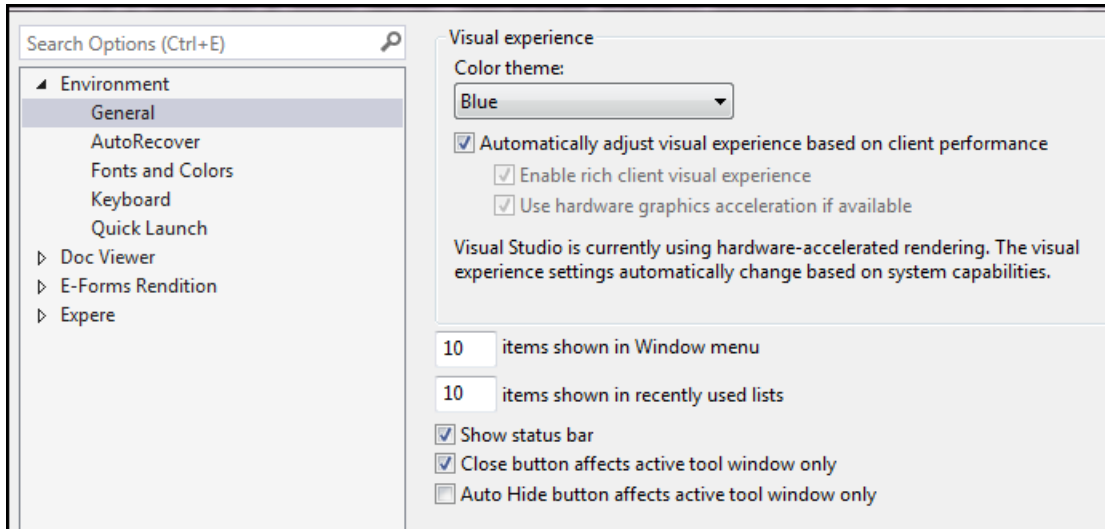
The application provides several options allowing you to define the look and feel of the interface, how toolbars are displayed, auto-recovery and start up behavior, and other features to tailor the working environment. These customization options are available in the **Options** dialog box (**Tools > Options**).



### General

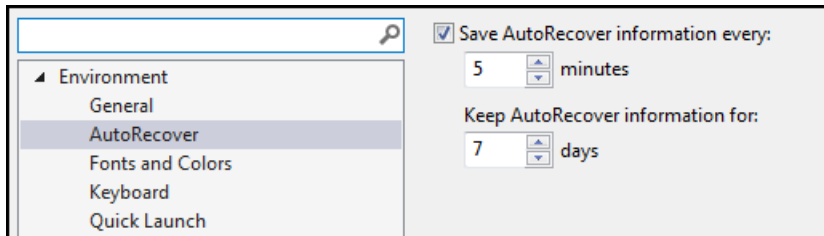
You can select a color theme to apply to the Requirements Editor. By default, the blue theme is selected. You have the option of selecting the default blue theme or either a light or dark color theme.

In addition to a color theme, the General environment options let you choose how many items to show in the **Recent Files** list (**File > Recent Files**) and in the list of available open **Windows** (Window menu).



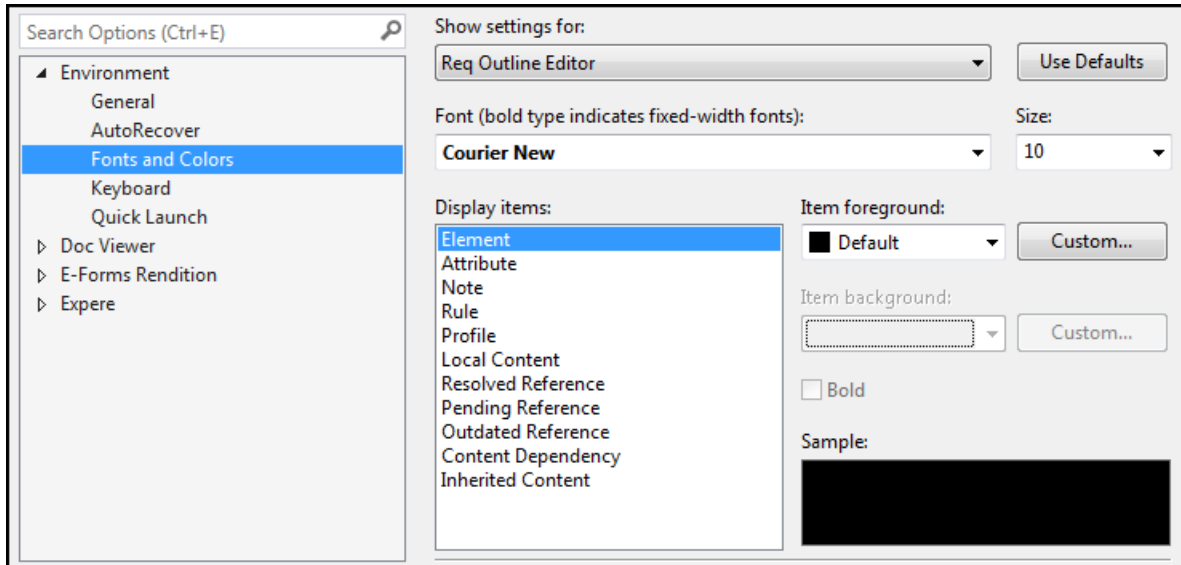
### Auto Recovery

The AutoRecover feature is used in case of an unexpected system failure to ensure that any recent work is saved and can be recovered when the application is restarted.



### Fonts and Colors

The Fonts and Colors option allows you to define how text and elements appear in the Outline Editor. You can apply a font and color settings to all text presented in the editor and define a foreground or background color to specific elements within a document.

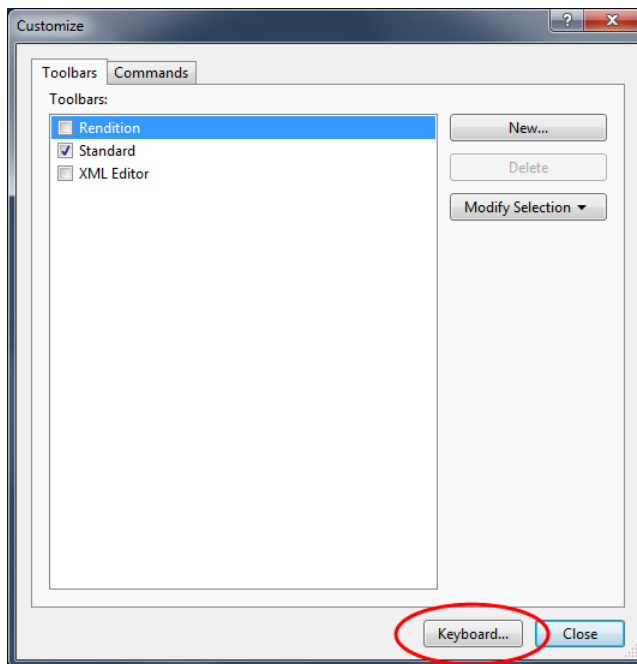


## Keyboard

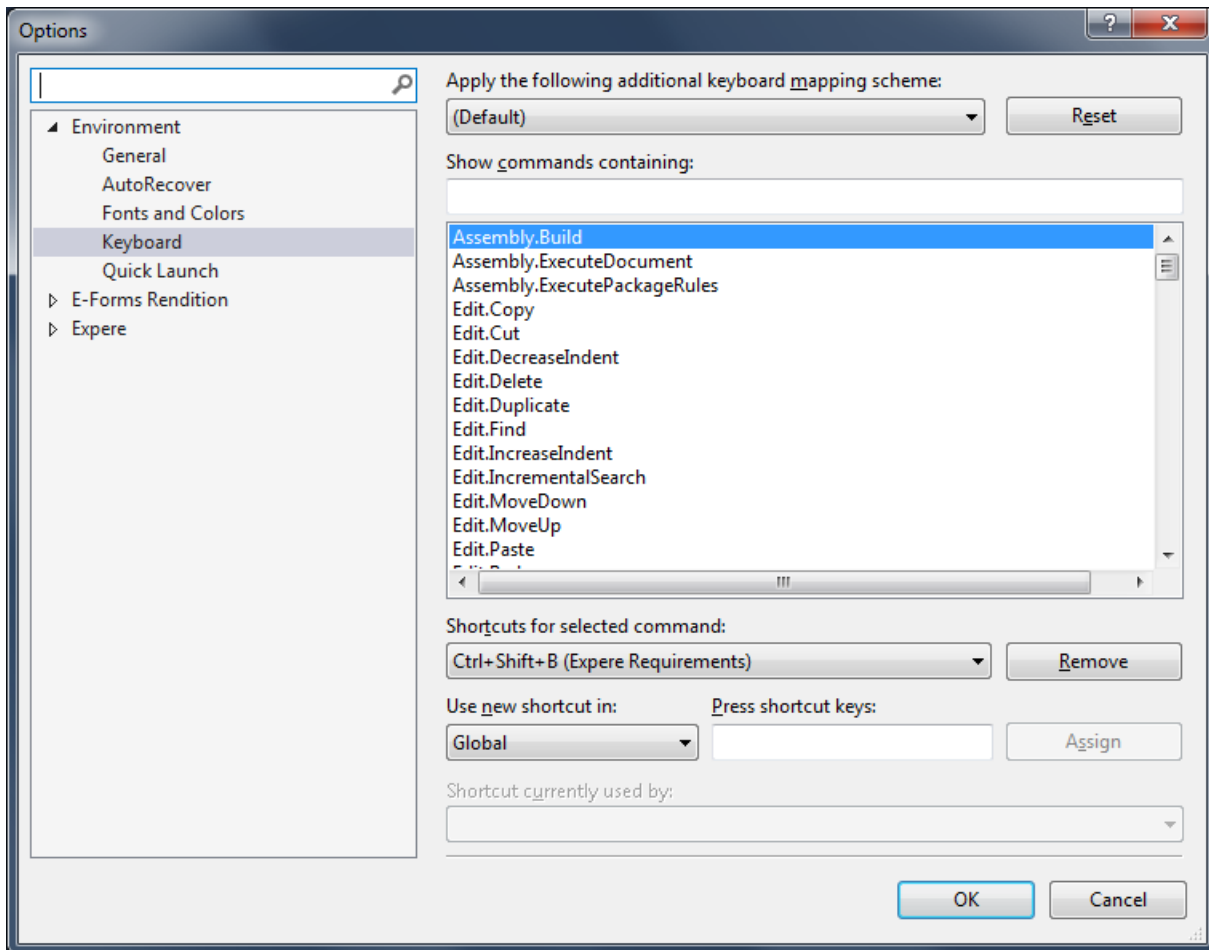
The Keyboard option is used to set keyboard shortcut keys.

### Note:

- The Keyboard options are also available by clicking the **Keyboard** button located at the bottom of the **Customize** dialog box (**Tools > Customize**).

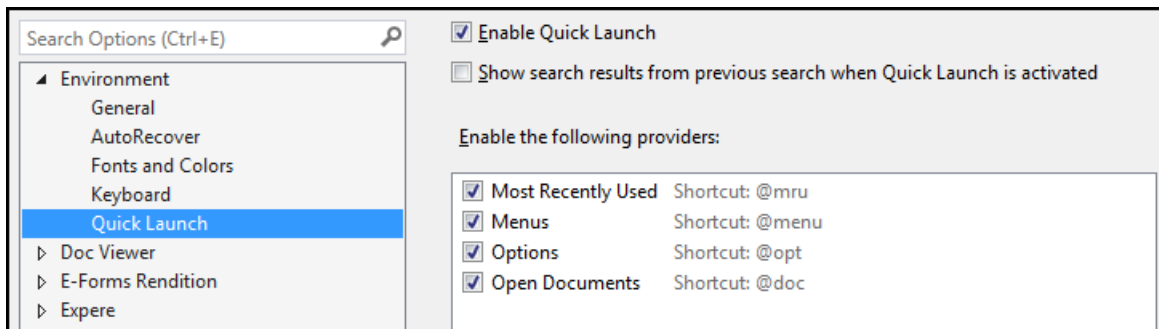


- Keyboard shortcuts are retained if you install Requirements Editor over a previously installed version.



### Quick Launch

The Quick Launch option lets you enable/disable the feature as it appears in the top-right corner of the screen. The feature is enabled by default.



### Error Log and the Application Data Folder

In addition to the Message, Build Errors, and Content Errors views available in the **View > Other Windows** menu option, an error log is generated each time a document is assembled. The error log contains a rolling list of error events and the associated trace record for each error returned by the system.

The file, *errors.txt*, is saved to the application data folder (*\AppData*) and can be opened for viewing using a text editor such as Notepad.

- File Name: *errors.txt*
- Path: *C:\Users\user.name\AppData\Local\Wolters Kluwer\RequirementsEditor\3.0*

For example, if a content reference cannot be resolved during document assembly a record is created in the error log. The *error.txt* might look like this:

```
1/22/2014 4:38:18 PM -----
Content reference(s) could not be loaded Couldn't find organization name.
Trace:
  at RequirementsEditor.ViewModel.ExternalDocumentCache.DownloadDocument(String orgID,
  String docID, Boolean reportProgress)
  at RequirementsEditor.ViewModel.ExternalDocumentCache.GetDocument(String orgID,
  String docID)
  at
  RequirementsEditor.ViewModel.ContentReferences.ContentReferencesManager.AutoUpdateContentReferences
  state)
```

This error log, along with the error reporting available in the **Other Windows** options, can be useful in troubleshooting issues and is particularly helpful in communicating errors to technical support.

The Requirements Editor stores settings and locally created files to the *Windows application data* folder for the user account that the Requirements Editor was associated with when installed. As we noted earlier, the error log (*errors.txt*) is one such file stored on this location. In addition to the error log, the Wolters Kluwer Financial Services repository that is referenced by a document during assembly is stored in this folder as well as a settings file. The files are stored as *Local* application data associated with a machine; they are not stored as *Roaming* application data files or the *LocalLow* folder.

Here is an example of the files that may be stored on the application data folder. The path is defaulted to:

```
C:\Users\User.Name\AppData\Local\RequirementsEditor\3.0
```

- Repository
  - ORG.WoltersKluwerFinancialServices.Root
  - Organizations.xml
  - Profile.xml
- errors.txt
- organizations.xml
- RequirementsEditor.settings

## Authoring Perspectives

The Authoring Perspectives are displayed in the primary window, the document pane, and display information about the active file in a series of tabbed views. These perspectives provide access to a majority of the document authoring features. The available authoring perspectives for dynamic and static documents vary and you will notice these differences depending on the type of document that you are working with.

The following authoring perspectives are available when a requirements file is opened in the editor. Each perspective shows document information available for authoring as you create a new document or edit an existing document.

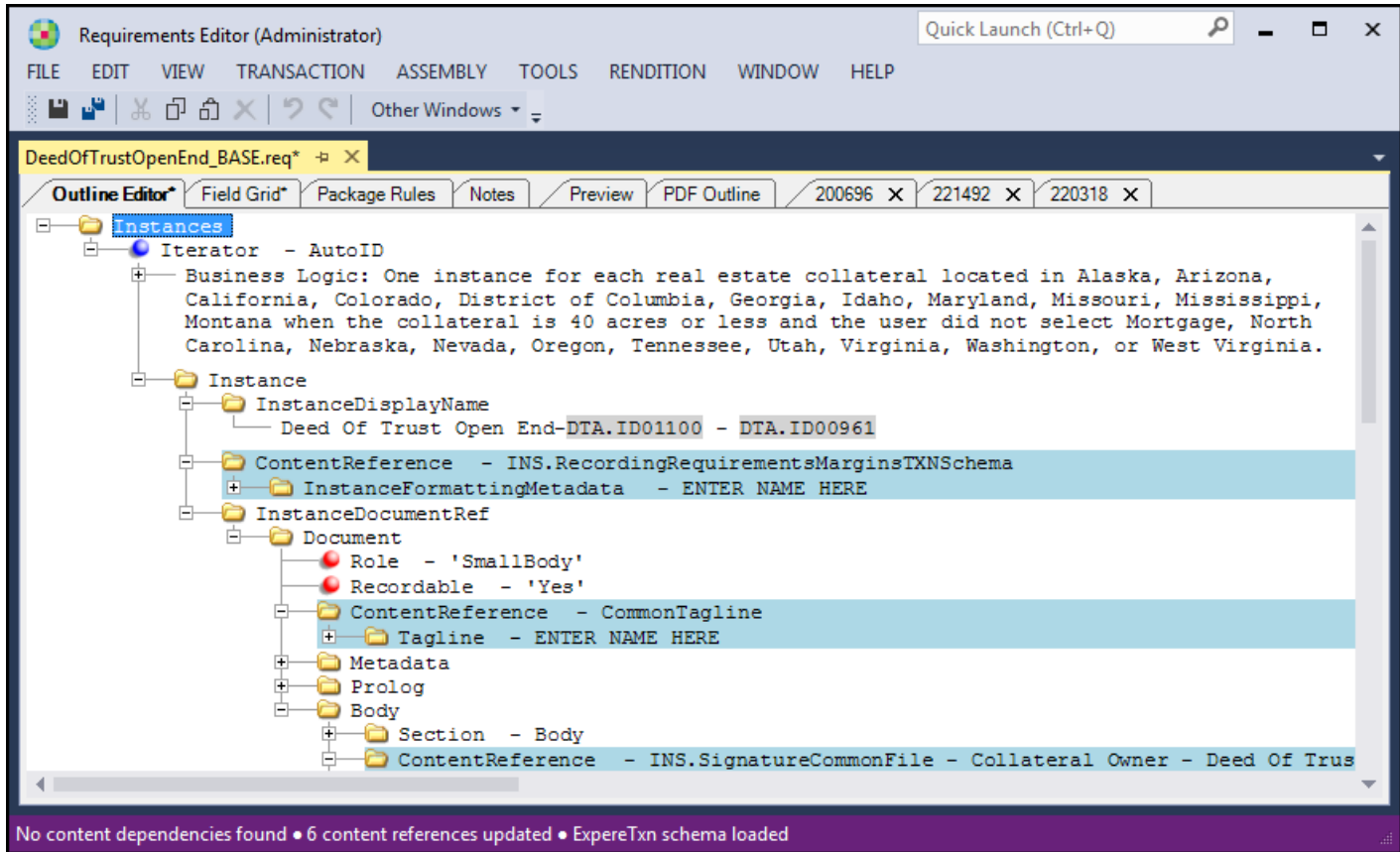
- Outline Editor
- Form View
- Field Grid
- Package Rules
- Notes
- PDF Outline
- Preview
- Instance Rules (Static Documents)
- Instance Display Rules (Static Documents)

## Outline Editor

The **Outline Editor** perspective is used as the main view to author documents; both in authoring new and editing existing requirements files. This perspective provides an XML Editor experience where the requirements file is presented in an outline based on the XML structure of the file.

Authors utilize this perspective and the main view of a requirements files for authoring and testing. It is here where you can build the file and edit it's contents by adding and removing elements and attributes to adjust the content of the document and to apply formatting requirements to the document.

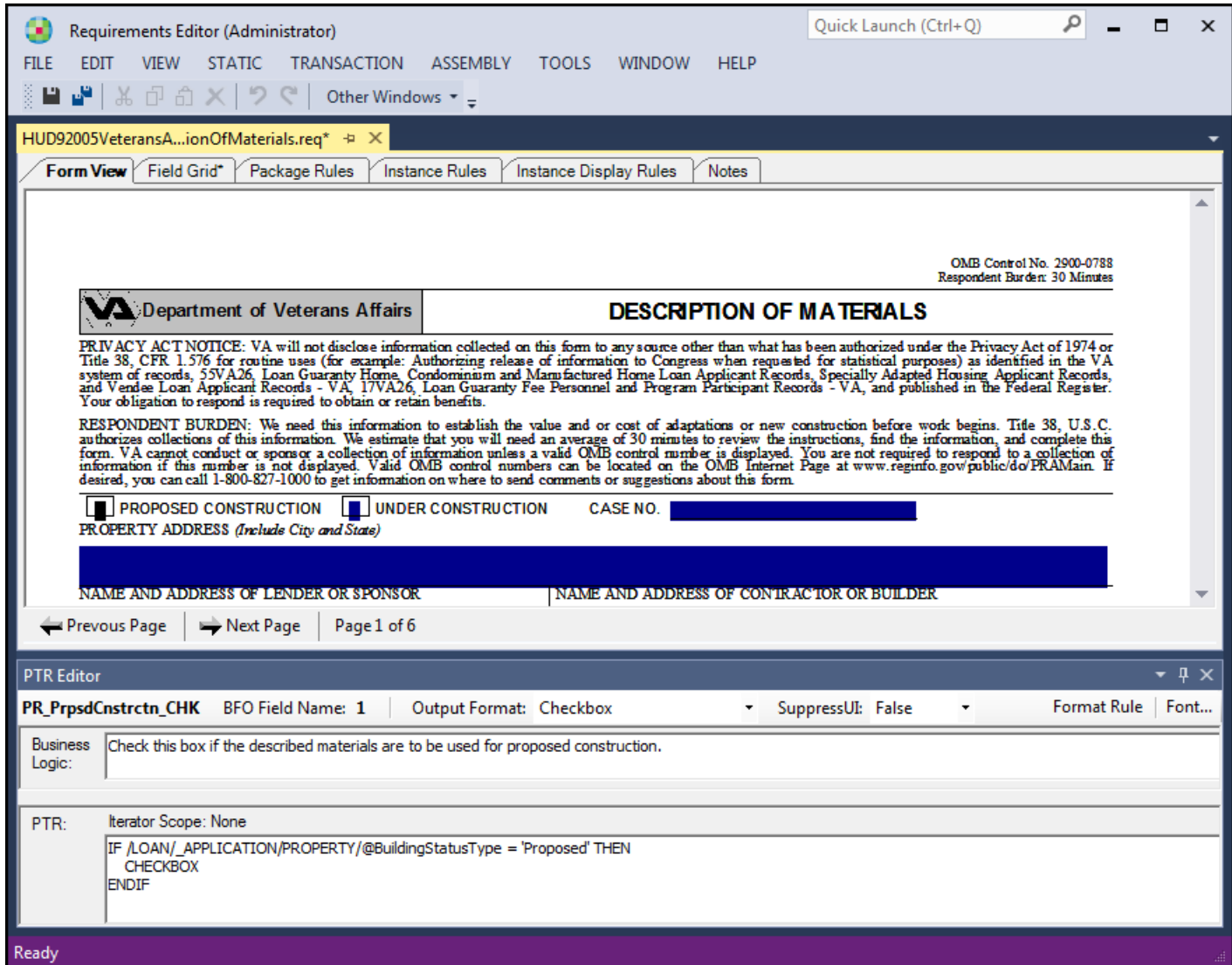
The perspective is available by clicking the **Outline Editor** tab in the document pane. The Outline Editor is available exclusively for dynamic documents and is the default perspective loaded when opening a dynamic document. This is an example of the outline editor perspective in a dynamic document.



## Form View (Static Documents)

The **Form View** perspective displays the entire document in a manner that resembles the structure and look of the form as it will appear when generated as a PDF. When using this perspective to view and edit a static requirements file, it is a common practice to open the PTR Editor view window. The PTR Editor displays the business logic and product technical rule for each field in the requirements file.

The perspective is available by clicking the **Form View** tab in the document pane. The Form View is available exclusively for static documents and is the default perspective displayed when a static requirements file is opened. This is an example of the form view perspective with the PTR Editor view window also open in a static document.



Requirements Editor (Administrator) Quick Launch (Ctrl+Q)

FILE EDIT VIEW STATIC TRANSACTION ASSEMBLY TOOLS WINDOW HELP

HUD92005VeteransA...ionOfMaterials.req\* -> X

Form View Field Grid\* Package Rules Instance Rules Instance Display Rules Notes

OMB Control No. 2900-0788  
Respondent Burden: 30 Minutes

**Department of Veterans Affairs** **DESCRIPTION OF MATERIALS**

PRIVACY ACT NOTICE: VA will not disclose information collected on this form to any source other than what has been authorized under the Privacy Act of 1974 or Title 38, CFR 1.576 for routine uses (for example: Authorizing release of information to Congress when requested for statistical purposes) as identified in the VA system of records, 55VA26, Loan Guaranty Home, Condominium and Manufactured Home Loan Applicant Records, Specially Adapted Housing Applicant Records, and Vendor Loan Applicant Records - VA, 17VA26, Loan Guaranty Fee Personnel and Program Participant Records - VA, and published in the Federal Register. Your obligation to respond is required to obtain or retain benefits.

RESPONDENT BURDEN: We need this information to establish the value and or cost of adaptations or new construction before work begins. Title 38, U.S.C. authorizes collections of this information. We estimate that you will need an average of 30 minutes to review the instructions, and the information, and complete this form. VA cannot conduct or sponsor a collection of information unless a valid OMB control number is displayed. You are not required to respond to a collection of information if this number is not displayed. Valid OMB control numbers can be located on the OMB Internet Page at [www.reginfo.gov/public/do/PRAMain](http://www.reginfo.gov/public/do/PRAMain). If desired, you can call 1-800-827-1000 to get information on where to send comments or suggestions about this form.

PROPOSED CONSTRUCTION  UNDER CONSTRUCTION CASE NO. [REDACTED]

PROPERTY ADDRESS (Include City and State) [REDACTED]

NAME AND ADDRESS OF LENDER OR SPONSOR [REDACTED] NAME AND ADDRESS OF CONTRACTOR OR BUILDER [REDACTED]

Previous Page Next Page Page 1 of 6

PTR Editor

PR\_PrpsdCnstrctn\_CHK BFO Field Name: 1 Output Format: Checkbox SuppressUI: False Format Rule Font...

Business Logic: Check this box if the described materials are to be used for proposed construction.

PTR: Iterator Scope: None  
IF /LOAN/\_APPLICATION/PROPERTY/@BuildingStatusType = 'Proposed' THEN  
CHECKBOX  
ENDIF

Ready

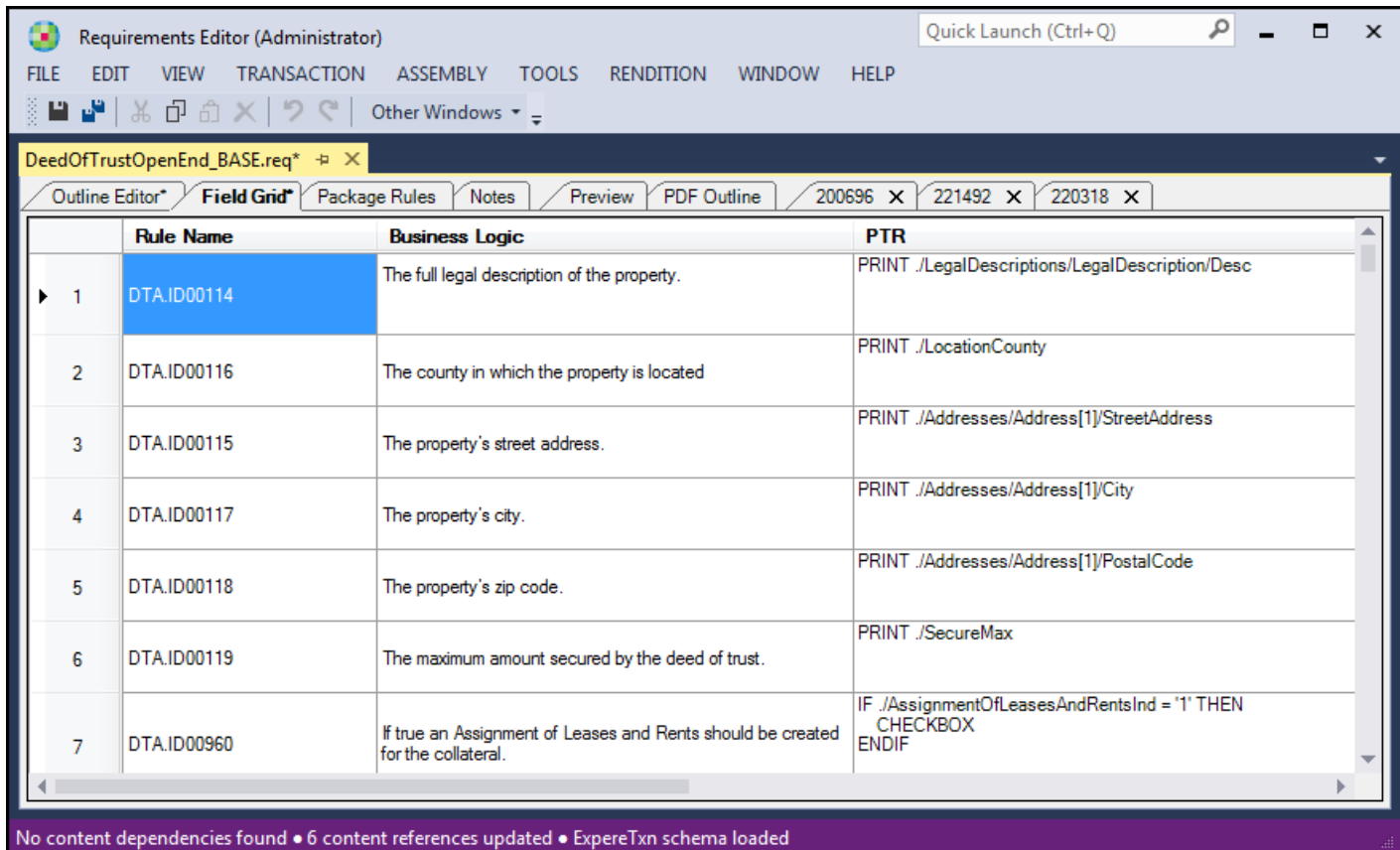
In the **Form View** perspective, you can navigate through the document using the *Previous Page* and *Next Page* buttons. You can also adjust how the document is displayed in the perspective through several options available in the *View* menu.

- **Normal:** Displays the document within the document pane in the normal, or default size. This is slightly smaller than full page width expansion.
- **Page Width:** Displays the document with the full width expanded.
- **Page Height:** Displays the document with the full height of the document showing (from the top to bottom).
- **Full Page:** Displays the entire document within the document pane

## Field Grid

The **Field Grid** perspective allows the author to scroll through all data points (DTAs) and identify any invalid rules or missing GDD names/Public names. This is helpful after the PTRs are complete to be able to quickly scroll through all of the PTRs and identify possible validation issues. Potentially invalid rules will be highlighted in yellow, but note that some rules that will not validate in the field grid are actually valid rules. The Outline Editor perspective is much more accurate for validating PTRs.

The perspective is available by clicking the **Field Grid** tab in the document pane. The Field Grid is available for both dynamic and static documents. This is an example of the field grid perspective in a dynamic document.



	Rule Name	Business Logic	PTR
▶ 1	DTA.ID00114	The full legal description of the property.	PRINT ./LegalDescriptions/LegalDescription/Desc
2	DTA.ID00116	The county in which the property is located	PRINT ./LocationCounty
3	DTA.ID00115	The property's street address.	PRINT ./Addresses/Address[1]/StreetAddress
4	DTA.ID00117	The property's city.	PRINT ./Addresses/Address[1]/City
5	DTA.ID00118	The property's zip code.	PRINT ./Addresses/Address[1]/PostalCode
6	DTA.ID00119	The maximum amount secured by the deed of trust.	PRINT ./SecureMax
7	DTA.ID00960	If true an Assignment of Leases and Rents should be created for the collateral.	IF ./AssignmentOfLeasesAndRentsInd = '1' THEN CHECKBOX ENDIF

No content dependencies found • 6 content references updated • ExpeTxn schema loaded

The following values are used to define a data field and are presented in the Field Grid:

- **Rule Name:** A unique ID for each data field entered.
- **Business Logic:** A plain English description of the purpose of the data field, or rule. This is not used for EPM content.
- **Product Technical Rule:** The PTR represents the processing instructions, or code, that defines the behavior of the rule. This is not used for EPM content.
- **GDD Name:** The data name used for VMP tech/board and E-Forms content.
- **Public Name:** The name used to define the rule for the BSI tech/board and E-Forms content.

- **Substitution Name:** The data name used when completed with values from the Edit Instance screen.
- **Enterprise Name:** This value is not used in current authoring practices.

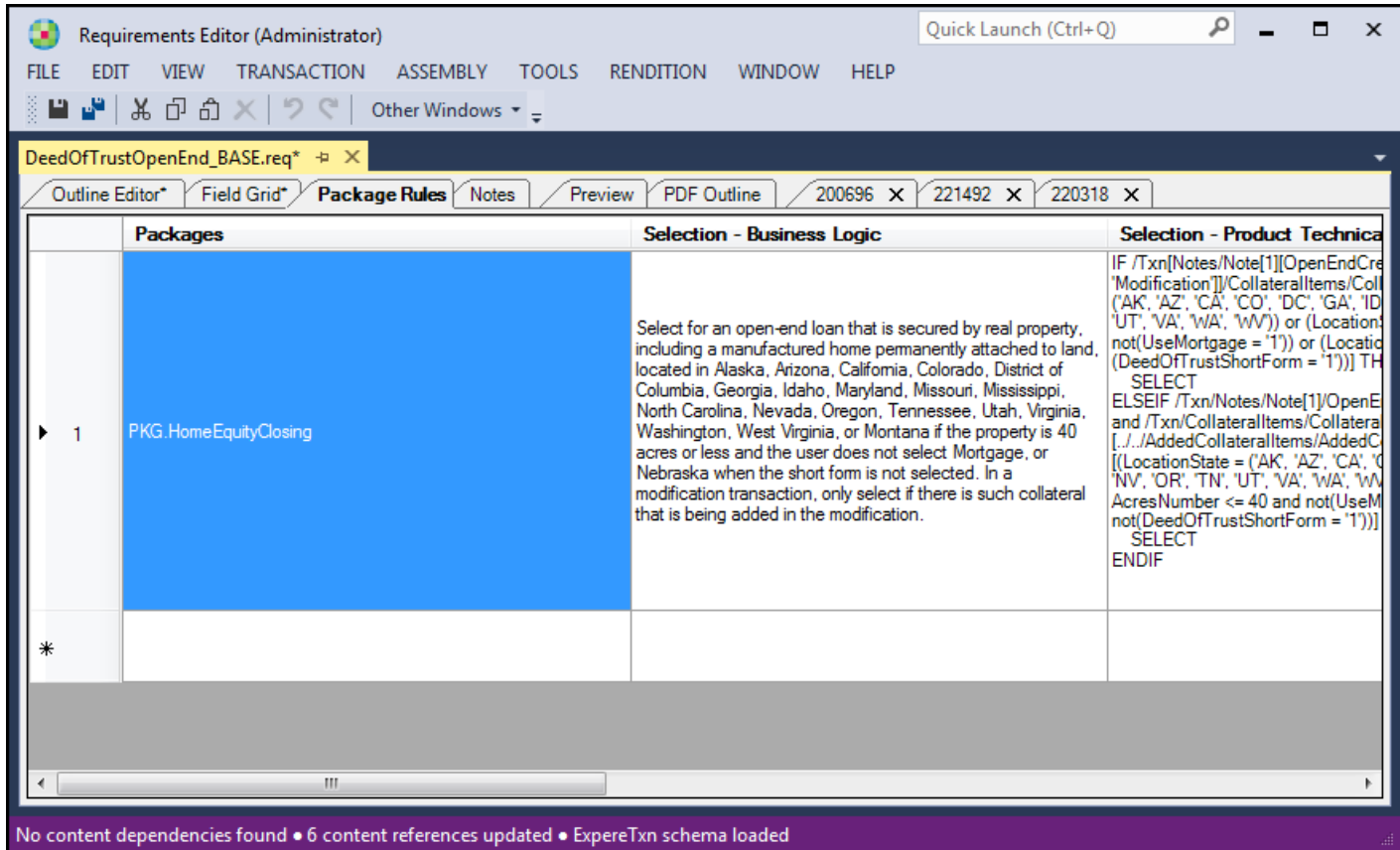
## Package Rules

The **Package Rules** perspective allows the author to define rules to apply to an associated package (also referred to as Auto-Selection and Auto-Suggestion rules) that determine when the document should be selected or suggested based on the transaction data. In this perspective, an author can create new package rules or modify existing ones. A document can have Auto-Selection and Auto-Suggestion rules for each phase that it is in. A separate PTR is written for each selection and suggestion rule.

Package Rule PTRs always start with “IF”, include “THEN” and “SELECT” and end with “ENDIF”. The rules within the Package Rule follow the same Keyword conventions as Condition and Data PTRs. There can be multiple IFs in a Package Rule. For example, the following rule selects the document any time either of the two conditions were met. :

```
IF /Txn/Deposit/IRAPPlans/IRAPlan[Type = '1'] THEN
SELECT
ELSEIF /Txn/Deposit/IRAPPlans/IRAPlan[SelfDirectedInd = '1'] THEN
SELECT
ENDIF
```

The perspective is available by clicking the **Package Rules** tab in the document pane. The Package Rules perspective is available for both dynamic and static documents. This is an example of the field grid perspective in a dynamic document.



## Notes

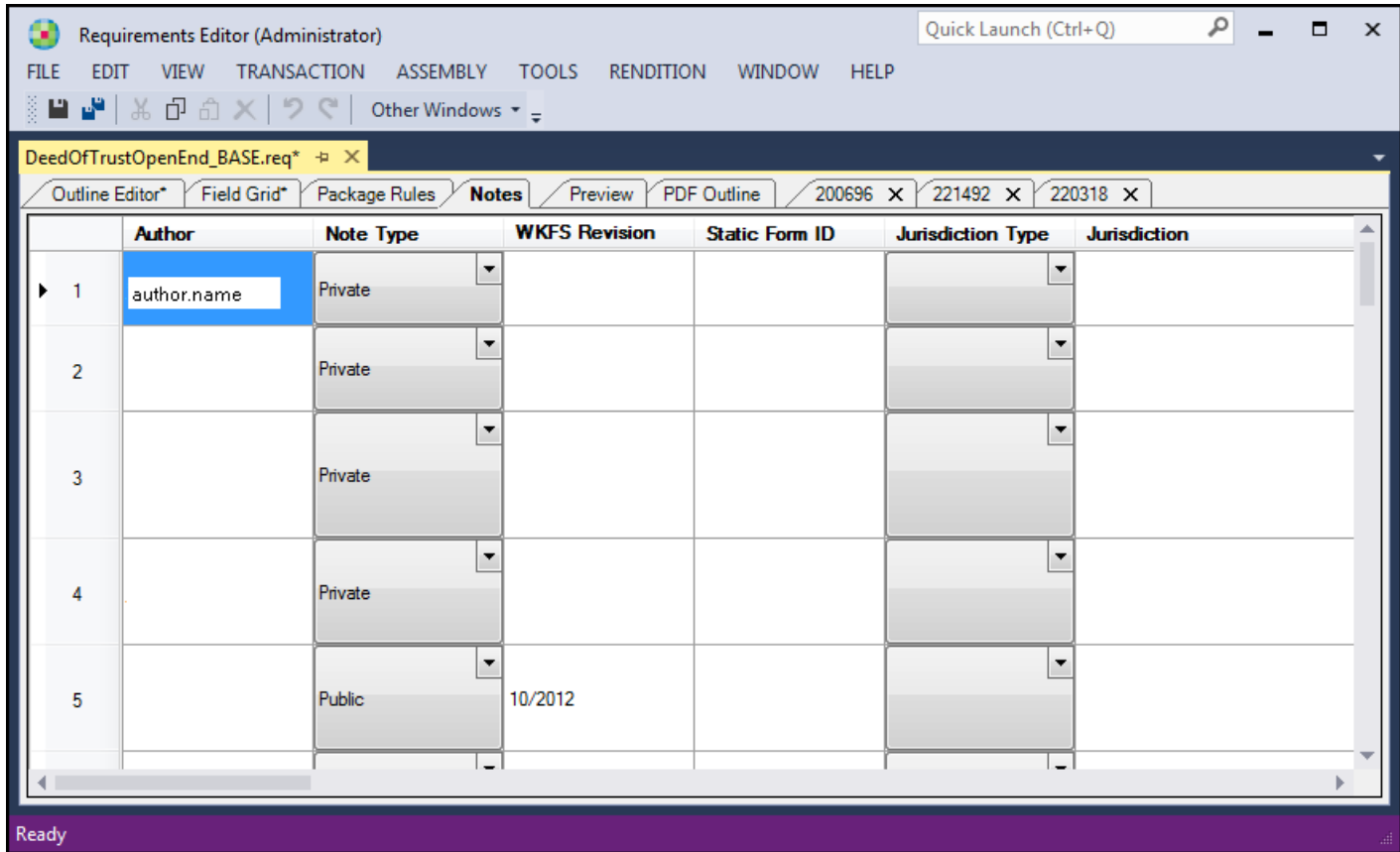
The **Notes** perspective allows the author to document activities in the requirements file for both internal-only and external (public) view. The perspective presents a grid of pre-defined values that can be entered by the author to provide reference, guidance, and summary information about authoring activity and changes made to a requirements file. Notes provided in the **Notes** perspective are not converted into content; however, they may be useful to those reviewing requirements files.

**Note:**

The **Note** element can be used to provide documentation at the element level.

Notes are not converted into content; however, they may be useful to those reviewing requirements files. Only Citation notes are viewable by the end customer. When authoring a new .REQ file, typically only Citation Notes and Private Notes are used. Citation notes include the cite dictating the content or format. Private Notes can be used to document notes about a discussion that resulted in the content appearing a certain way, or a reminder for the author or reviewer to handle the content a specific way.

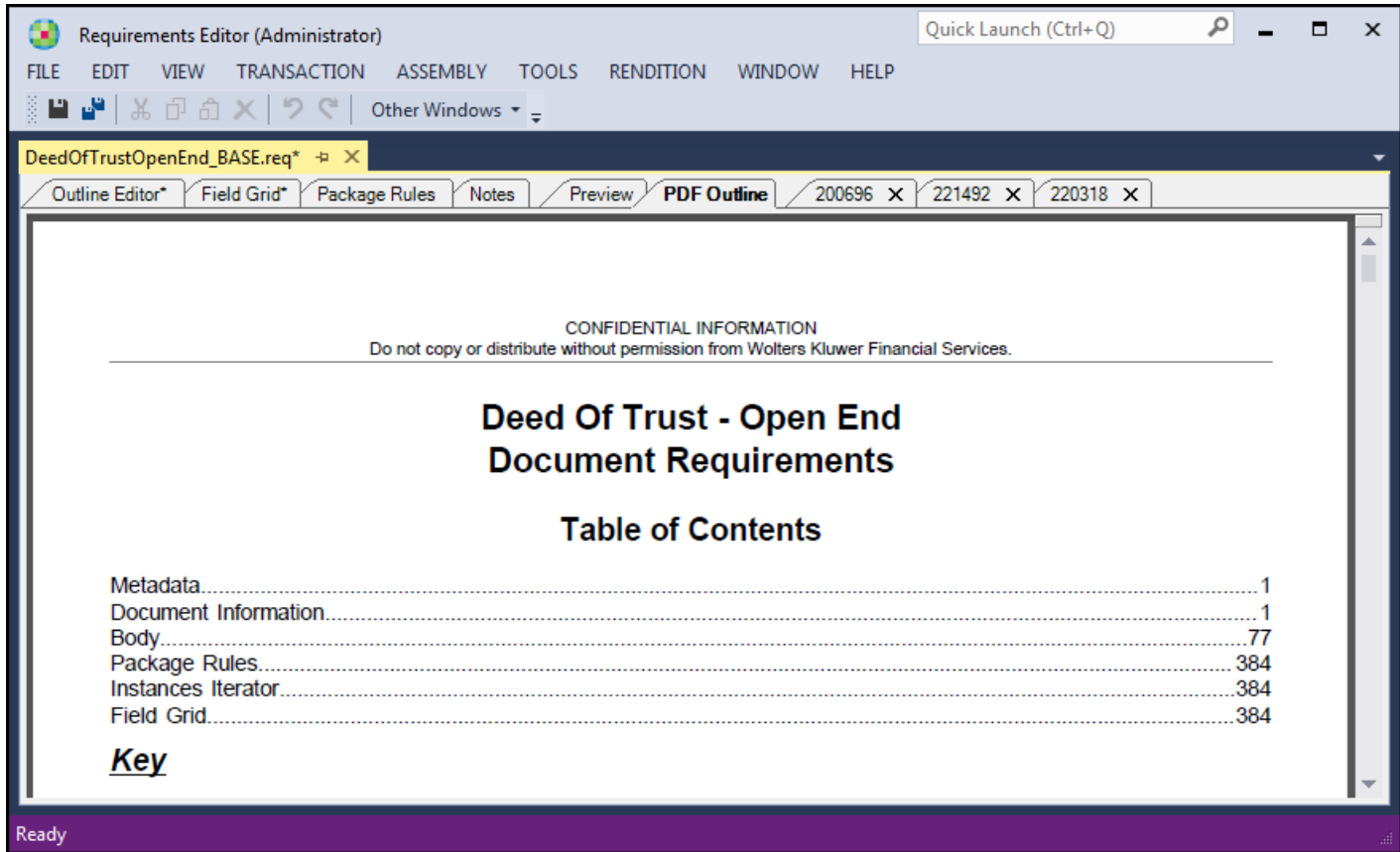
The perspective is available by clicking the **Notes** tab in the document pane. The Notes perspective is available for both dynamic and static documents. This is an example of the notes perspective in a dynamic document.



## PDF Outline

The **PDF Outline** perspective allows the author to create a PDF version of the requirements file as it appears in the **Outline Editor**. Included in the PDF outline are all metadata, document information, body sections, package rules, instances, and the field grid. This perspective is often used in testing a requirements file. The document opens in an integrated Adobe Acrobat Reader as a PDF.

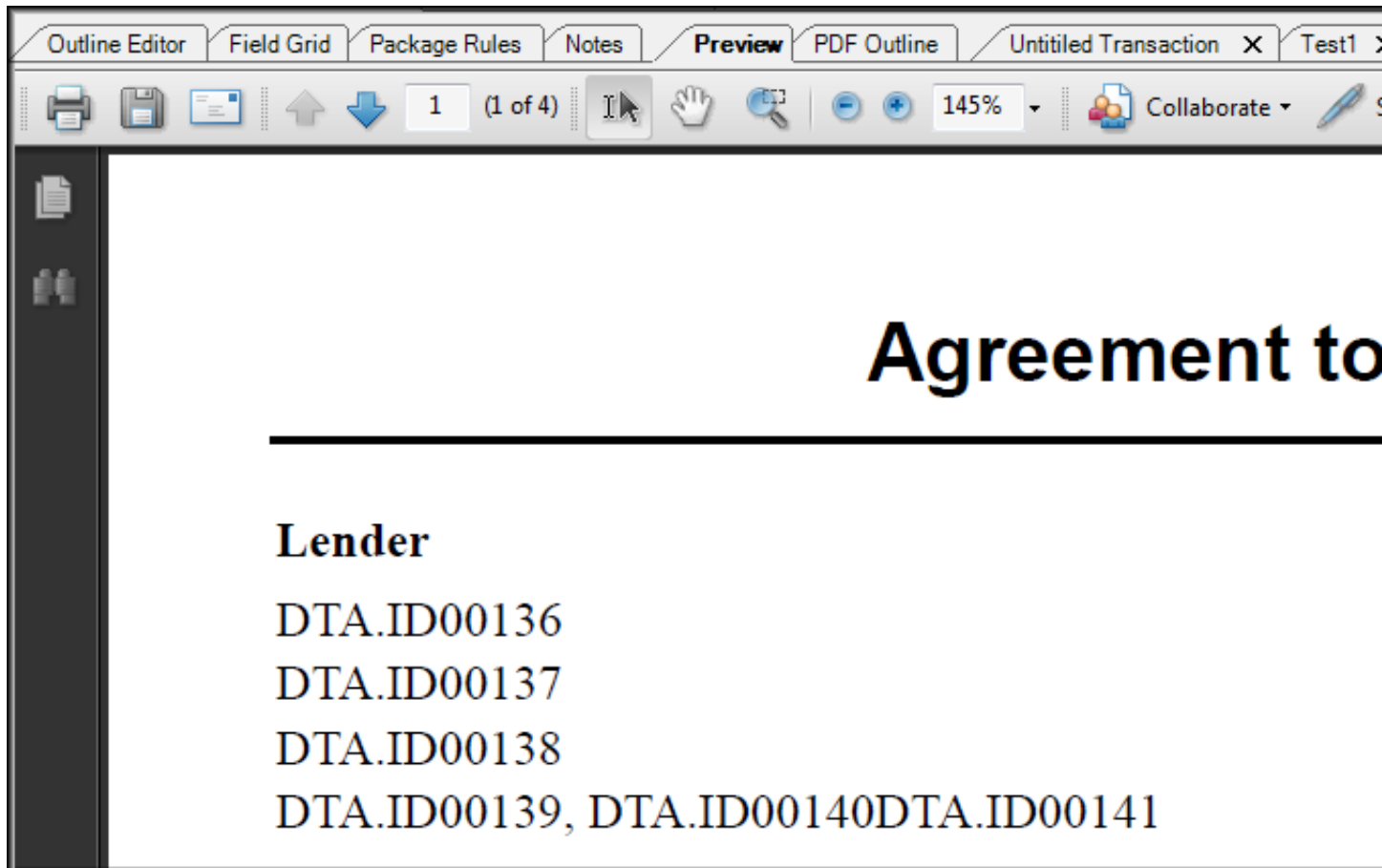
The perspective is available by clicking the **PDF Outline** tab in the document pane. The PDF Outline is available for dynamic documents only. This is an example of the PDF outline perspective in a dynamic document.



## Preview

The **Preview** perspective allows the author to view a requirements files in the format that most closely resembles how the file will appear as a generated PDF and includes a listing of all the data points (DTAs) contained in the file as well as any invalid rules or missing GDD names/Public names.

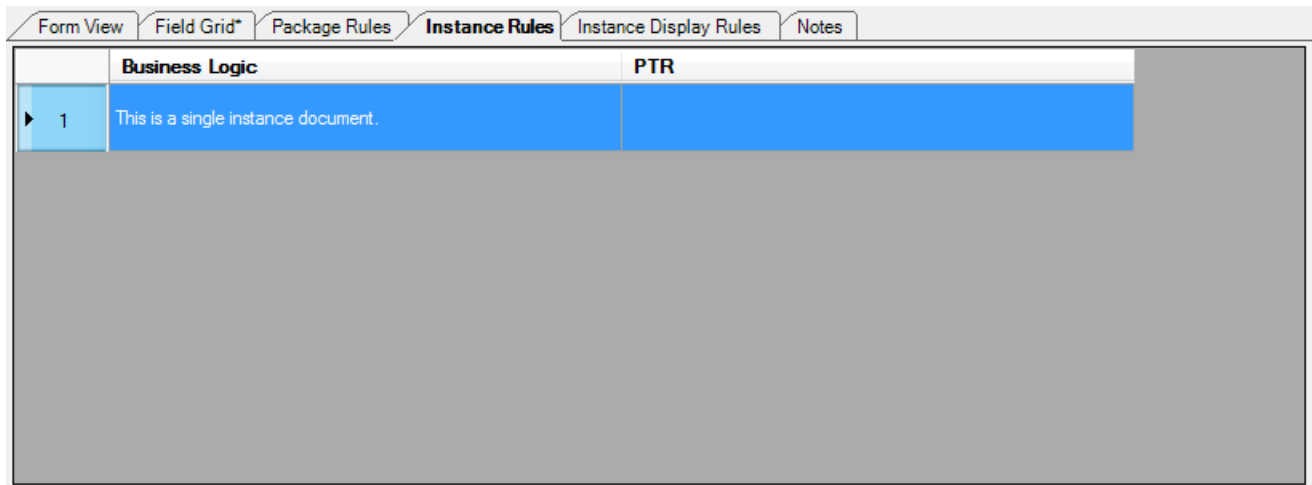
The perspective is available by clicking the **Preview** tab in the document pane. The Preview perspective is available for dynamic documents only. This is an example of the preview perspective in a dynamic document.



### Instance Rules Tab

The **Instance Rules** perspective allows the author to create unique instances of the document for a transaction. For example, a unique instance might be used to take a piece of data and apply it for different scenarios in the same form. When certain options are fulfilled, specific documents are selected for the transaction based on the defined instance rules.

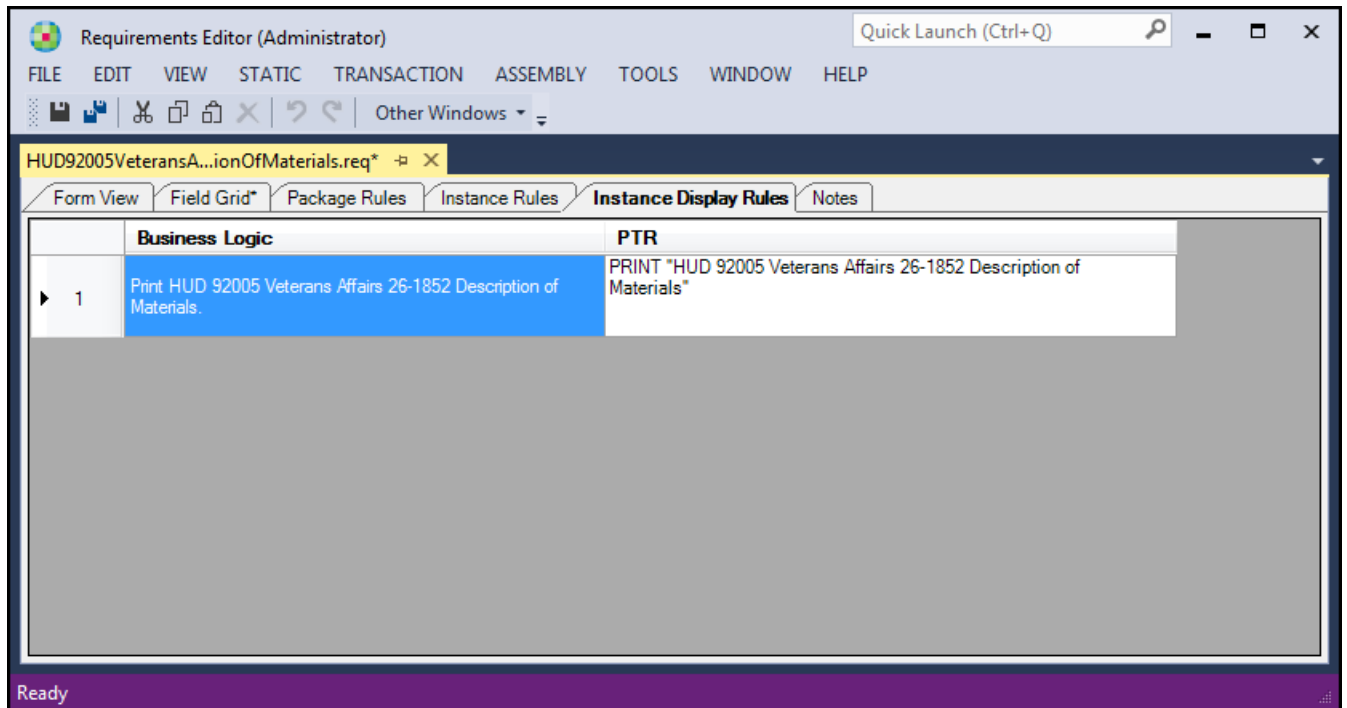
The perspective is available by clicking the **Instance Rules** tab in the document pane. The Instance Rules perspective is available exclusively for static documents and is displayed when a static requirements file is opened. This example is from a single instance document.



## Instance Display Rules (Static Documents)

The **Instance Display Rules** perspective enables the author to identify cases when the form includes more than one variation (as distinguished from a copy of a form). Each static requirements file has an Instance Display Name and that name appears in a list of documents generated by a transaction. If the document is a multiple instance document, the instance display name is followed by a data item that determines which type of identifying information should print after the instance display name. The additional identifying information is typically an account number, borrower name, or some other piece of data that will distinguish one instance of the document from another.

The perspective is available by clicking the **Instance Display Rules** tab in the document pane. The Instance Display Rules perspective is available exclusively for static documents and is displayed when a static requirements file is opened.



## Arranging and Docking Windows

The Requirements Editor environment allows for viewing multiple simultaneous document windows. You can adjust the viewing and editing space for code, depending on how you arrange the windows in the environment. The following options are just some of the ways that windows can be arranged:

- Pin document windows to the left of the tab well.
- Tab-dock windows to the editing frame.
- Float windows over or outside the environment.
- Hide windows along the edge of the environment.
- Display windows on different monitors.
- Reset window placement to the default layout.

Document windows can be arranged by dragging, by using commands on the Window menu, and by right-clicking the title bar of the window to be arranged. Toolbars can be arranged by dragging, or by using the Customize dialog box.

### Floating Windows

Any document window can be undocked from the IDE and moved anywhere on the desktop.

**To position a tool window or document window outside the IDE:** Drag the window to the location you want.

To return a tool window or document window to its most recent docked location: Press CTRL while you double-click the title bar of the window.

**Note:**

Double-clicking the title bar of a window outside the environment without pressing CTRL maximizes the window.

## Docking Windows

By default, document windows are arranged as floating windows in the environment. If you have customized your viewing configuration and select **Window > Reset Window Layout** the windows will be reconfigured in this manner.

Tool windows can be fastened to one side of a frame in the IDE. A guide diamond appears when you drag a tool window to another location in the IDE. The guide diamond helps you re-dock the window on one of the four sides of the IDE, or to a location in the editing frame.

**Note:**

To move a dockable window without snapping it into place, press CTRL while you drag it.

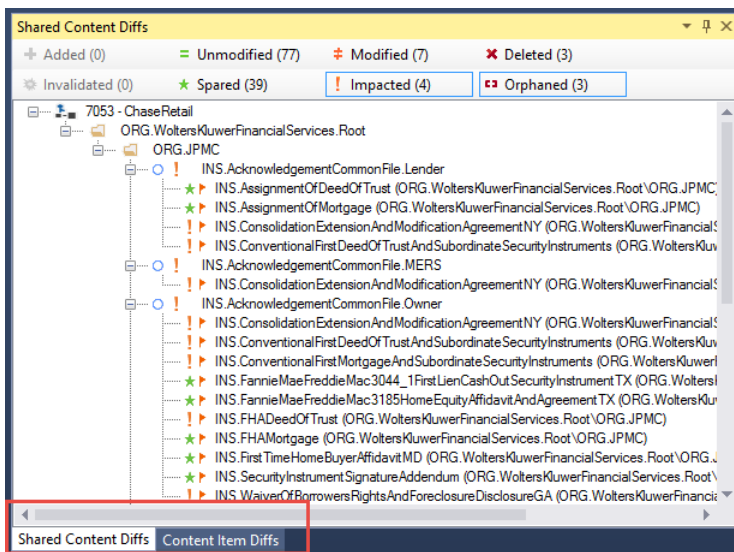
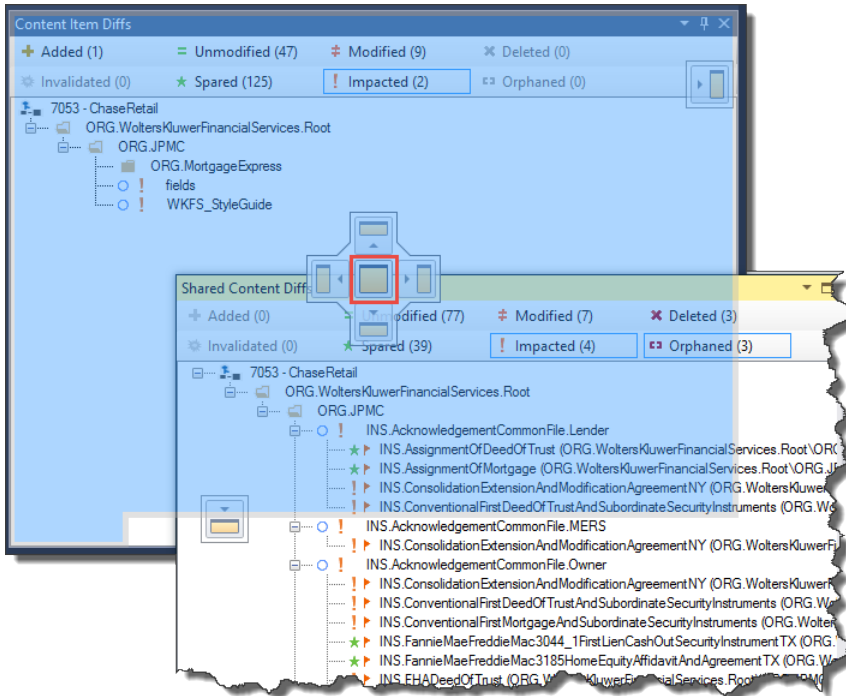
The following example shows the guide diamonds that appear when you drag a tool window or document window toward the center of the environment. The diamond on the right edge only appears when you drag a tool window toward the edge of the environment.



### To dock tool windows and document windows:

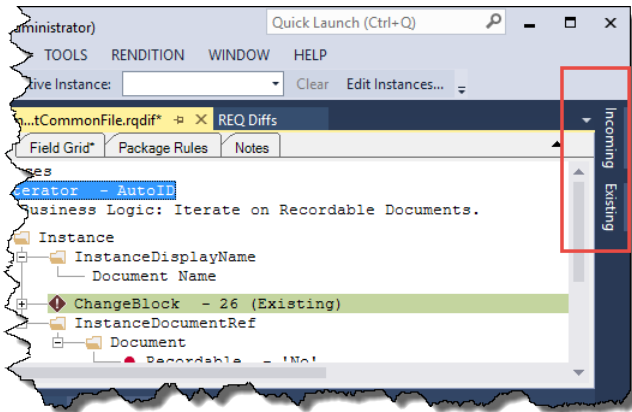
1. Click the document window you want to dock.
2. Drag the window toward the middle of the environment.  
A guide diamond appears. The four arrows of the diamond point toward the four sides of the editing pane.
3. When the window you are dragging reaches the location where you want to dock it, move the pointer over the corresponding portion of the guide diamond.  
The designated area is shaded.
4. To dock the window in the position indicated, release the mouse button.

For example, if Merge Session is docked on the right edge of the environment and you want to dock it on the left edge, drag Merge Session toward the middle of the environment, move the pointer over the far left arrow of the guide diamond, and then release the mouse button. Alternatively, you can dock a tool window to a portion of one of the side walls of the environment by dragging it to the side until you see a secondary guide diamond. Click one of the four arrows to dock the tool window to that portion of the side wall. If you drag and release a window to the center of the guide diamond, the window will join the window in that location and the window titles appear as tabs.




## Hiding/Minimizing Windows

Document windows, the names of which appear on the **View** menu, support a feature named Auto Hide. Auto Hide causes a window to slide out of the way when you use a different window. When a window is auto-hidden, its name and icon are displayed on a tab at the edge of the environment (see below). To use the window again, move the pointer over the tab so that it slides back into view.



### To turn on Auto Hide

1. Click the window you want to hide.
2. On the **Window** menu, click **Auto Hide**.

Alternatively, you can click the pushpin icon (  ) on the title bar of the window.

#### Note:

To turn off Auto Hide, click the window you want to keep visible. On the **Window** menu, click **Auto Hide** to clear the check mark. Alternatively, you can click the pushpin icon on the title bar of the window.

## Resetting Window Layouts

You can return the IDE to the original window layout for your settings collection by using the Reset Window Layout command. When you run this command, the following actions occur:

- All windows are moved to their default positions.
- Windows that are closed in the default window layout are closed.
- Windows that are open in the default window layout are opened.

You can specify a different default window layout by modifying your current settings.

### To reset window placement to the default layout

1. On the **Windows** menu, click **Reset Window Layout**.
2. In the message box, click **Yes**.

## View Windows

The View Windows (Windows) display specific information about the active document in the document pane. You can open the view windows through the *View* menu by selecting *Other Windows* or from the standard toolbar in the *Other Windows* list.

The following view windows are available when a requirements file is opened in the editor.

- Attributes
- Cell Contents (Static Documents)
- Dynamic Help
- Active Conditions
- PTR Editor
- Precedence Status
- Build Errors
- Content Errors
- Messages

### Attributes

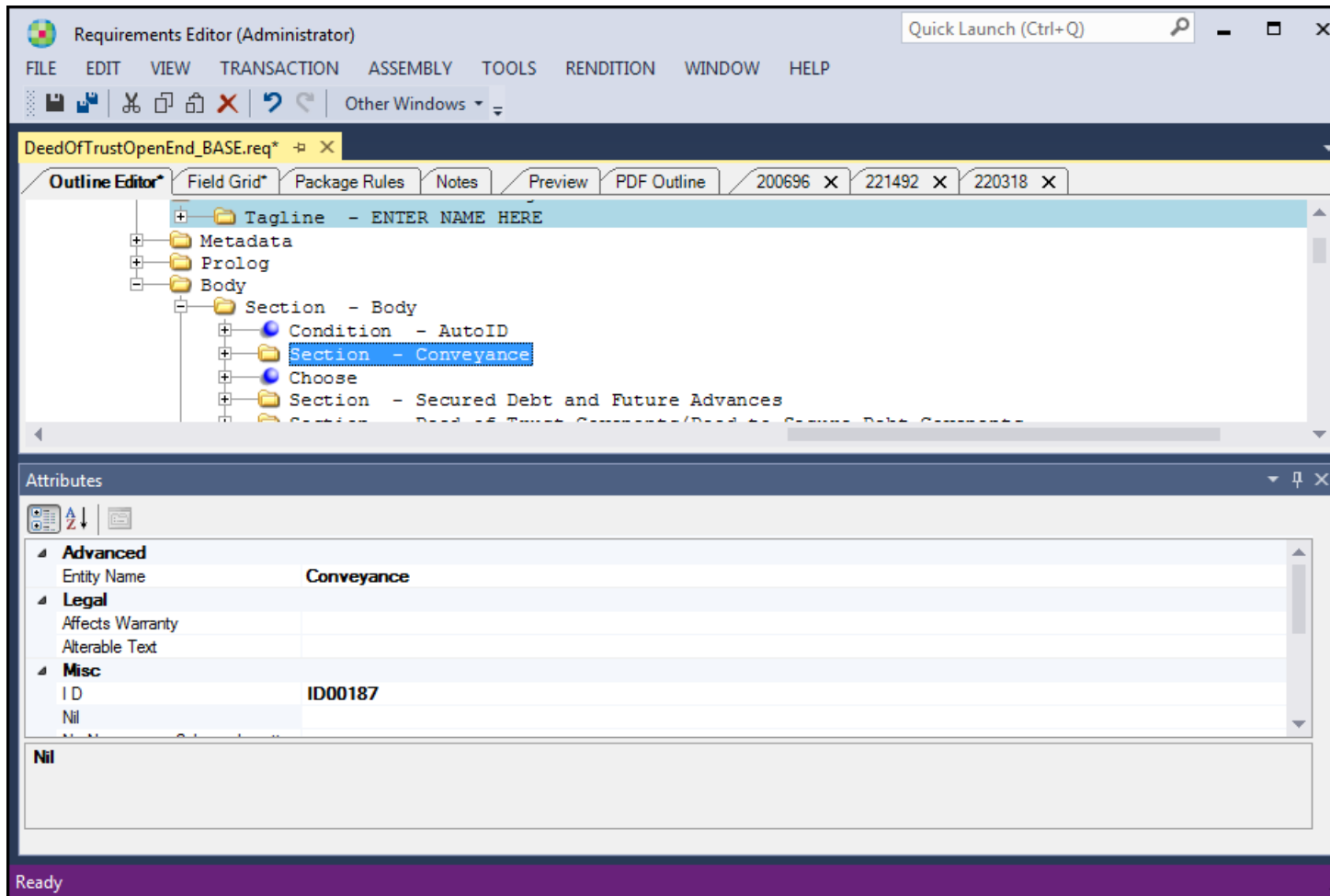
The **Attributes** view window displays advanced attribute properties for an element contained in the requirements file.

The view window is available by clicking the **Attributes** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic documents; for static documents, the view is available but the functionality is not available.

**Note:**

The *Attributes* view window is typically used when authoring in the Document Editor perspective. The Document Editor perspective is available as an advanced feature option during installation of the Requirements Editor.

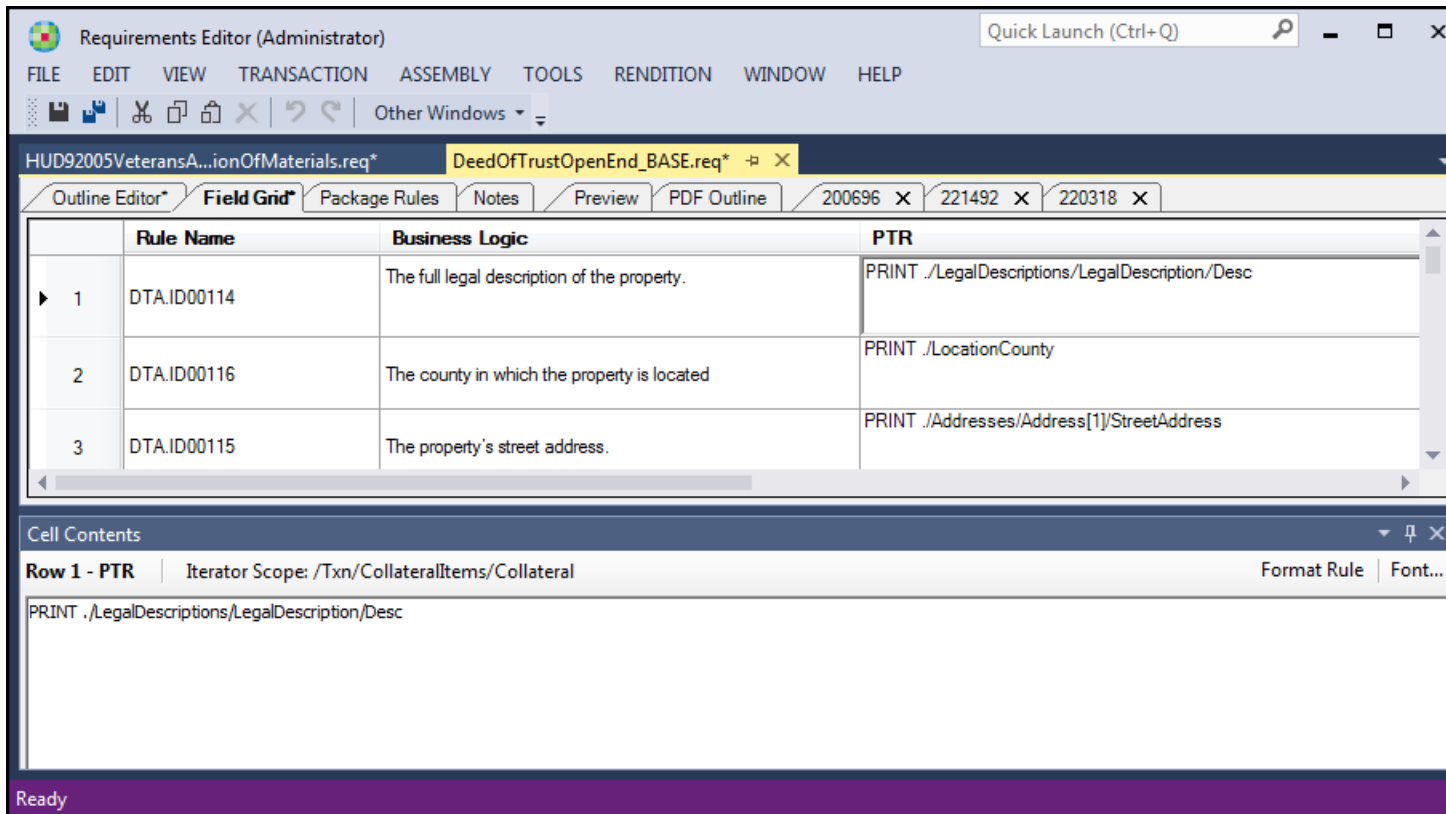
This is an example of the attributes window when a *Section* element is selected (*Section - Conveyance*).



## Cell Contents

The **Cell Contents** window provides an editing view of the contents of a particular cell, or field, that is active in an authoring perspective. This feature allows authors to edit the cell contents in an editing view and offers some formatting options for the author to apply exclusively to the cell contents. This view provides an option for author to edit outside of the cell in a given authoring perspective.

The view window is available by clicking the **Cell Contents** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic and static documents in any of the authoring perspectives except *Preview* and *PDF Outline*. This is an example of the cell contents window when the *Field Grid* perspective for a dynamic document is active and the PTR field for a data point is selected.



Requirements Editor (Administrator) Quick Launch (Ctrl+Q)

FILE EDIT VIEW TRANSACTION ASSEMBLY TOOLS RENDITION WINDOW HELP

HUD92005VeteransA...ionOfMaterials.req\* DeedOfTrustOpenEnd\_BASE.req\* - X

Outline Editor\* **Field Grid\*** Package Rules Notes Preview PDF Outline 200696 X 221492 X 220318 X

	Rule Name	Business Logic	PTR
▶ 1	DTA.ID00114	The full legal description of the property.	PRINT ./LegalDescriptions/LegalDescription/Desc
2	DTA.ID00116	The county in which the property is located	PRINT ./LocationCounty
3	DTA.ID00115	The property's street address.	PRINT ./Addresses/Address[1]/StreetAddress

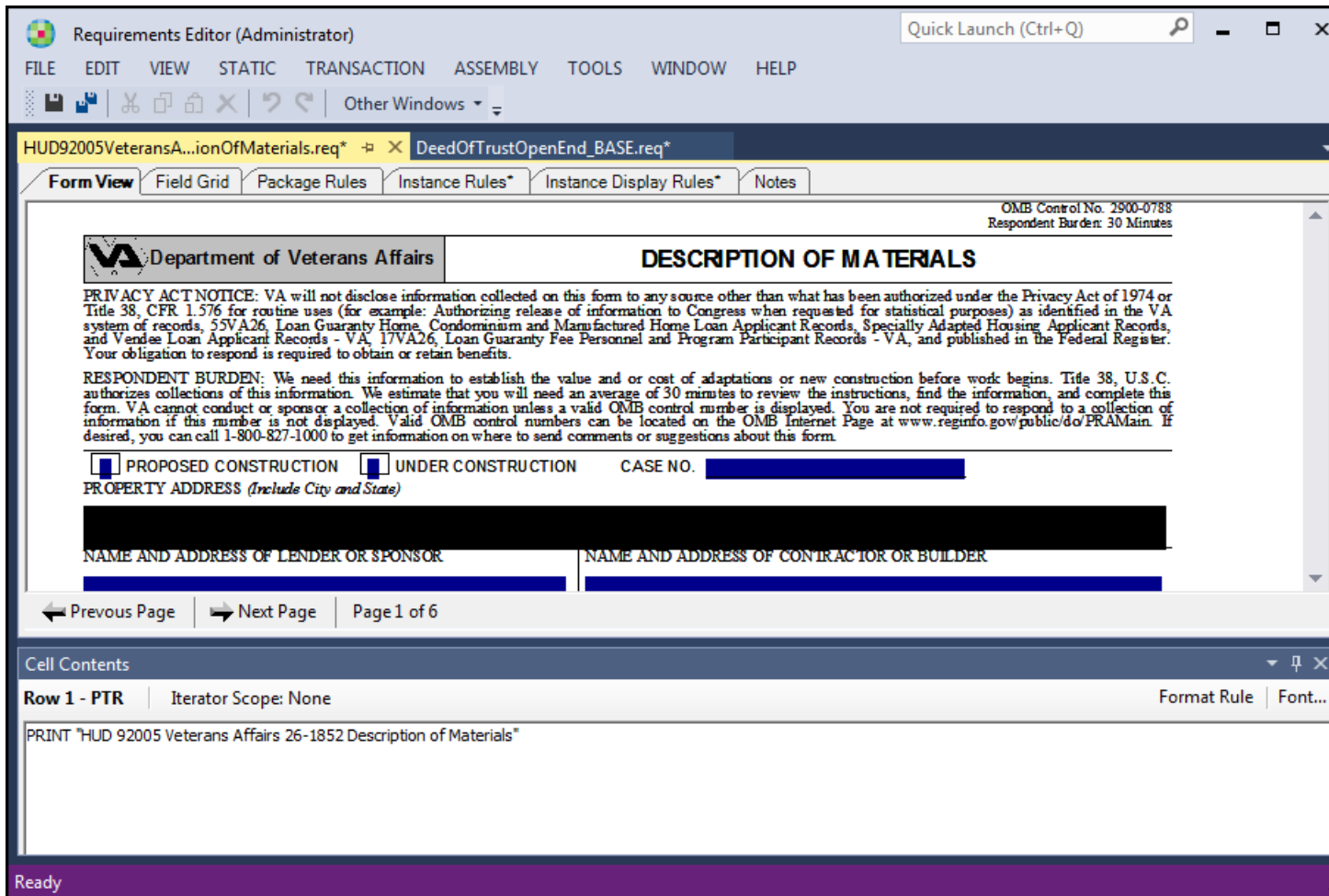
Cell Contents

Row 1 - PTR | Iterator Scope: /Txn/CollateralItems/Collateral | Format Rule | Font...

PRINT ./LegalDescriptions/LegalDescription/Desc

Ready

Here is an example of the cell contents window when the *Form View* perspective for a static document is active and an editable field (the property address field) is selected.



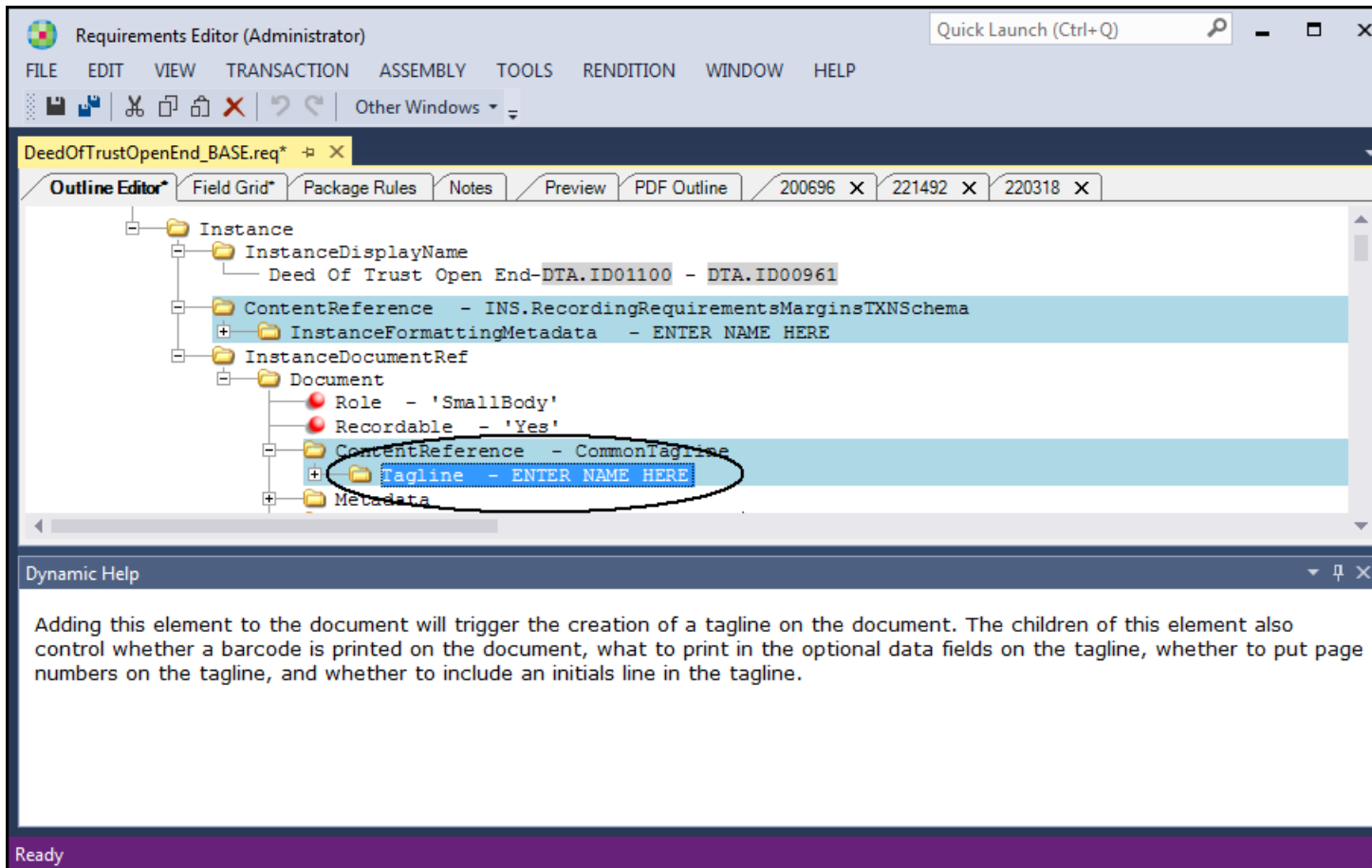
## Dynamic Help

The **Dynamic Help** view window provides basic guidance and best practice tips for a selected element within the requirements file. The help is derived from annotations in the source code and displays for a given element when that element is selected in the Outline Editor. In addition to the Dynamic Help view, the help copy also is displayed as a tool tip when you hover over an element in the Outline Editor.

**Note:**

Dynamic Help is not implemented in all possible situations; when not implemented, default help copy is displayed.

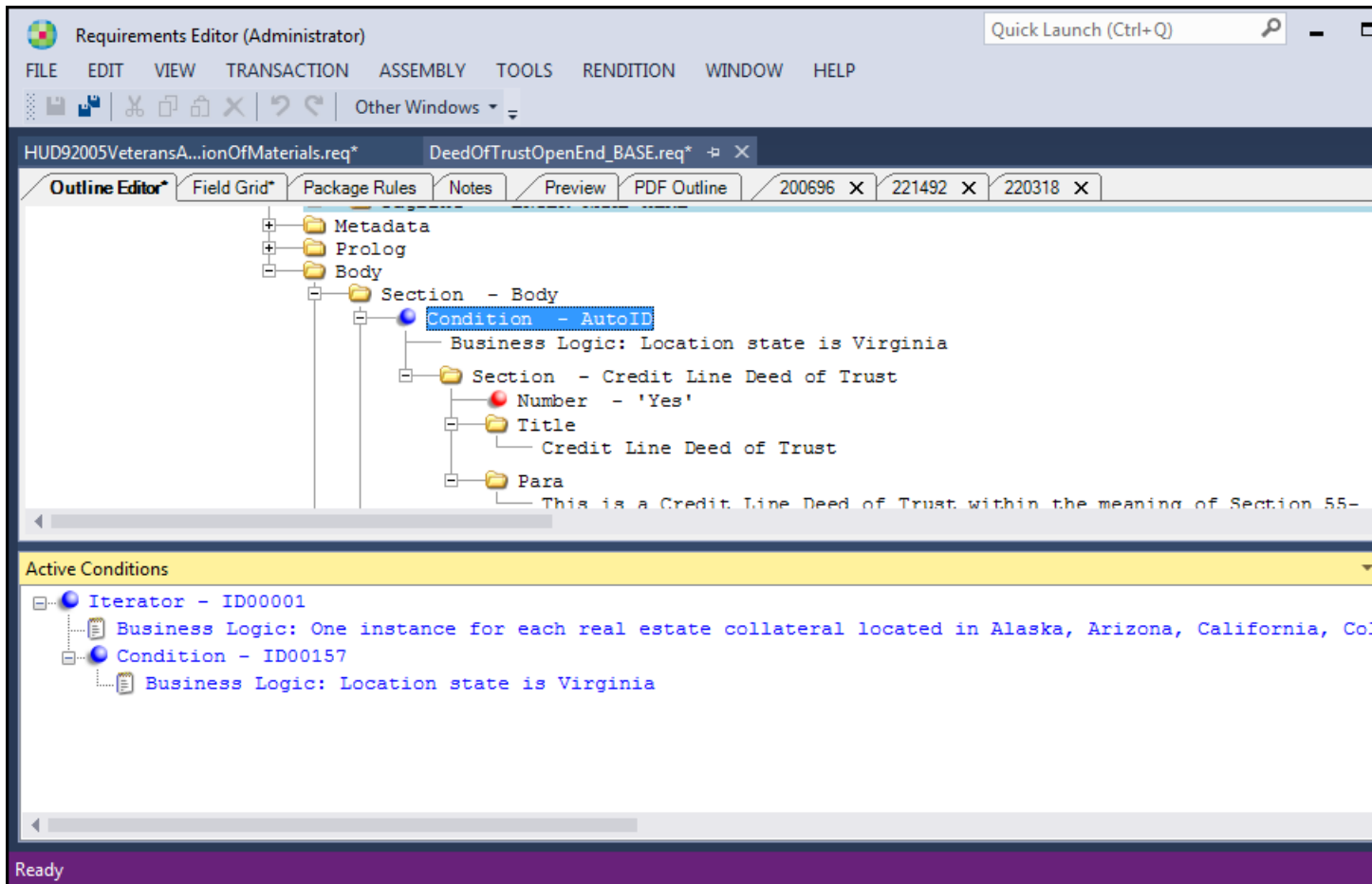
The view window is available by clicking the **Dynamic Help** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic documents; for static documents, the view is available but help text is not implemented. This is an example of the dynamic help window when the *Tagline* element is active.



## Active Conditions

The **Active Conditions** view window displays any rules and subsequent business logic that apply to the selected element node. The view is a read-only presentation of the rules selected in the Outline Editor perspective for a dynamic document.

The view window is available by clicking the **Active Conditions** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic documents; for static documents, the view is available but static documents contain no active conditions. This is an example of the active conditions window when a *Condition* element is selected in the Outline Editor.



## PTR Editor

The **PTR Editor** view windows provides an editing view for product technical rules (PTRs) contained within a dynamic or static requirements files. Authors may edit business logic and technical rules as well as navigate to specific locations in the files where the condition is used. The view also presents options to view and edit the output format, GDD name, and Public name along with changing the formatting of the PTR within the Outline Editor.

**Note:**

If the rule comes from a content reference, then **Output Format, GDD Name** and **Public Name**, as well as the PTR and business logic, are read only.

The view window is available by clicking the **PTR Editor** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic and static requirements files and is available from the Outline Editor perspective when a field is selected. This is an example of the PTR Editor when a *Condition* is active in a dynamic requirements file.

## PTR Editor - Loan Estimate

**COND.ID00019**

Locations(1) ▾

Output Format: ▾

GDD Name: ▾

Public

Business Logic: If the total number of origination charges exceeds 12.

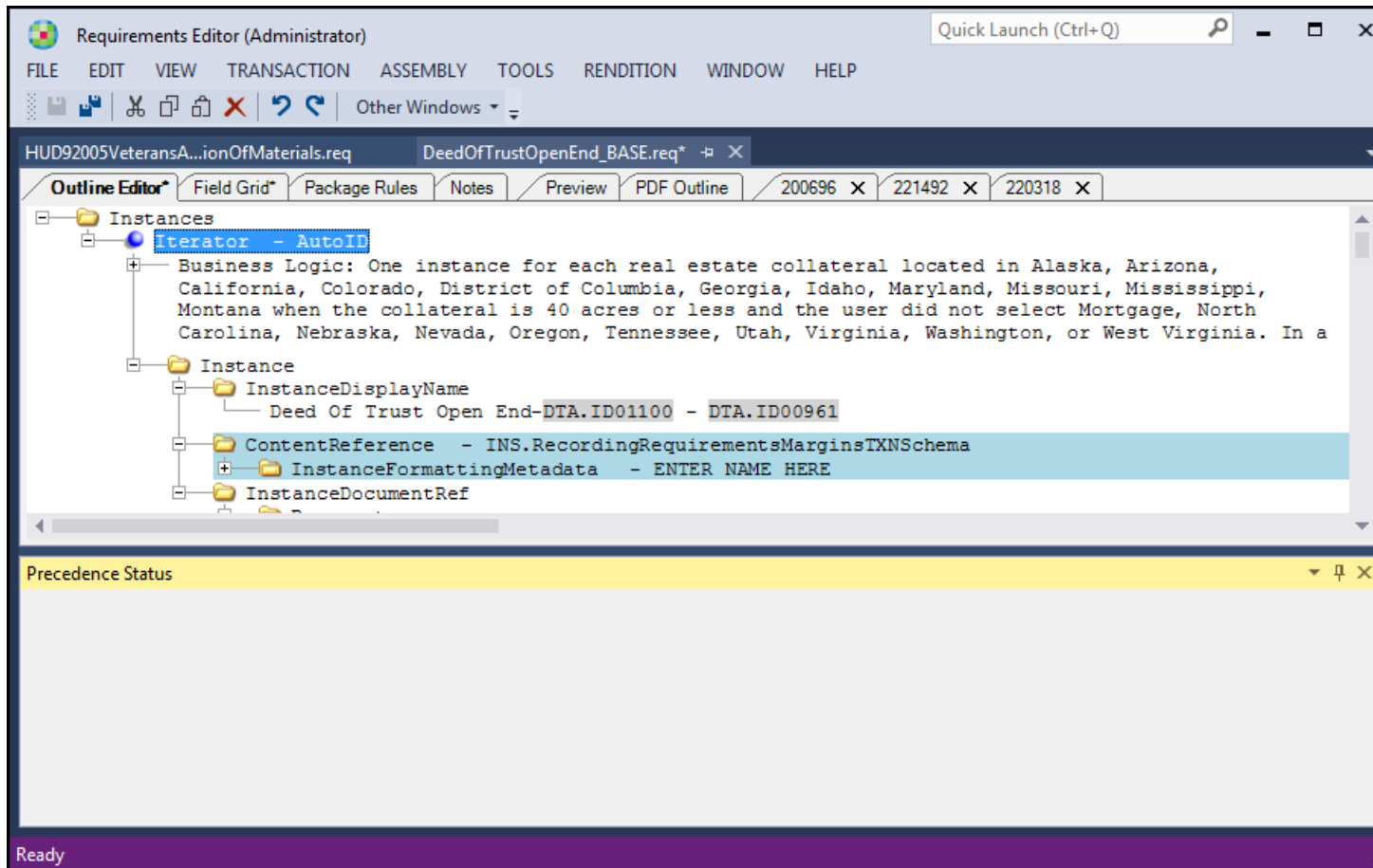
PTR: Iterator Scope: None

```
IF count(/Txn/Notes/Note[1]/FeesAndChargesBaseFees/FeesAndChargesBaseFee[string(FeeName) != "Loan Discount"])[LoanEst] > 12  
  INCLUDE  
ENDIF
```

## Precedence Status

The **Precedence Status** view window displays the precedence status of an element within the requirements files. When working with inherited documents, you can view the original document (the source of the inherited document) in the precedence window for any element that has been modified from its original version.

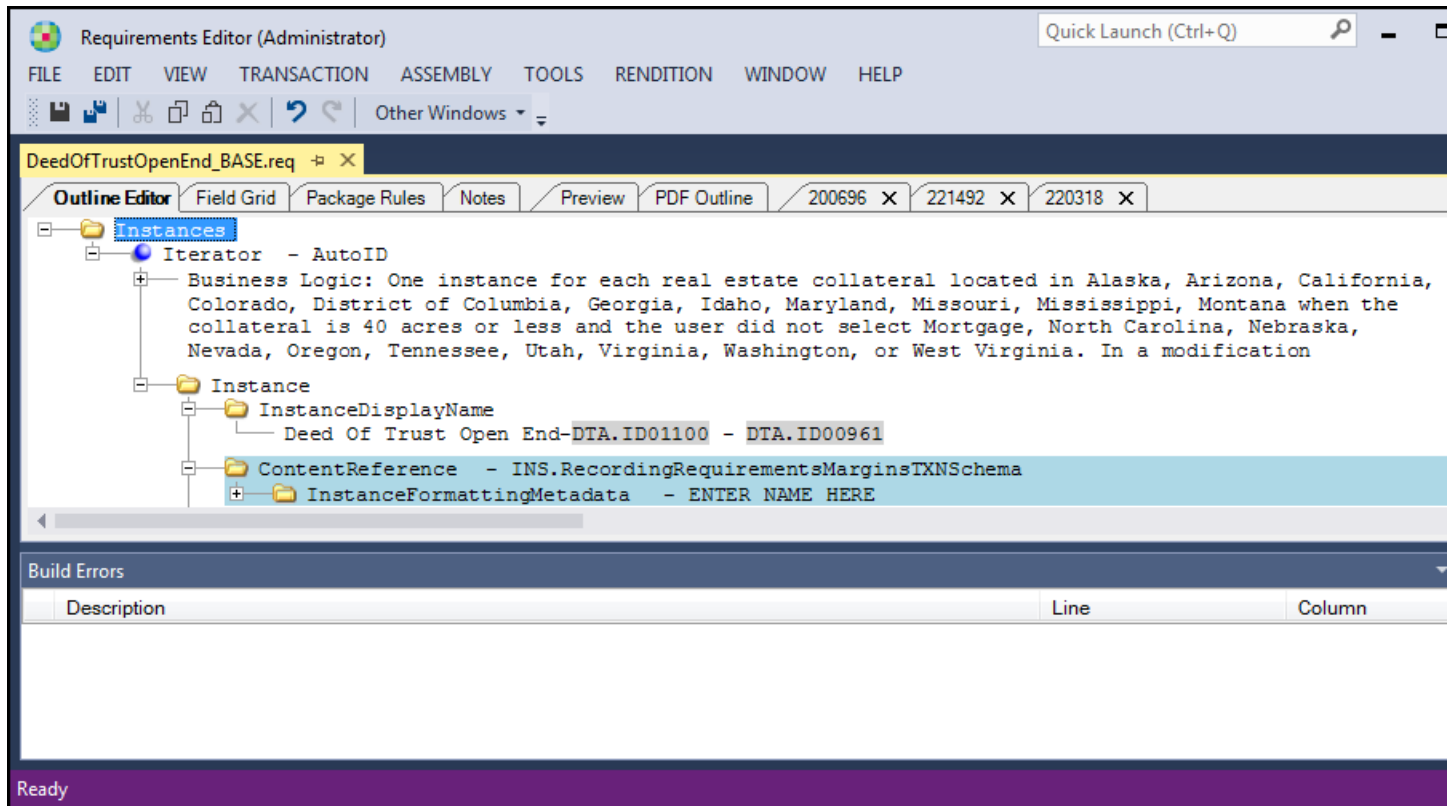
The view window is available by clicking the **Precedence Status** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic and static documents. From this view you can load the original document by right-clicking on a selected element and then selecting **View Original Document** in the shortcut menu.



## Build Errors

The **Build Errors** view window provides a listing of errors returned by the system as a result of the document assembly (build) process. When a document is submitted for assembly using any of the three assembly options (Build, Execute Document, or Execute Package Rules); any system errors are returned in the **Build Errors** or **Content Errors** view. The reported errors include fatal errors that prevented document assembly and warning errors where the document assembled with the noted errors. The **Build Errors** view provides a list of all errors that are preventing a successful build.

The view window is available by clicking the **Build Errors** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic and static documents. This is an example of the build errors view for a dynamic requirements file that was successfully built (and hence no errors were reported).



## Content Errors

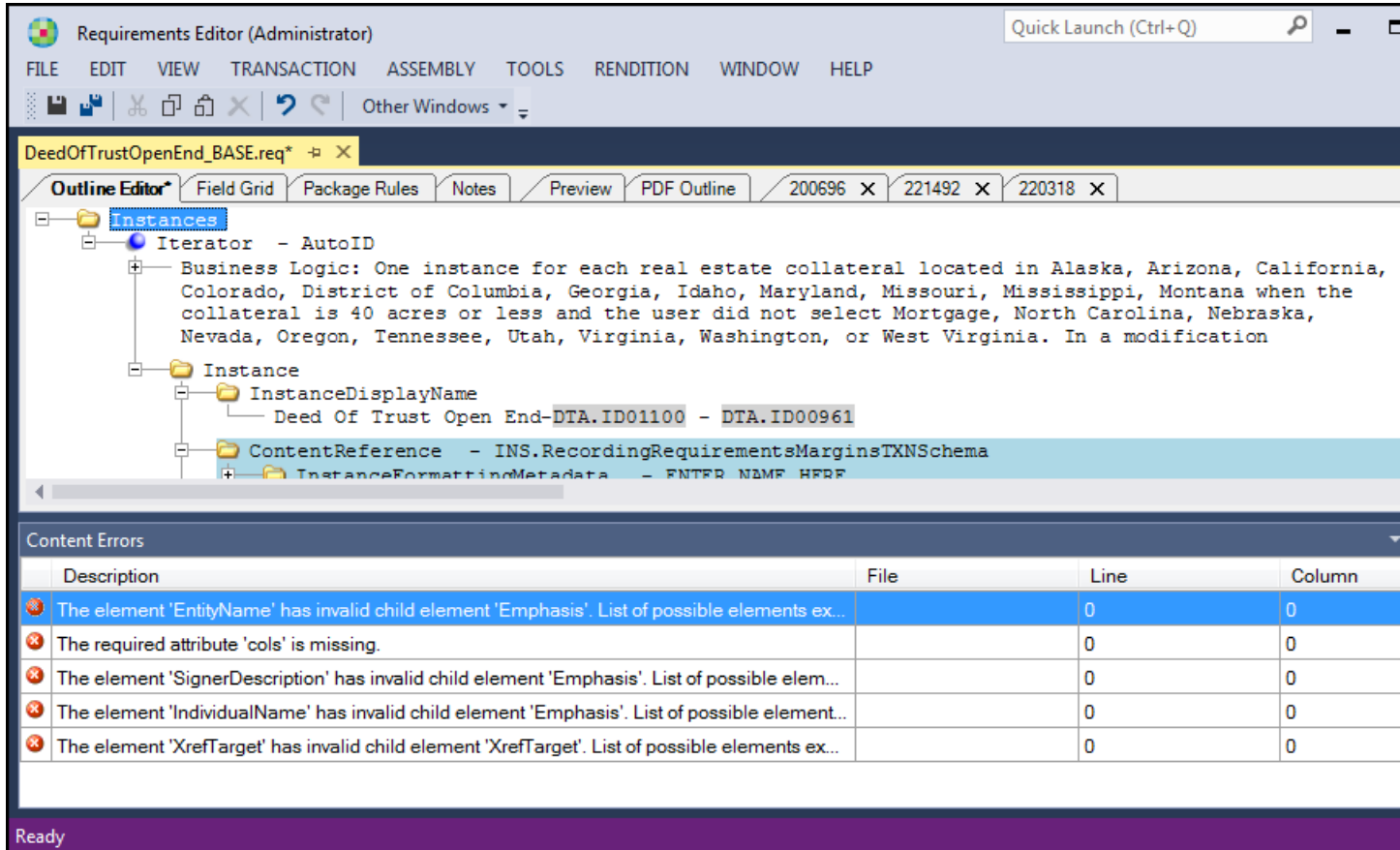
The **Content Errors** view window provides a listing of errors returned by the system as a result of the document assembly processing. When a document is submitted for assembly using any of the three assembly options (Build, Execute Document, or Execute Package Rules); any system errors are returned in the **Content Errors** or **Build Errors** view. The reported errors include fatal errors that prevented document assembly and warning errors where the document assembled with the noted errors.

The **Content Errors** view provides list of each error, an area for analysis, or an areas to resolve after running the build along with the following information:

- **Description:** A summary or description of the error encountered by the system. Double-click on the error description to go to the specific line where the error was encountered.
- **File:** The file, if applicable, where the error was encountered.
- **Line:** The specific line in the file where the error was encountered.
- **Column:** The column in the file where the error was encountered.

Authors should review the error lists to identify the data item or condition causing the error and resolve. These errors are typically caused by things like missing or uneven numbers of [] or () or missing characters in schema paths, but any type of error in a PTR can cause issues so using the error list reports is imperative to being able to execute documents. Before a document can be considered complete, the errors must be fixed from both lists.

The view window is available by clicking the **Content Errors** option in the *View > Other Windows* menu and also through the standard toolbar in the *Other Windows* list. The view is available for dynamic and static documents. This is an example of the content errors view for a dynamic requirements file.



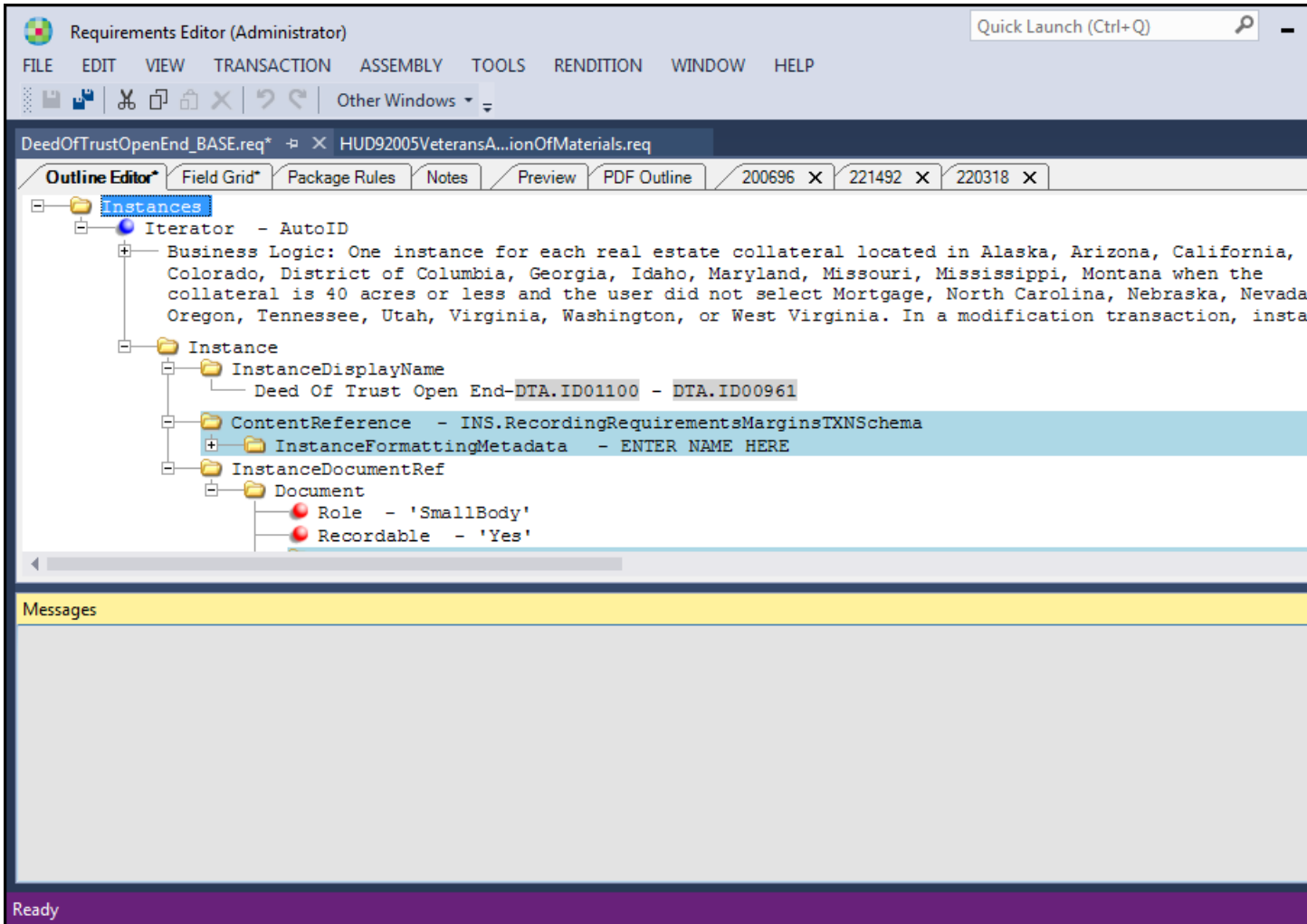
The screenshot shows the Requirements Editor (Administrator) interface. The main window displays a tree view of a requirements file named 'DeedOfTrustOpenEnd\_BASE.req'. The tree structure includes an 'Instances' folder, an 'Iterator - AutoID' node, and an 'Instance' folder. The 'Instance' folder contains 'InstanceDisplayName' (Deed Of Trust Open End-DTA.ID01100 - DTA.ID00961) and 'ContentReference - INS.RecordingRequirementsMarginsTXNSchema'. The 'Content Errors' window is open at the bottom, displaying a table of errors.

Description	File	Line	Column
The element 'EntityName' has invalid child element 'Emphasis'. List of possible elements ex...		0	0
The required attribute 'cols' is missing.		0	0
The element 'SignerDescription' has invalid child element 'Emphasis'. List of possible elem...		0	0
The element 'IndividualName' has invalid child element 'Emphasis'. List of possible element...		0	0
The element 'XrefTarget' has invalid child element 'XrefTarget'. List of possible elements ex...		0	0

## Messages

The **Messages** view window provides a log of system-generated messages related to actions taken on the selected requirement files.

The view window is unique in that it is available directly through the **View** menu and not the *Other Windows* options. In the **View** menu, click **Messages**. The view is available for both dynamic and static documents.



Requirements Editor (Administrator) Quick Launch (Ctrl+Q)

FILE EDIT VIEW TRANSACTION ASSEMBLY TOOLS RENDITION WINDOW HELP

DeedOfTrustOpenEnd\_BASE.req\* HUD92005VeteransA...ionOfMaterials.req

Outline Editor\* Field Grid\* Package Rules Notes Preview PDF Outline 200696 X 221492 X 220318 X

Instances

- Iterator - AutoID
  - Business Logic: One instance for each real estate collateral located in Alaska, Arizona, California, Colorado, District of Columbia, Georgia, Idaho, Maryland, Missouri, Mississippi, Montana when the collateral is 40 acres or less and the user did not select Mortgage, North Carolina, Nebraska, Nevada Oregon, Tennessee, Utah, Virginia, Washington, or West Virginia. In a modification transaction, insta
  - Instance
    - InstanceDisplayName
      - Deed Of Trust Open End-DTA.ID01100 - DTA.ID00961
    - ContentReference - INS.RecordingRequirementsMarginsTXNSchema
      - InstanceFormattingMetadata - ENTER NAME HERE
    - InstanceDocumentRef
      - Document
        - Role - 'SmallBody'
        - Recordable - 'Yes'

Messages

Ready

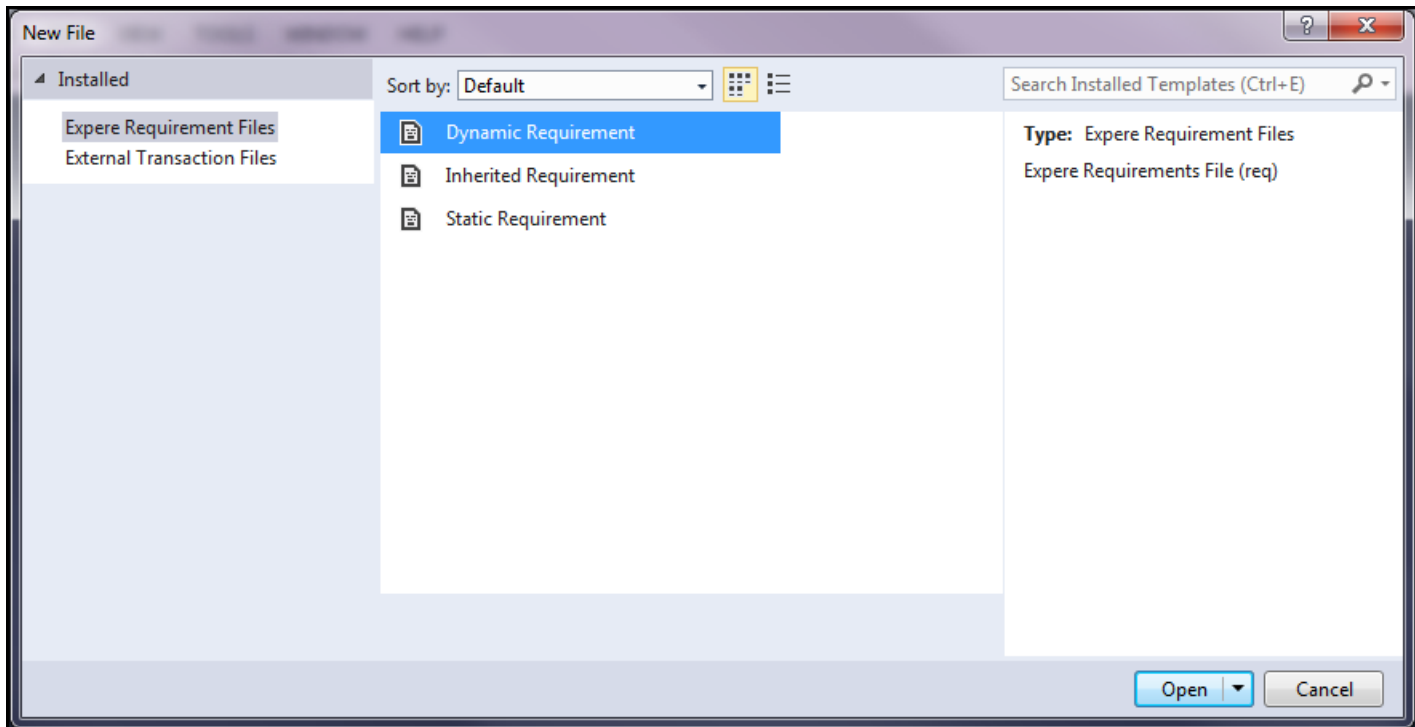
## Creating Requirements Files (REQ Files)

This section explains how to create an Expere Requirements File (REQ). Before you begin, you will want to determine what kind of REQ file to create. There are three kinds of REQ files to choose from:

- Dynamic Requirement
- Inherited Requirement
- Static Requirement

In addition to the kind of requirements file, you will want to also know how the naming convention for the files and some of the basic requirements for the REQ. These are typically defined in a REQ requirements document or some other defining document that is prepared during analysis.

With the requirements for the file known in advance, you can create the REQ file by clicking **New** in the **File** menu. The **New File** dialog box list all the available REQ file templates (typically the three mentioned above but you may have additional templates available to you).



### REQ File Reference

The following information is defined as you create a new REQ file as the **Document Properties**. Document properties must be defined in order for the document to be saved and successfully completed and saved. The majority of the information defined in the document properties is provided in the REQ requirements document.

- **Schema:** Identifies the data schema to be used for the document. The schema options are contained in the drop-down list with the default schema as *Expere TXN*. The Expere TXN schema is the base schema for the majority of documents while the Mortgage line of business typically are modeled on the VMP MISMO schema.

**Note:**

The schema selection does not impact Non-Runtime content which uses Public Names and GDD names (you can use the Expere TXN default option).

- **Precedence Unit:** By default, this value populates with as ORG.WoltersKluwerFinancialServices.Root representing the base precedence structure for an REQ file.
- **Form Name:** The form name is typically determined prior to authoring the REQ file and is supplied in a requirements document (along with the *Document ID* and *Document Display Name*). The naming convention for a form name is to mirror the document name with no spaces. The form name is also the value used as the REQ file name.
- **Revision Date:** A revision date is used exclusively for static documents and represents the date the REQ-based document was last revised.
- **Document ID:** A required value defining the technical name for the REQ file that is used in processing. Valid values include the characters a-z, A-Z, 0-9, underscore, and a dash; invalid characters include spaces or special characters (&, ', []).
  - **Dynamic Requirements:** The naming convention for dynamic documents begins with INS followed by the form name and then the XML file extension. For example, *INS.ContributionInstructions.XML*.
  - **Static Requirements:** The naming convention for static documents begins with XTMP followed by the form name and then the XML file extension. For example, *XTMP.PowerOfAttorney.XML*.

**Note:**

The Document ID can not be changed after content references have been added to the document.

- **Document Display Name:** The form name as it is intended to appear in a list of documents. For example, a form name of *ContributionInstructions* might have a display name of *Contribution Instructions*. The display name is optional.
- **Document Style** Identifies the stylesheet used for default metadata values and to define the presentation of the form when a preview is generated.
- **State:** Used to define a jurisdiction target for the REQ file.
- **XTemplate:** A required value that associates the REQ file to an XTemplate file. This is used exclusively in static documents. Click **Browse** to navigate to the target XTemplate file.

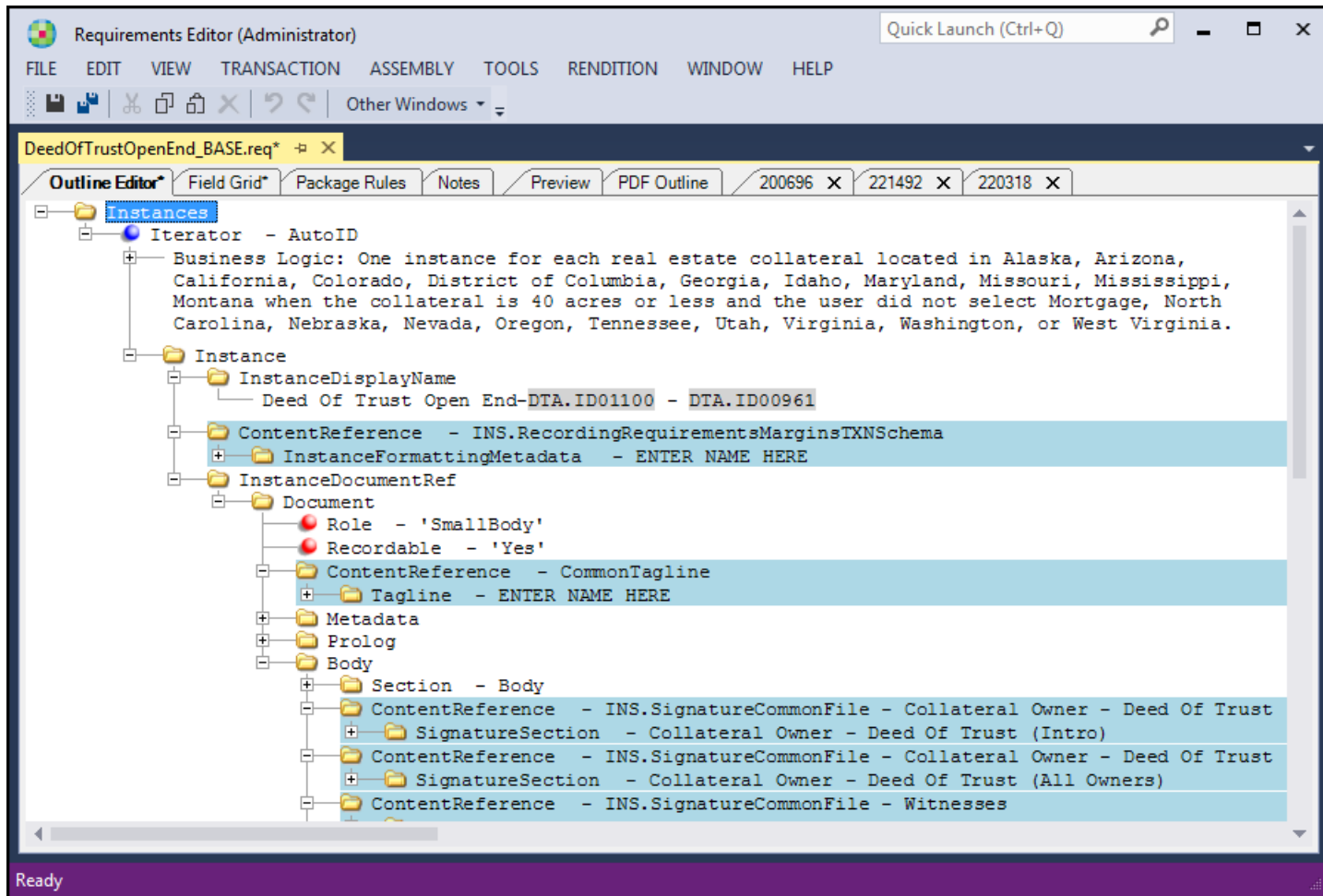
## Authoring Dynamic REQ Files

A dynamic requirement file, or dynamic document, is used to create a document on demand for financial transactions. Dynamic documents are tailored to a specific transaction and include only the text defined as relevant for a given transaction, hence the dynamic nature of the requirement file.

A dynamic document is constructed from words, sentences, paragraphs, and sections that are selected for the specific transaction and typically implemented through a series of data points, conditions, and rules. The content of a dynamic document is determined by applying a set of rules to prewritten content variations and a particular data set as part of content authoring performed in the Requirements Editor. For example, a dynamic document may contain information that applies to a unique jurisdiction or multiple collateral items.

By applying lender-specific business rules and style guides, documents are assembled and formatted during the transaction. Each document package can be tailored by a financial organization to fit its policy and transaction needs.

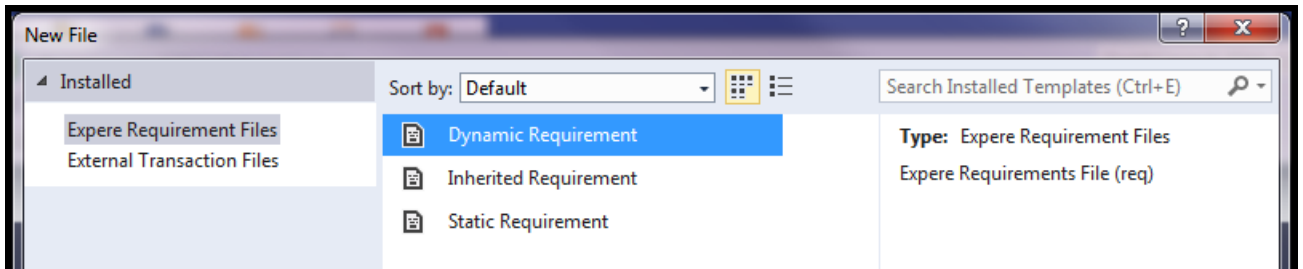
The Requirements Editor displays a dynamic document in the *Outline Editor View* and provides several views that are exclusive to dynamic documents.



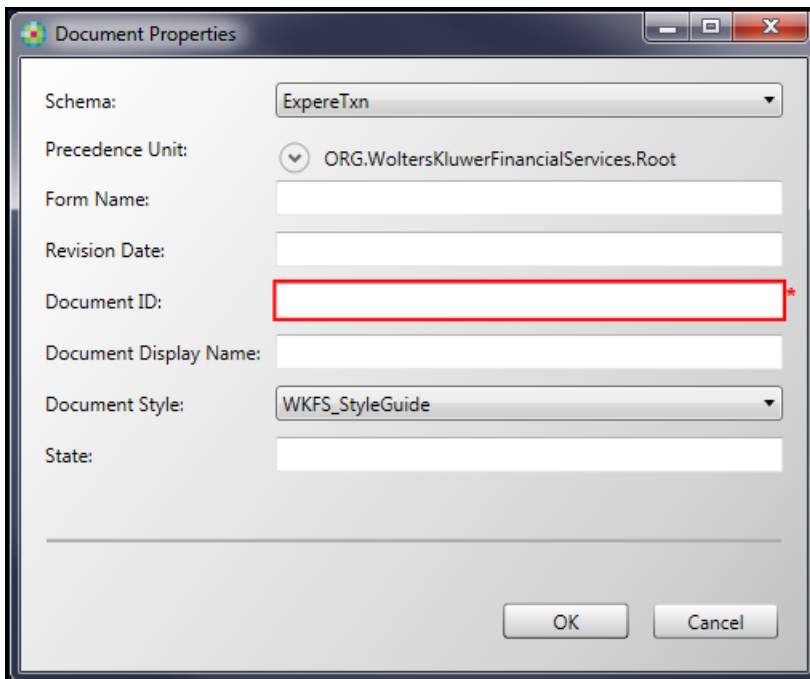
## Create a New Dynamic Requirements File

Before creating a new requirement file you should have available a requirements document that outlines the properties and design options for the new file.

1. In the **File** menu, click **New**.
2. In the **New File** dialog box, select **Expere Requirement Files** and then select **Dynamic Requirement** from the list of available options.



3. In the **New File** dialog box, click **Open**.  
The **Document Properties** dialog box opens.
4. Complete the file by filling in the text boxes with information from the requirements document.

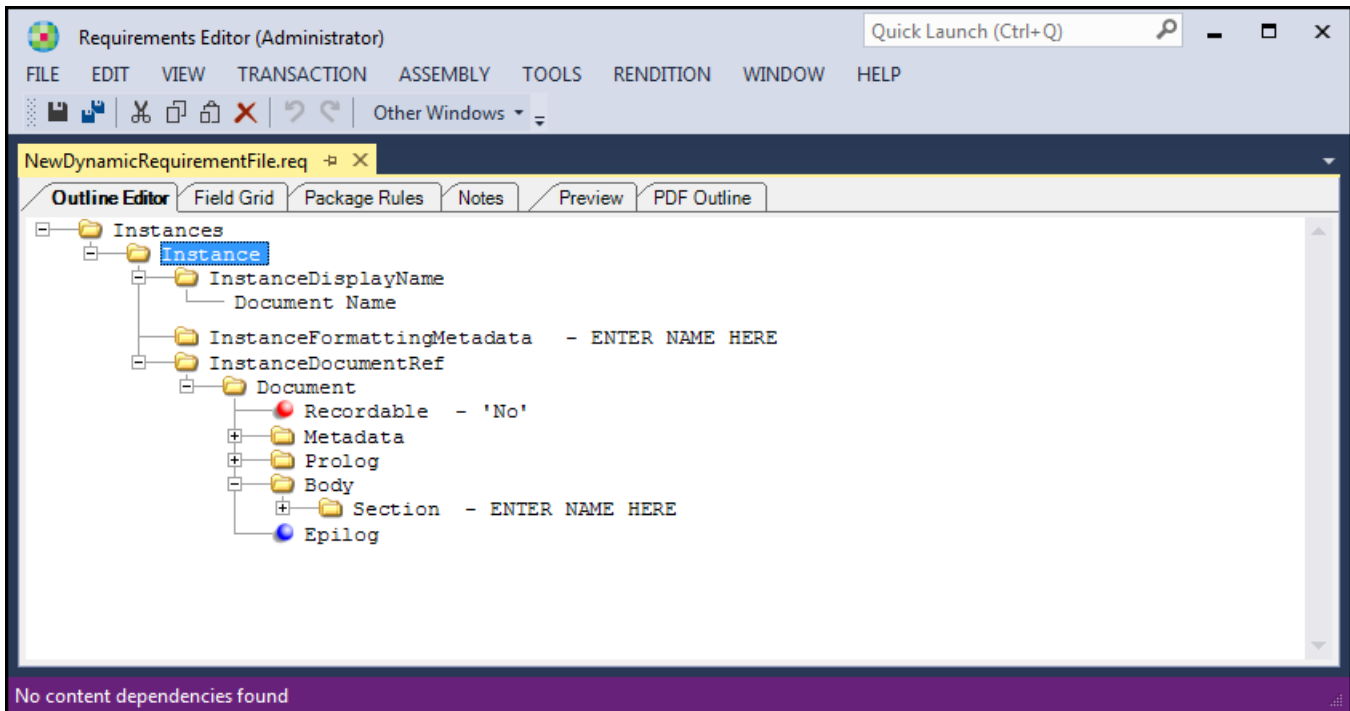


A dynamic requirement file document must have a *Document ID* defined in order to create the file.

5. Click **OK**.

The dynamic requirement file is opened in the Requirements Editor in the Outline Editor view as *NewFileNN.req*.

6. In the Requirements Editor, save the file by clicking **File > Save**. The best practice is to use the form name as the REQ file name; the target file name will, most likely, be defined in the requirements.



The dynamic requirement file is saved and available for authoring.

## Authoring Static REQ Files

A static requirement file, or static document, is written and prepared in advance for particular processes or lending transactions. Static documents are created with specific content in a fixed format with compliance content that does not change with each transaction. In a pure static model, static content is brought into the Requirements Editor through an XTemplate (layout-driven) image. Data fields are in a fixed position; if the same data is required in more than one place, the data field is positioned in each of those places. Data populates the data fields at runtime and any discrepancy between transaction data received and the properties of the fixed data fields is handled by overflow rules including overflow to a target (an addendum, for example). Other mechanisms to accommodate data overflow include truncated text, and reduced font size.

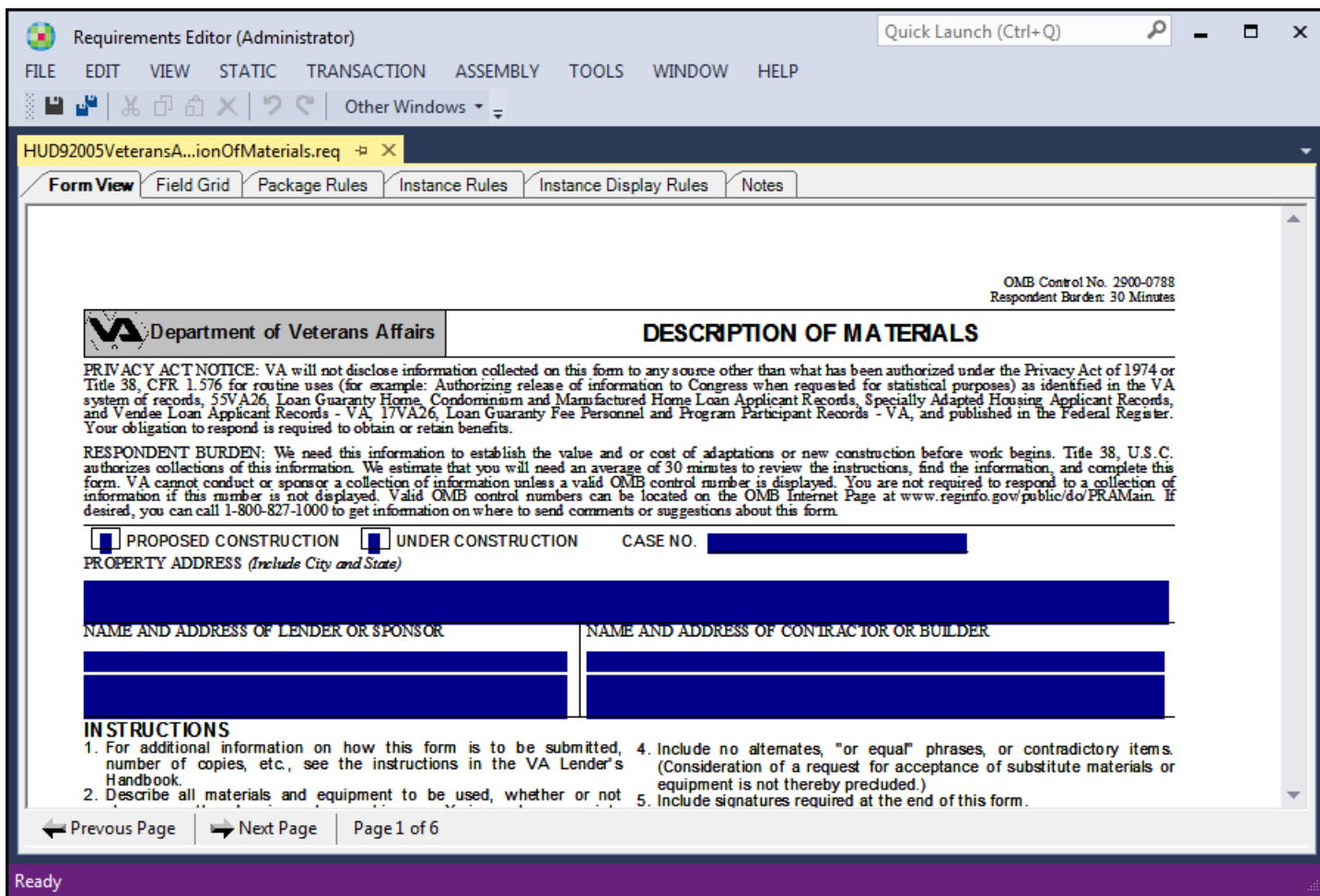
Static documents may include some or all of the following features:

- Defined input fields
- Calculations
- Validations

- Links to instruction sheets
- Intelligent data merge
- Overflow for handling Signatures or specific data fields

In a static document, check boxes and empty text fields that must be completed during each transaction appear on the documents with data that is typically entered by completing prompts during document preparation. If not populated during document preparations, the static documents can be printed with blank values or non-filled check boxes and text fields.

The Requirements Editor displays a static document in the *Form View* and provides several views that are exclusive to static documents.



Requirements Editor (Administrator) Quick Launch (Ctrl+Q)

FILE EDIT VIEW STATIC TRANSACTION ASSEMBLY TOOLS WINDOW HELP

HUD92005VeteransA...ionOfMaterials.req

Form View Field Grid Package Rules Instance Rules Instance Display Rules Notes

OMB Control No. 2900-0788  
Respondent Burden: 30 Minutes

**Department of Veterans Affairs** **DESCRIPTION OF MATERIALS**

PRIVACY ACT NOTICE: VA will not disclose information collected on this form to any source other than what has been authorized under the Privacy Act of 1974 or Title 38, CFR 1.576 for routine uses (for example: Authorizing release of information to Congress when requested for statistical purposes) as identified in the VA system of records, 55VA26, Loan Guaranty Home, Condominium and Manufactured Home Loan Applicant Records, Specially Adapted Housing Applicant Records, and Vendor Loan Applicant Records - VA, 17VA26, Loan Guaranty Fee Personnel and Program Participant Records - VA, and published in the Federal Register. Your obligation to respond is required to obtain or retain benefits.

RESPONDENT BURDEN: We need this information to establish the value and or cost of adaptations or new construction before work begins. Title 38, U.S.C. authorizes collections of this information. We estimate that you will need an average of 30 minutes to review the instructions, find the information, and complete this form. VA cannot conduct or sponsor a collection of information unless a valid OMB control number is displayed. You are not required to respond to a collection of information if this number is not displayed. Valid OMB control numbers can be located on the OMB Internet Page at [www.reginfo.gov/public/do/PRAMain](http://www.reginfo.gov/public/do/PRAMain). If desired, you can call 1-800-827-1000 to get information on where to send comments or suggestions about this form.

PROPOSED CONSTRUCTION  UNDER CONSTRUCTION CASE NO. [REDACTED]

PROPERTY ADDRESS (Include City and State)  
[REDACTED]

NAME AND ADDRESS OF LENDER OR SPONSOR [REDACTED]

NAME AND ADDRESS OF CONTRACTOR OR BUILDER [REDACTED]

**INSTRUCTIONS**

1. For additional information on how this form is to be submitted, number of copies, etc., see the instructions in the VA Lender's Handbook.
2. Describe all materials and equipment to be used, whether or not
3. Include a list of materials and equipment to be used, whether or not
4. Include no alternates, "or equal" phrases, or contradictory items. (Consideration of a request for acceptance of substitute materials or equipment is not thereby precluded.)
5. Include signatures required at the end of this form.

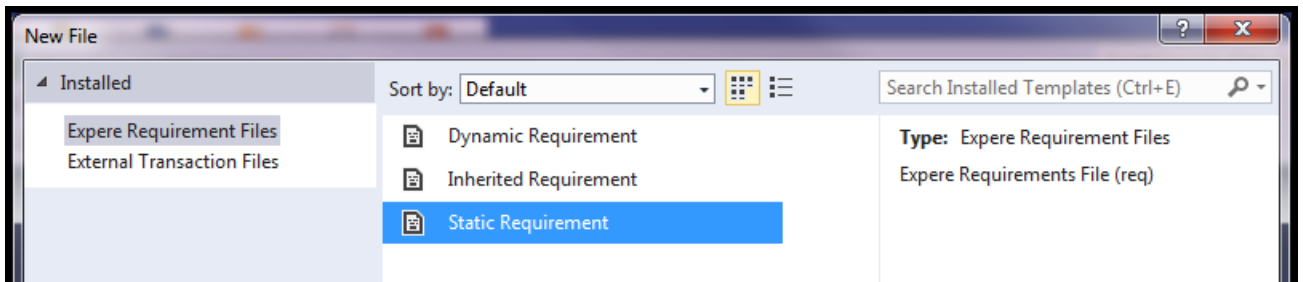
← Previous Page Next Page → Page 1 of 6

Ready

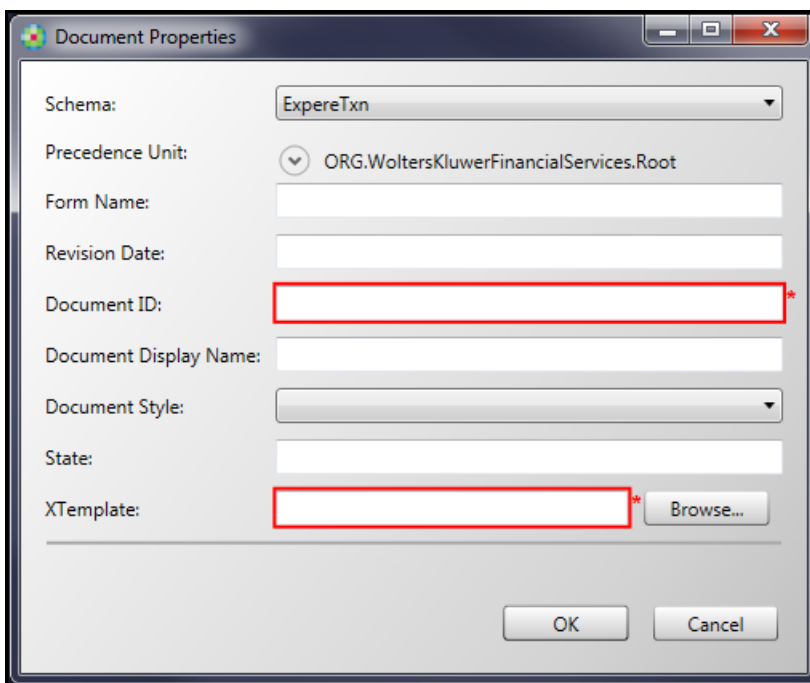
## Create a New Static Requirement File

Before creating a new requirement file you should have available a requirements document that outlines the properties and design options for the new file.

1. In the **File** menu, click **New**.
2. In the **New File** dialog box, select **Expere Requirement Files** and then select **Static Requirement** from the list of available options.



3. In the **New File** dialog box, click **Open**.  
The **Document Properties** dialog box opens.
4. Complete the file by filling in the text boxes with information from the requirements document.



A static requirement file document must have a *Document ID* and *XTemplate* defined in order to create the file.

5. Click **OK**.  
The static requirement file is opened in the Requirements Editor in the Outline Editor view as *NewFileNN.req*.

6. In the Requirements Editor, save the file by clicking **File > Save**. The best practice is to use the form name as the REQ file name; the target file name will, most likely, be defined in the requirements.

The static requirement file is saved and available for authoring.

## Authoring Inherited REQ Files

An inherited requirement file, or inherited document, is used to create a document on demand that is based on (*inherits*) the organizational structure of another document but does not carry over any content from the source document. It is in this respect that creating a new inherited document differs from creating a new document as a copy (save as) of another document.

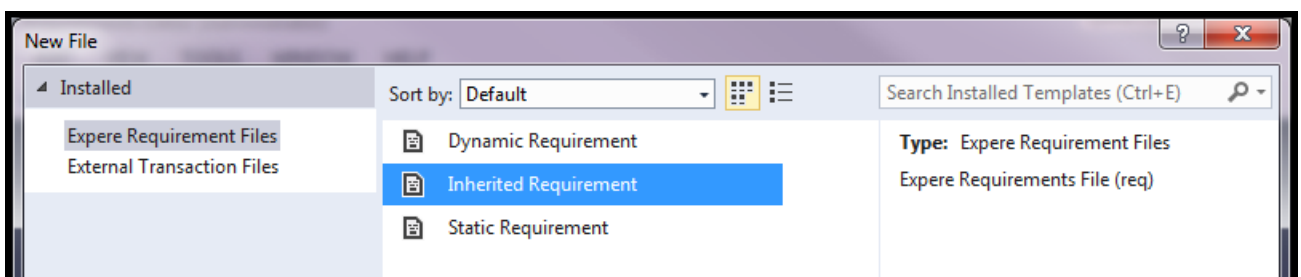
Inherited documents are tailored to a specific repository and organization structure. When creating a new inherited document, the *Authoring Web Service* must be pointed to a the repository containing the source document from which the organization structure is inherited. The Authoring Web Service is defined in the **Options** dialog (**Tools > Options**) for the Expere environment. If this setting is changed, you will want to close and reopen the Requirements Editor for the change to take effect.

The Requirements Editor displays an inherited document in the *Outline Editor View* and provides several views based on the source document type (dynamic or static).

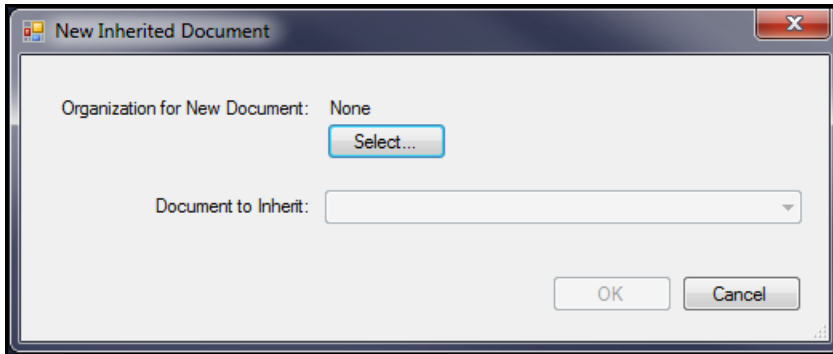
### Create a New Inherited Requirement File

Before creating a new requirement file you should have available a requirements document that outlines the properties and design options that should be used in creating the new file.

1. In the **File** menu, click **New**.
2. In the **New File** dialog box, select **Expere Requirement Files** and then select **Inherited Requirement** from the list of available options.



3. In the **New File** dialog box, click **Open**.
4. In the **New Inherited Document** dialog box:



- Click **Select** to define the organization to associate with the document.
- Select the document to inherit.

5. Click **OK**.

## Authoring EPX Files

### Creating a Requirements File from an Existing File (Save As)

The **Save As** option can be used to save an open requirements file as another file using a unique file name and document identifier. This option is mainly used to save an existing REQ file as a local copy for testing and development purposes but can also be used to save a copy to a network location or workspace.

In order to maintain the integrity of document identifiers (for example, the *Document ID*) and downstream impacts to content references in the REQ, you must enter a unique Document ID as part of saving an REQ file as another file (or copy). By default, the text *\_Copy* is appended to the Document ID in the *Document Properties dialog box*. You may keep the defaults or define a unique name for both the file and Document ID.

**Note:**

The Save As option can be used to save both locked (read-only) and writeable REQ files. When the source file is locked with the read-only restriction, the save as operation removes the restriction and saves the file as writeable.

### Create a New Requirements File From an Existing File

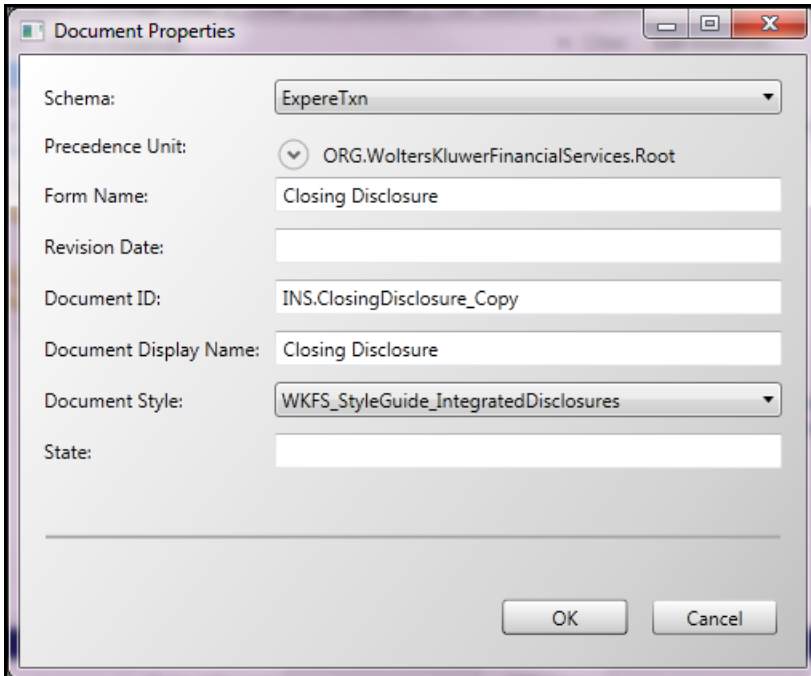
The requirements file that you want to base the new file from must be open in the Requirements Editor.

As an author, you may wish to create a new REQ file, based on an existing file, as a local copy for testing and development purposes. This feature is used to save an open requirements file as another file using a unique file name and document identifier.

1. In the **File** menu, click **Save As**.

2. In the **Save As** dialog box, enter new file name/location and click **Save**.
3. In the **Document Properties** dialog box, the **Document ID** field is presented as an editable field with *\_Copy* appended to the Document ID value (*[DocumentIDValue\_Copy]*).
  - Accept the default Document ID (*\_Copy*) or,
  - Enter a unique Document ID value.

Once a Document ID is established and saved; you cannot edit it.



The screenshot shows the 'Document Properties' dialog box with the following values:

Field	Value
Schema	ExpereTxn
Precedence Unit	ORG.WoltersKluwerFinancialServices.Root
Form Name	Closing Disclosure
Revision Date	
Document ID	INS.ClosingDisclosure_Copy
Document Display Name	Closing Disclosure
Document Style	WKFS_StyleGuide_IntegratedDisclosures
State	

4. Click **OK**.

The requirements file is displayed in the **Outline Editor** perspective and saved to the location defined in this procedure.

## Setup and Configuration Features

This section describes the key configuration options available from within the Requirements Editor. These features are mainly related to establishing connects to a data source and defining certain dependencies required for successful content authoring.

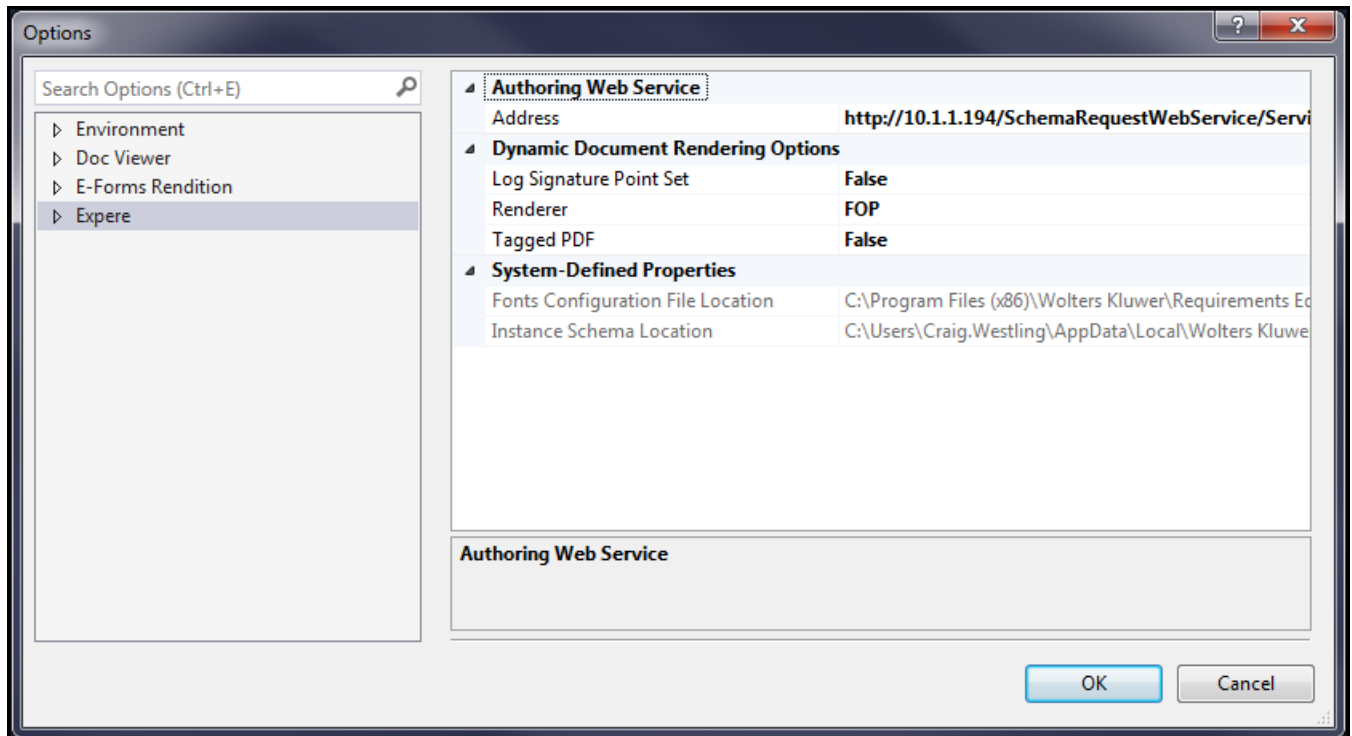
While most of these options remain consistent as you perform content authoring, you may run into situations that require an adjustments to the configuration settings for your Requirements Editor environment.

### Configuring for Expere Content Repositories

In most cases you will be working with a collection of files contained within a repository when creating new files or modifying existing files. The Requirements Editor allows you to open a content file (REQ or other XML file) that is contained within a repository or outside a repository. There are several different ways to access files for authoring; for example, you might

- copy a file or group of files from a repository in a build folder to your local machine,
- work directly in a repository on your local machine or network share, or
- access a repository from a content library contained on a CD or other media.

Each REQ file is associated to a repository through the content references contained in the file. While you can author an REQ outside of a repository; the content references must resolve to the repository that is pointed to in the content references. These pointers are defined in the *Authoring Web Service* that the Requirements Editor is configured to access in the **Options** dialog box. In order for an REQ to resolve content references and successfully generate a document, the Requirements Editor must be pointed to a valid repository accessible to the *Authoring Web Service*. The Expere Authoring Web Service is defined in the **Options** dialog box (**Tools > Options**).



### About the Expere Knowledge Base Repository

Content is often accessed through a Expere Knowledge Base repository corresponding to a particular release, build, or line of business. Most often, content is saved back to the same repository or new content files are created and saved to a particular repository.

An Expere Knowledge Base repository contains content that, ultimately, is used to generate compliance documentation for financial transactions. For the purposes of this guide, there will be two repositories discussed: the authoring repository, where custom content is created and integrated with Wolters Kluwer Financial Services' content, and the build repository, which may be deployed to a production environment or to another environment such as testing. The content in the authoring repository will generally be under some type of source control or content management system that will place ("check out") files on the file system to be edited. You can open these files from the file system into the Requirements Editor.

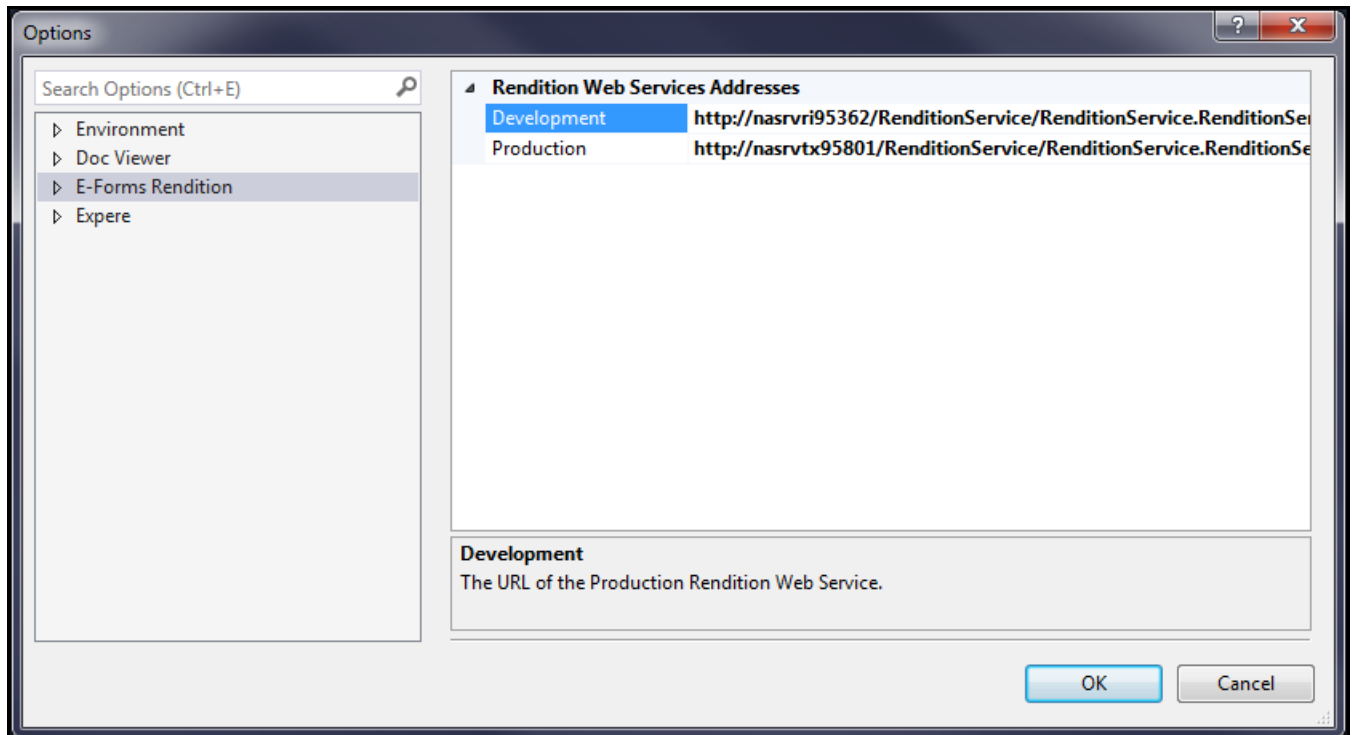
When installed, the standard Expere Knowledge Base repository structure consists of a file system folder containing a folder named *WoltersKluwerFinancialServices.root*, which contains Expere content organized in sub-folders, and an XML file, *Organizations.xml*, that contains information about the repository structure. Within the repository root, the repository is organized to contain folders for each line of business (LOB). From a customer perspective, each LOB is licensed independently; consequently, Wolters Kluwer Financial Services has organized the repository folder structure to contain a folder for each LOB. The folder structure also contains common and shared files that are used across LOBs. At the LOB root folder, the repository is contained in the ExpereRepository folder and will look something like this:

ExpereRepository	2/26/2014 2:39 PM	File folder
ORG.WoltersKluwerFinancialServices.Root	2/26/2014 2:39 PM	File folder
Schemas	2/26/2014 2:39 PM	File folder
Organizations.xml	2/26/2014 3:54 AM	XML File
Profiles.xml	2/26/2014 3:54 AM	XML File

## Configuring for E-Forms Rendition Integration

The Requirements Editor provides support for using the E-Forms Rendition plug-in. Support for the Rendition Plug-in is enabled during installation using the *Advance Installation* option. Once installed, you must configure the E-Forms Rendition Web Services for use in the Requirements Editor to establish a connection to the E-Form Rendition environment.

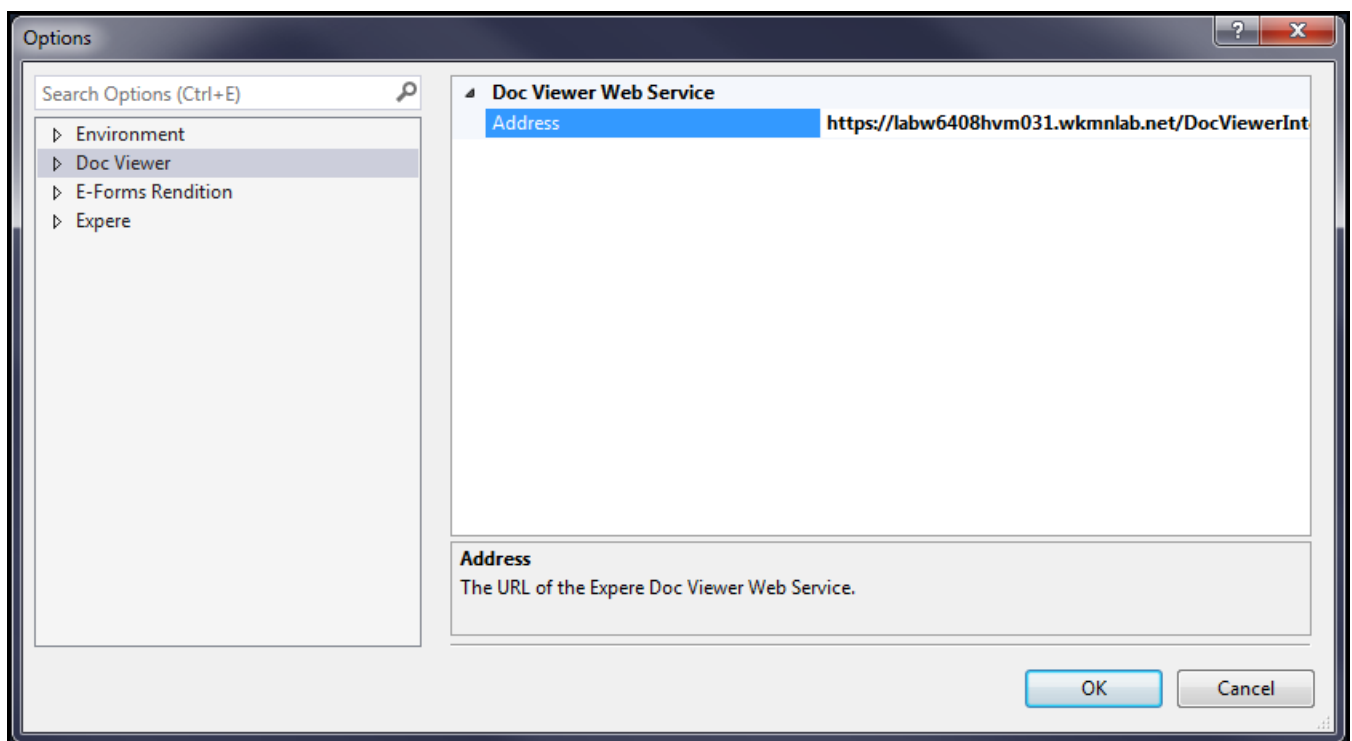
The URL for the web service is defined in the *Options* dialog box (*Tools > Options*). In order for an REQ to resolve content references and successfully integrate with the E-Forms environment, the Requirements Editor must be pointed to a valid production and development repository accessible through the *Rendition Web Services*. Click the *Development* or *Production* text box and copy/paste a valid URL for the target web service.



## Configuring for DocViewer Integration

The Requirements Editor provides support for using the tool in an integrated fashion with the DocViewer application. Support for the DocViewer is enabled during installation using the *Advance Installation* option. Once installed, you must configure the DocViewer Web Service for use in the Requirements Editor to establish a connection to the DocViewer repository.

The URL for the web service is defined in the *Options* dialog box (*Tools > Options*). In order for an REQ to resolve content references and successfully integrate with a DocViewer environment, the Requirements Editor must be pointed to a valid repository accessible through the *DocViewer Web Service*. Click the *Address* text box and copy/paste a valid URL for the target web service.



## Working with Stylesheets (StyleGuides)

This section introduces the collection of StyleGuides and describes how to update style sheets and author style sheet overrides in a document.

- About StyleGuides
- Updating Stylesheets
- Stylesheet Options and Override Details

## About StyleGuides

The Requirements Editor, and Expere authoring generally, uses a collection of StyleGuides that define the document transformation and presentation. These style guides are based on Extensible Stylesheet Language Family (XSL) recommendations and, as XSL style sheets, describe how to display an REQ document.

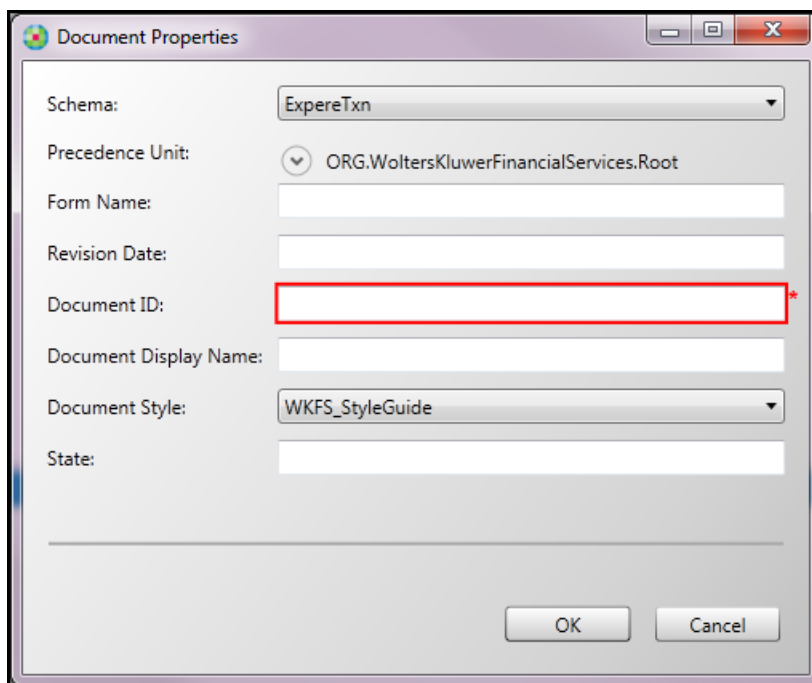
The style guides are included in the Requirements Editor program files along with all other XSL style sheets available to the tool. The location of the style guides is:

```
C:\Program Files (x86)\Data Management\Requirements Editor\Resources\ContentItems
```

The primary style guides published by Wolters Kluwer Financial Services and supported for authoring in the Requirements Editor include:

- WKFS\_StyleGuide.xsl (default)
- WKFS\_StyleGuide\_Commercial\_ModifiedStandard.xsl
- WKFS\_StyleGuide\_Commercial\_Standard.xsl
- WKFS\_StyleGuide\_FannieFreddie.xsl
- WKFS\_StyleGuide\_IntegratedDisclosures.xsl

These style guides (or XSL style sheets) are defined when you create a new document as the **Document Style**. When creating a new document, whether a dynamic or static document, you must define the style guide to associate with the document in the *Document Properties* dialog box.



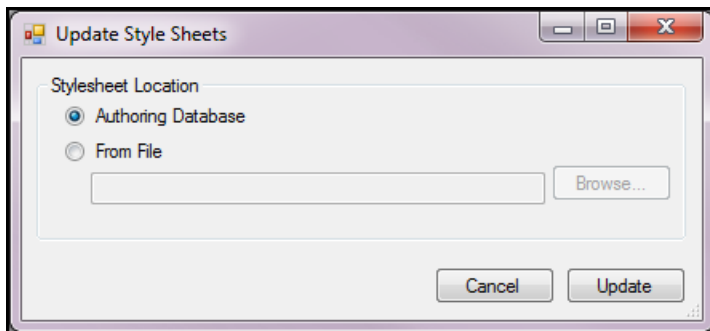
The associated style guide, as the Document Style, is defined in the source XML in the *document type* element.

```
<document type="Dynamic"
transactionSchemaUri="http://schemas.bankerssystems.com/2005/VMPMISMO"
documentStyle="WKFS_StyleGuide"
formName="AddendumToLoanApplication"
revisionDate=""
documentID="INS.AddendumToLoanApplication"
documentDisplayName="Addendum To Loan Application"
organization="ORG.WoltersKluwerFinancialServices.Root"
state=""
xTemplatePath="">
```

## Updating Style Sheets

You can update the installed style sheets from the primary authoring database, or upload a custom style sheet for use in authoring REQ files by using the update style sheet feature. In the **Tools** menu, select **Update Style Sheets**.

The **Update Style Sheets** window allows users to update the style sheets used by the Requirements Editor in content assembly. The options available for updating the style sheets are: **Authoring Database** (requires updating the authoring database before you can select this option) and **From File**.



Select the *Authoring Database* option to retrieve style sheet items from the authoring database associated with the active Expere Authoring Web Service URL (defined in the Options dialog). After you click **Update**, the style sheets contained in the Requirements Editor program files folder (*C:\Program Files (x86)\Data Management\Requirements Editor\Resources\ContentItems*) are replaced with style sheets from the authoring database.

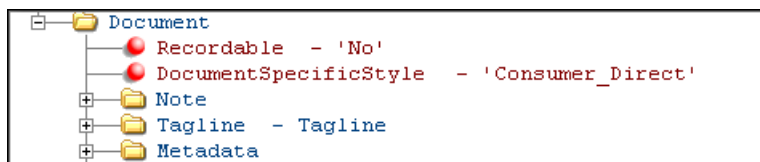
**Note:**

This option required an update to the Expere Administration Tool so it can load the style sheet items from content to the authoring database.

Select the *From File* option to retrieve style sheets stored on a local or network drive. This option is used when editing and testing style sheets that are not included in the Requirements Editor program files or the Expere Authoring Web Service. When choosing this option, click **Browse** to open a folder location where the target style sheets are stored.

## Stylesheet Options and Override Details

The following style sheets are used when creating content. The bullet items below each style sheet detail the items that are an override from the standard WKFS\_StyleGuide. To select a style sheet override, in the outline editor, enter one of the following values into the **DocumentSpecificStyle** attribute on the **Document** element.



- **WKFS\_StyleGuide:** Follows the WKFS StyleGuide requirements. Standard 10pt base font, 11” page length, etc.
- **Consumer\_Direct:** 9pt base font size override. CT state specific changes for title size (2pts above base font size) and spacing between sections (3/16th). The 9pt base font size meets the minimum required for some NY and NJ documents.
- **Consumer\_Indirect:** 8pt base font size override in addition to using condensed font as the base. Eform outputs will have field underlines for all fields except multiline fields. CT state specific changes for title size (2pts above base font size) and spacing between sections (3/16th). Other state required style overrides should be set using the InstanceFormattingMetadata (ex. NY and NJ 9pt minimum base font)
- **Privacy:** The tagline will only print on page one (1).

## Working with Schema Files

This topic describes the configuration options available for managing and defining schema files. If a document is authored to use data from a particular transaction schema it cannot use data from a different transaction schema. Therefore, data must be transformed to the schema used by the documents of interest.

Documents are collected by line of business. In the current Expere IE release, documents for the Commercial, Consumer Direct, Deposit, IRA, HSA and Home Equity are authored to use data from the ExpereTxn schema and documents for the Mortgage line of business are authored to use data from the VMPMISMO schema. While ExpereTxn is the preferred authoring schema, documents created in the future for other lines of business may employ ExpereTxn, VMPMISMO, or another schema.

### Schema Types

Associating a schema to a document is required. By default, three schema files (xsd) are supported by the solution. Each is installed with the solution and are configured to display as options in the *Document Properties* dialog box when creating a new document.

- **ExpereTxn:** The base schema for the majority of documents and the default schema.
- **VMPMISMO:** The VMP Mismo schema is used primarily for documents supporting the Mortgage lending line of business.
- **QuickDoc:** The QuickDoc schema is used in conjunction with documents that are processed through the QuickDocs component to render eForms documents.

The configuration file contains records that define the available schema. The configuration file is located in the default installation path.

```
C:\Program Files (x86)\Data Management\Requirements Editor
\RequirementsEditor.exe.config
C:\Program Files\Data Management\Requirements Editor\RequirementsEditor.exe.config
```

You can modify this setting to add support for additional custom schema files. In this example, a custom schema is defined, along with the three packaged schema files, and are available as options when creating a new document.

```
<section name="transactionSchemaSection" type="PTR.TransactionSchemaSection, PTR" />
<Schemas>
  <Schema name="ExpereTxn" default ="true" displayName ="ExpereTxn" rootElement="Txn"
  targetNamespace="http://schemas.bankerssystems.com/2004/ExpereTxn" location="http://
schemas.bankerssystems.com/ExpereTxn/ExpereTxn.xsd" />
  <Schema name="VMPMISMO" default ="false" displayName ="VMP MISMO" rootElement="LOAN"
  targetNamespace="http://schemas.bankerssystems.com/2005/VMPMISMO" location="http://
schemas.bankerssystems.com/Mortgage/VMP/VMP_Expere.xsd" />
  <Schema name="QuickDoc" default ="false" displayName ="Quick Docs"
  rootElement="QuickDocs" targetNamespace="http://schemas.bankerssystems.com/QuickDoc"
  location="http://schemas.bankerssystems.com/QuickDocs/QuickDocs.xsd" />
  <Schema name="custom_schema" default ="false" displayName ="CustomSchema"
  rootElement="CustomSchema" targetNamespace="http://schemas.bankerssystems.com/
custom_schema" location="http://schemas.bankerssystems.com/custom/custom_schema.xsd" />
</Schemas>
</transactionSchemaSection>
```

**Attention:**

Adding a schema through the Requirements Editor makes that schema available; it does not, however, imply the schema is valid. All schema files should be tested to ensure compatibility with the Requirements Editor prior to being added to the solution.

Attribute	Description
name	The name of the schema, this must match the last part of the targetNamespace.
default	Define this as <i>true</i> to establish the schema as the default schema for the system.
displayName	The schema name as you want it to appear in the list of available options and within the application.
rootElement	The name of the root to use as a starting point for finding other elements.
targetNamespace	Specify the namespace that the schema targets, or validates.
location	The location, HTTP or file system, of the schema file (xsd).

## Managing Fonts

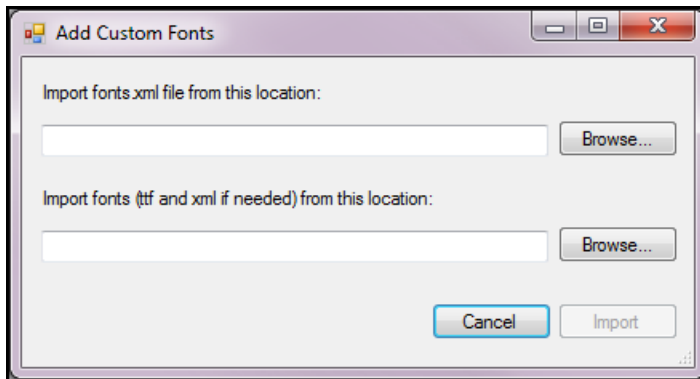
This section discusses configuring the Requirements Editor to utilize a custom font collection when rendering output documents. You can add a custom fonts set from within the Requirements Editor or you through modifying configuration files.

**Note:**

To change or modify the font and color scheme used in the application interface, select **Tools > Options** and in the Environment options, click **Fonts and Colors**.

### Adding Custom Fonts Through the Requirements Editor

In the Tools menu, click **Add Custom Fonts**. The feature is disabled if an REQ file is open; close any open files before choosing this option.



Using this option you can import a fonts.xml file as well as the individual font files (ttf).

### Adding Custom Fonts Through Configuration Files

**Note:**

FOP users must have the Apache library installed prior to completing this procedure.

1. Copy the desired font file to the following directory:
  - Expere .NET: C:\inetpub\wwwroot\Expere\bin\resources\fonts
  - Requirements Editor: C:\Program Files (x86)\Data Management\Requirements Editor\ExpereNet\resources\fonts
  - WebSphere: update the ExpereIEServices.ear file at ExpereIEServices.ear\resources\fonts
2. Add the desired fonts to one of the following directories, observing the naming conventions as shown below. Save this file.
  - Expere .NET: C:\inetpub\wwwroot\Expere\bin\resources\fonts
  - Requirements Editor: C:\Program Files (x86)\Data Management\Requirements Editor\ExpereNet\resources\fonts
  - WebSphere: ExpereIEServices.ear\fonts.xml

```

<font>
<font name="Agency FB" resource="resources/fonts/AGENCYR.TTF" style="normal"
weight="normal" embed="false"/>
<font name="Agency FB" resource="resources/fonts/AGENCYB.TTF" style="normal"
weight="bold" embed="false"/>
<font name="Pi Font" resource="resources/fonts/pifont.ttf" style="normal"
weight="normal" embed="true"/>
.....
  
```

```
<font name="Verdana" resource="resources/fonts/verdanaz.ttf" style="italic"
weight="bold" embed="false"/>
</font>
```

3. Within *<release version>\Release\BuildFOPFontMetricsInstaller* folder, select **setup.exe** to launch the **BuildFOPFontMetricsInstaller**. Double-click the file to install.

**Note:**

The install screens follow standard installation protocol; click **Next** to advance through each screen. Click **Close** to exit and complete the installation.

4. FOP users: open a command line by selecting Start/Run/Cmd:
  - Change the folder to reflect where BuildFOPFontMetricsInstaller was installed.
  - Run the batch file to see the usage instructions.
  - Copy the .XML file that was created to the pertinent location (reflected in step 1 above).

To test this configuration, go to the **Preview** view in the Requirements Editor and right-click in the PDF. Select **Document Properties** and, in the Fonts tab, verify that the new font exists.

## Understanding Key Concepts

This section describes some of the key architectural concepts providing the foundation for how requirements files are integrated into the Expert Knowledge Base content.

- Business Logic
- Precedence (Organizational Hierarchy)
- Instancing
- References

### About Business Logic

All of standard rules, whether they are inclusion rules (conditions) or data rules, reside in a database; this allows another user to reuse rule rather than creating new ones. When creating or revising a document, review the database and choose an existing rule. The rules are going to be linked with a PTR, so that once the outline drafter chooses an existing business rule, a PTR will not have to be written (although the PTR will have to be reviewed to make sure that it is correct for the given document). Testing this guy.

Users should ensure that rules should contain sufficient description. For example, "Owner's name," may be adequate rule when working with IRA documents, but would be ambiguous when also considering lending rules; the "Owner" in lending is an owner of collateral.

Business Logic must be written in a complete sentence. For example:

- The state laws of Alaska apply to the transaction.
- The IRA Owner is an individual.
- The IRA Plan Type is Roth.

### About Precedence

Precedence works to override content from the top (parent) to the lower (child) organizations. For example, all base content fragments that might be included in a document are attached to the Wolters Kluwer Financial Services precedence level and may be overridden at any other level in the hierarchy. If Organization 2 overrides a particular content fragment, documents assembled in the context of Organization 5 or Organization 6 will incorporate the override rather than the base version of that fragment, but organizations 4, 9, and 10 would be unaffected. Each organization in the precedence hierarchy is identified by a Universal Unique Identifiers (UUID).

Several organizations may exist in an enterprise and relate to each other in a hierarchical fashion; precedence is the expression of that organization hierarchy. Wolters Kluwer Financial Services is always at the top of the hierarchy and the financial institution and suborganizations (e.g., lines of business) are below.

Precedence provides the infrastructure for the following:

- Utilize base Expere Knowledge Base content (language, rules, etc.)
- Supersede base Expere Knowledge Base or other parent\ancestor organization content at a particular org level and its descendents
- Create custom content for use at a specific org level and its descendents

When the document assembler encounters two or more components with the same name that meet the requirements of the needed document, the assembler chooses the component based on precedence, which is a process by which the system selects document components according to the identity of the entity requesting the document, or, if no matching component is found, according to the identity of one of the entity's parents or ancestors. If no match is found, a default component is provided. This concept is referred to as "precedence" or "owner precedence."

## About Instancing

Use an **Iterator** to perform document instancing before creating content within the **Body**. The iterator resides directly before the **Instance** element. The document level iterator contains business logic and a corresponding PTR. The document level iterator determines how the rules are written for the entire document.

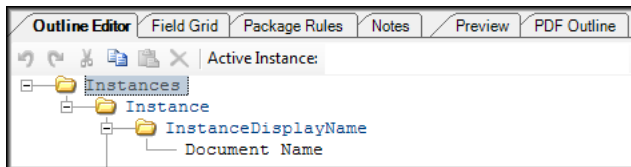
Each instance signifies a separate document for a transaction. For example, a document may be required for each borrower on the loan.

**Instances:** The container for all of the instances.

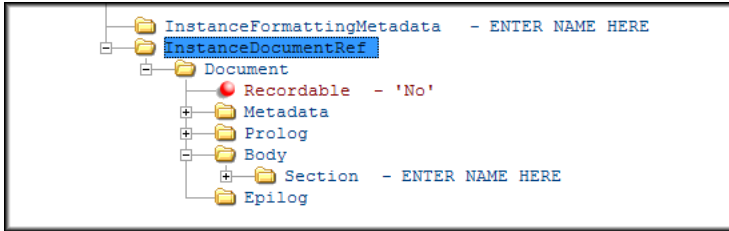
**Instance:** The container for a specific version of the printed document.

**InstanceFormattingMetadata:** When a new REQ file is created, the base InstanceFormattingMetadata is blank. The InstanceFormattingMetadata can be changed if there is a business need. The stylesheet will apply standard formatting. Elements under InstanceFormattingMetadata only need to be entered in the .REQ if it needs to override the defaults in the stylesheet. To override the available elements, right mouse click on the InstanceFormattingMetadata element and add the needed elements and their values.

**InstanceDocumentRef:** Contains content referenced from a specific schema (for example: ExpereDoc)

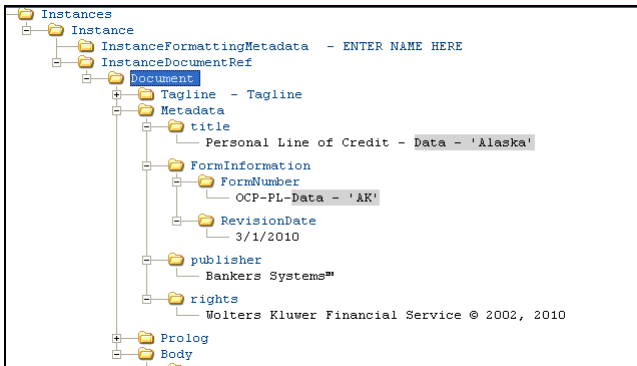


**Document:** The **Document** element is a container for content of the document (everything except the instancing and formatting metadata. Its children typically include the Tagline, Metadata, Prolog and Body.

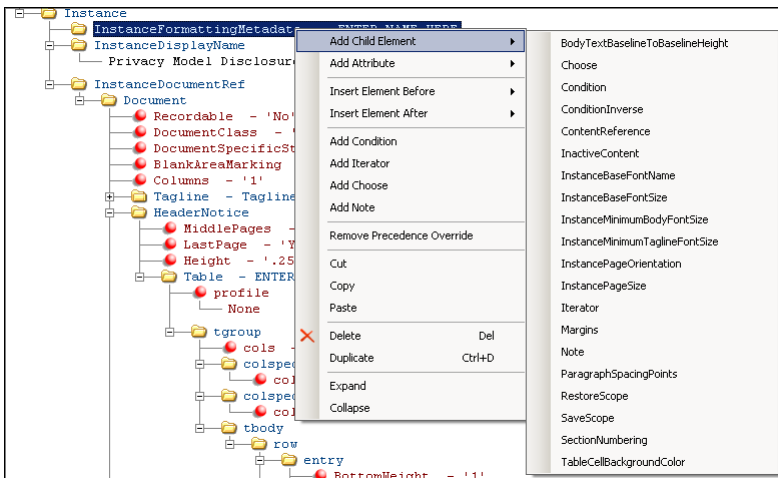


## Instance Formatting Metadata

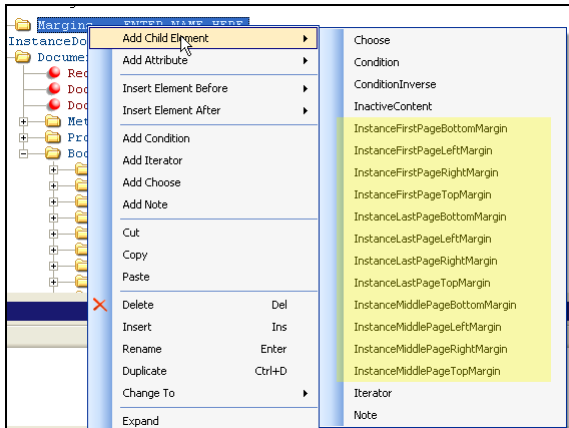
After creating a new .REQ file, the **InstanceFormattingMetadata** appears with no value by default: no overrides have been indicated for the stylesheet selected.



Users can modify the **InstanceFormattingMetadata** if a business need exists. The stylesheet will apply standard formatting. Enter elements under **InstanceFormattingMetadata** only if a need exists to override the defaults in the stylesheet. To add override elements, right-click on the **InstanceFormattingMetadata** element and add the needed elements and their values.



- **InstancePageSize** - The default is **Letter**; however, you can set the value to **Legal** to create the document for legal size paper or set the number of inches.
- **InstancePageOrientation** - The default is **Portrait**; however, you can set the value to **Landscape** to rotate the page orientation..
- **InstanceBaseFontName** - The default is **Serif**; only change this value if an approved product override or a statutory requirement exists. The supported overrides are:
  - **Sans** - prints the body of the document in Sans Serif font
  - **Condensed** - prints the body of the document in a condensed font (ex. Arial Narrow)
- **InstanceBaseFontSize** - The default is **10pt**. Only change this value if an approved product override or a statutory requirement exists.
- **InstanceMinimumBodyFontSize** - Certain states require a minimum font size. For example, IL UCC require a 12 point font; the user should set Instance Minimum Body Font Size to 12 and Instance Base Font Size to 12.
- **InstanceMinimumTaglineFontSize** - For recordable documents, several states that in addition to having a requirement for the body font size also require the tagline to be a minimum font size. The requirement is typically **10pt** for the override.
- **BodyTextBaselineToBaselineHeight** - It is sometimes necessary to modify the spacing between lines of text slightly to fit all necessary text on the first page or to avoid having a last page with only a few lines of text. The default value is always one (1) pt larger than the base font size (ex. 11pt on our standard 10pt documents). If implementing an override, reduce this number by .5 pt (ex. 10.5pt on our standard 10pt documents). Additional reduction will cause text to begin overlapping. This value should be profiled if only required for certain instances (ex. governing jurisdictions) of the document.
- **ParagraphSpacingPoints** - Similar to **BodyTextBaselineToBaselineHeight**, use this option to modify spacing in the document. This option modifies the setting for the spacing between paragraphs and sections. The default value is 3pt; it is suggested that the override should be at least 1pt to maintain some spacing between paras and sections. This value should be profiled if only required for certain instances (ex. governing jurisdictions) of the document.
- **SectionNumbering** - Use this option to specify if only certain governing jurisdictions of the EPM content require section numbering. This occurs with the NJ Motor Vehicle Lease agreement.
- **Margins** - The following example displays the available elements for children of **Margins**:



- For all of the **InstanceXXX** elements (highlighted in yellow above) the default for this value varies depending on the stylesheet used. For non-recordable content the default is 0.5in; for recordable content the default is 0.6in. Modify this only if a statutory requirement (ie. recordable document) exists. For recordable documents, use the common file to maintain consistency across documents. The common file uses both the top margin value and a table with a **RowHeight** attribute to meet the overall top margin requirements.
- The last page's margins cannot be larger than any middle page margins.

**Note:**  
Rider documents should always have its margins set to the 'last page margin' requirements

- **TableCellBackgroundColor** - Use this value to determine the color or shading used for those table cells with the **BackgroundColor** attribute set to **MetadataValue**. This allows the user to make the change in one place vs changing it at each table cell. The supported values are:
  - **Black**
  - **Gray**
  - **LightGray**
  - **User defined value** - If the color needed is not listed as a standard option the user can enter an RGB color in the following format (rgb must be lowercase): `rgb(0,0,0)`

## Duplicate an Instance

The best practice to create an instance based on an existing instance is to duplicate the specific element or node that requires the new instance. The **Duplicate** feature (Edit menu > Duplicate) allows the author to create a copy, or clone, of a selected node and place it as a sibling with the Outline Editor. The object contents, actions, triggers and properties are all copied to the duplicate as well as any child elements or attributes. This is an optimal function when creating an instance.

Using this feature, all of the values (Name, Output Name, Profile values and Substitution values) for the instance you have selected are copied and a new instance with the same name followed by (2) is

created using the original (copied) values. You can then rename the new instance and modify the values as needed.

## About References

This section explains how to create different types of references in a requirements file.

- Document References
- Content References
- Internal References (Xref)

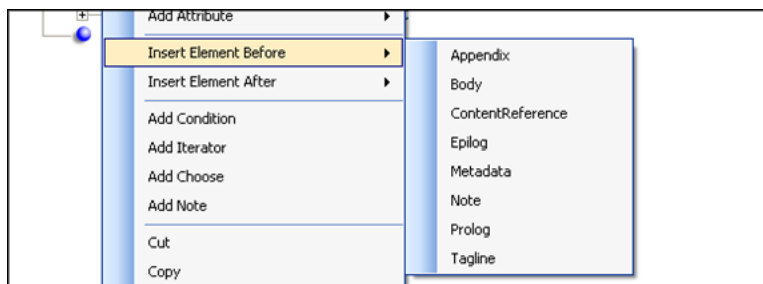
### About Content References

A Content Reference element allows the author to reuse content from another document, or within the same document. Taglines are set up as Content References and other documents have the same content, allowing the content to be authored once and reused on subsequent documents.

**Note:**

The Document ID can not be changed after content references have been added to the document.

To create a Content Reference, verify that both documents are checked out through TFS (the document that is being authored as well as the source document). Right click to add the Content Reference element in the appropriate location.



1. Click **Yes** in the resulting dialogs.
2. Navigate to the location where you saved a copy of the Content Reference source file (REQ). The tool will require you to navigate to that working folder.
3. Once you select the source REQ file, a list of all the **Sections** and **Notices** in the referenced document will appear. Select the content you need and click **OK**. Multiple **Sections** or **Notices** can be selected by using the **Ctrl** and **Shift** keys. If multiple **Sections** or **Notices** are selected they will be inserted into the new REQ file in the same order that they appear in the source REQ file.
4. The tool will prompt you to verify your selection; click **OK** to continue.

**Note:**

The content will be brought into your file and appear with blue highlighting. If future changes are needed to the content, the changes must be made in the source document and saved. When the new document is opened after changes have been made to the source document the content references will be updated automatically. This keeps the two documents synchronized.

The ContentReference appears in Yellow when the referring document is opened; the validation of the ContentReference is Pending, signifying that the tool is verifying with the database that the ContentReference is current. If the tool is unable to contact the database, the ContentReference remains Yellow. If the tool is able to contact the database and determine if the content is current, the ContentReference appears in blue for Up to Date.

Here is a complete list of the ContentReference colors:

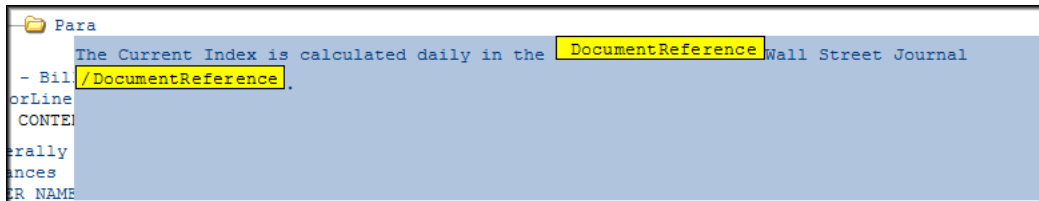
- Blue: connected to authoring DB and matches what is in the authoring DB
- Red: connected to authoring DB but does NOT match what is in the authoring DB
- Orange: (new color) connected to authoring DB but cannot find the reference in the authoring DB
- Yellow: not connected to Authoring DB

Content references are supported as elements that can be added to a table (through the <entry> element) as a child element. Adding a content reference tag (<ContentReference>) is supported in the list of elements permitted within a table structure. Specifically, the tag is permitted within the <entry> tag used to populate a table cell. The content reference tag is available in the Add Child Elements menu.

## Document References

Use document reference tags for text that refers to an external document, such as the Wall Street Journal. Apply the tag only to the first occurrence of the reference. The text inside the tags is formatted in bold italics, for example: **Wall Street Journal**.

When you open an editing box, document references appear with yellow tags, see the graphic below.

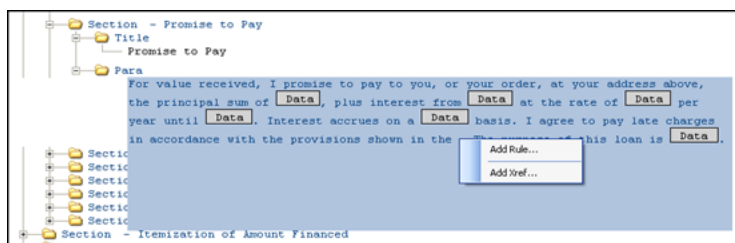


## About Internal References (XREF)

Use an **Xref** to reference sections or identified targets within the same document. The reference can be by section number, section title, both number and title, or page number where the section begins. To insert an **Xref**, right click at the location in the text where the reference needs to be added. A window will open with options for **Add Rule** and **Add Xref**. Click on **Add Xref**.

**Note:**

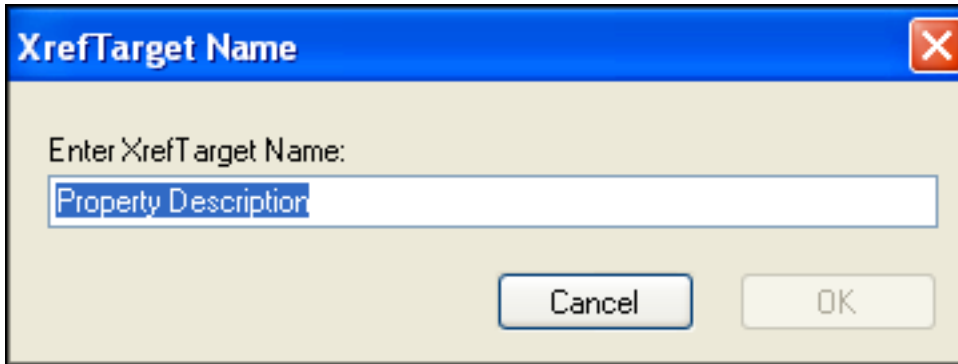
Note that this is a new feature and Expere content uses the **Document Reference** tag for internal and external content.



The Xref now appears in the content as **Xref – Unlinked**.

1. To point to the section to reference content, right click on the **Xref** element and choose **Link**. . . The list of sections populates from the name provided when authoring the section (not the actual title, but the name that appears to the right of the section element). The name of the section will now appear in the Outline Editor view.
2. Select the “style” for the reference. These options appear when right-click on the Xref:
  - **Title** - Will replace the Xref with the actual Section Title element contents during publishing (not necessarily the same as the Section Name). The word ‘section’ will not be added so it should be authored in the content if needed.
  - **Number** - Will replace the Xref with the Section number during publishing.
  - **Number and Title** - Will replace the Xref with both the section number and the title during publishing.
  - **Section Page Number** - Will replace the Xref with the page number that the linked Section Name appears on. For example: The Property Description appears on page ##.
  - **Last Page Number** - Will replace the Xref with the page number of the last page. No Link is necessary for this option.
  - **Current Page Number** - Will replace the Xref with the current page number. No Link is necessary for this option.
  - **Delete Xref** - will remove the Xref element from the content.

1. Use an **XrefTarget** to reference content at a more specific place within a section (for example one sentence or a data item).
2. To create an XrefTarget: select the text and right click; select **Wrap in ‘XrefTarget’ Tag** from the list. The **XrefTargetName** prompt appears; enter an XrefTarget Name.



3. Click **OK**; the Xref now appears in the Editor and will also be available in the **Link. . .** list described above for a standard Xref. This option is typically used if you need to ensure that the page number that is produced by the Xref is the actual page where that content appears.

## Product Technical Rule (PTR) Reference

In order for Requirements Editor to generate the code for documents, users must format the Product Technical Rules (PTRs) correctly and use the defined Keywords within the tool. PTRs are written for Conditions (inclusion rules), Data Elements, Packages, and Iterators.

### Key Words

**IF:** An IF would be used in each condition to indicate the criteria for the condition. IFs are valid within any PTR (other than the instance iterator) and each IF must have a THEN and an ENDIF. A space is always needed after an IF. IFs can use the following operators:

```
= (Equal)
!= (Not Equal)
> (Greater Than)
< (Less Than)
>= (Greater Than or Equal)
<= (Less Than or Equal)
```

**THEN:** Each IF has a corresponding THEN to indicate what should happen when the IF criteria is met. A space is always needed before a THEN and a return is used after a THEN.

**ENDIF:** Each IF has a corresponding ENDIF to indicate where the IF logic terminates. A rule with multiple IF statements would have multiple corresponding ENDIFs.

```
IF /LOAN/_CLOSING_DOCUMENTS/LOAN_DETAILS/@DocumentsDrawnInNameType = 'Lender'
THEN
  PRINT /LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress ", "
  IF string(/LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress2) != "" THEN
    PRINT ", " /LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress2
  ENDIF
ENDIF
```

**INCLUDE:** A condition always ends with INCLUDE. The condition will say IF, THEN, and end with INCLUDE to indicate that if the corresponding condition is met then any content that is a child of that condition would be included.

**CHECKBOX:** For a Field that is a checkbox (defined in the Output Format), the PTR would end with THEN CHECKBOX.

**SELECT:** For the Package Rule, the PTR would end with THEN SELECT. See Package Rule section for more information.

**PRINT:** A data element always includes at least one PRINT statement. Within a data PTR any text to print other than schema paths must be in quotes, including but not limited to spaces and commas. For example:

```
PRINT /Txn/OrgCity ." " /Txn/OrgState " " /Txn/OrgPostalCode
```

The quotes are needed for the comma and space after city as well as the space before postal code.

**ELSEIF:** For a series of possible criteria within a rule, an IF is used for the first criteria and an ELSEIF is used if there is an alternative to the IF that requires its own criteria. For example:

```
IF /Txn/OrgState = 'MN' THEN
  PRINT "Apples"
ELSEIF /Txn/OrgState = 'OK' THEN
  PRINT "Bananas"
ENDIF
```

**ELSE:** Once a rule has an IF, if there is an alternative rule that should apply any time the IF (as well as any ELSEIFs) does not apply then an ELSE can be used. For Example:

```
IF /Txn/OrgState = 'MN' THEN
  PRINT "Apples"
ELSEIF /Txn/OrgState = 'OK' THEN
  PRINT "Bananas"
ELSE
  PRINT "Pears"
ENDIF
```

In the above rule, “Pears” would print any time the OrgState was not MN or OK.

**or:** or can be used within a PTR to create alternative rules. **Or’s** should always be in lower case in the rule. A space is needed before and after the or. For example:

```
IF ./OrgResponsibilityType = '1' or string(./SelfDirectedInd) != '1' THEN
  INCLUDE
ENDIF
```

In the rule above, the content that is a child of the condition would be included when a transaction met either of the two criteria.

**and:** and can be used within a PTR to require multiple criteria to be met. And’s should always be in lower case in the rule. A space is needed before and after the and. For example:

```
IF ./OrgResponsibilityType = '1' and string(./SelfDirectedInd) != '1' THEN
  INCLUDE
ENDIF
```

In the rule above, the content that is a child of the condition would be included when a transaction met both of the two.

**[]:** Brackets are used for rules nested within an XPath. The most common use of brackets is when the code is looking at a repeatable collection/container and we need to identify which instance of that repeatable item to use. For example, there can be multiple Accounts in a Deposit transaction. If a rule needs to look for an Account that has a Type of Checking (1) the XPath would look like this:

```
/Txn/Deposit/Accounts/Account[AccountType = '1']
```

**()**: Parenthesis are used to group logic together. Much like in math problems, if there are multiple or's and and's ( ) are used to indicate which or's and and's go together.

For example, let's start with a rule with no ( ):

```
IF /Txn/OrgState = 'MN' or /Txn/OrgState = 'OK' or /Txn/OrgState = 'AK' and /Txn/OrgCharter = 'Federal' THEN
```

Without any ( ) the last criteria for OrgCharter will only be evaluated if OrgState is AK. In order to have OrgCharter considered with all of the states ( ) would be needed around the states:

```
IF (/Txn/OrgState = 'MN' or /Txn/OrgState = 'OK' or /Txn/OrgState = 'AK') and /Txn/OrgCharter = 'Federal' THEN
```

**Note:**

Any time there are multiple or's and and's in a rule it is best to use ( ) to group them to avoid confusion.

**Quotes:** Double Quotes (" ") are used around anything to Print within a Data Element other than a schema path. For example, printing a comma or a space in an address:

```
IF /LOAN/_CLOSING_DOCUMENTS/LOAN_DETAILS/@DocumentsDrawnInNameType = 'Lender' THEN
  PRINT /LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress ", "
  IF string(/LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress2) != "" THEN
    PRINT ", " /LOAN/_CLOSING_DOCUMENTS/LENDER/@_StreetAddress2
  ENDIF
ENDIF
```

**Single quotes** (' ') are used around specific enumerated values being checked in a rule ('Lender' in the rule above).

**String:** When evaluating Enumeration data points to be = "" (nothing) or != to certain values a String function must be added around the data point. A String function starts with the word String and is followed by an XPath in ( ). For example, if rule is entered as IF /Txn/OrgState != 'MN', the tool will convert it to IF String(/Txn/OrgState) != 'MN'. Rules looking for an Enumeration data point = to a certain value do not need the string function (for example IF /Txn/OrgState = 'MN' does not need String).

**Count:** A rule can be written to look at the number of instances of a given data point when the data point is in a repeatable collection/container. In order to do this the Count function is used. A count function always starts with the word "Count" and is followed by the XPath for the data point being counted in ( ). For example:

```
IF Count(/Txn/Deposit/IRAPPlans/IRAPlan/Fees/Fee[Type='99']) > 0 THEN
  INCLUDE
ENDIF
```

The above rule would return as True if there was at least one Fee in the transaction with a Type of 99.

## Assembling Documents

Within the Requirements Editor tool, you can assemble an REQ file to verify the structure (validity) of an REQ as well as execute and REQ file to assemble a PDF output of the file and create a document (PDF) based on a particular data set. You can assemble an REQ file through in the *Assembly* menu.

The following section discusses document assembly and the actions that can be taken related to assembling documents.

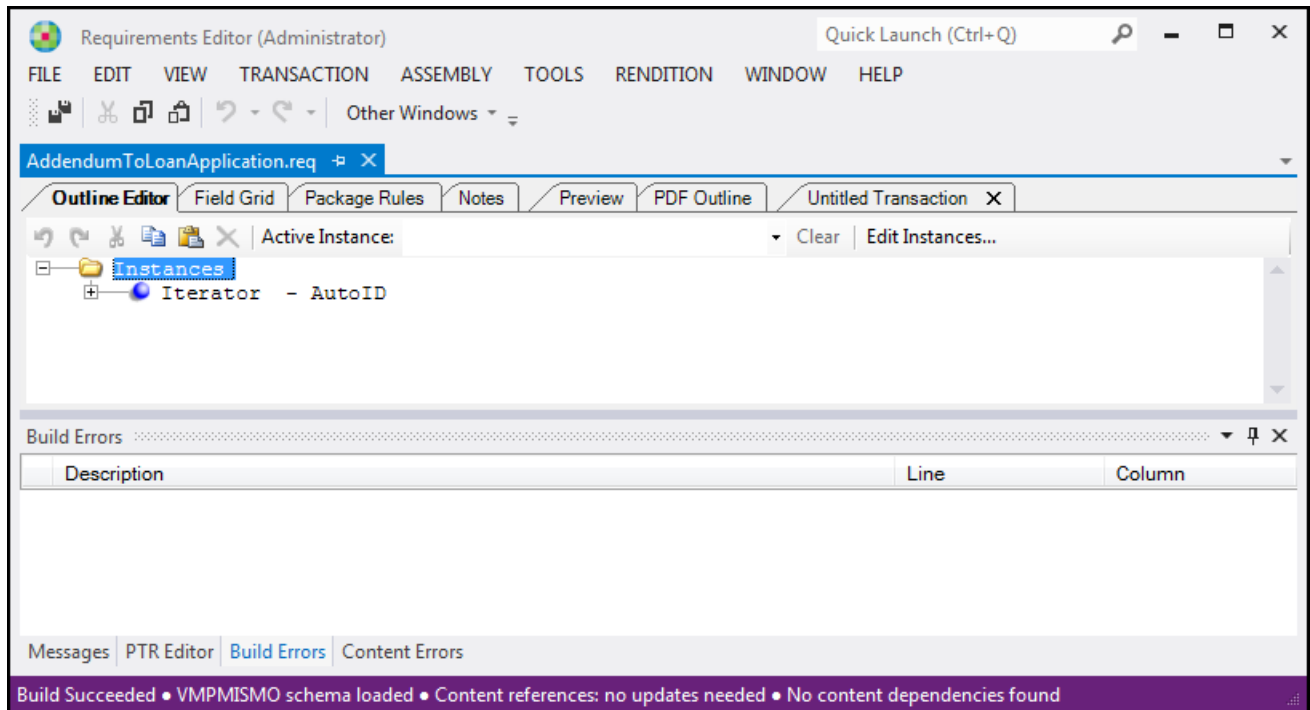
- About Document Assembly
- Assembling (Building) a Document
- Executing a Document
- Executing Document Package Rules

### About Document Assembly

#### Assembling (Building) a Document

The *Build* option is used to verify the validity of the source XML (REQ file) and perform troubleshooting by searching for errors or inactive rules within an REQ file. This feature assembles a single REQ file and you must have an REQ file open in the Requirements Editor to enable the option.

On the **Assembly** menu, click **Build**. The system then processes the REQ file and provides a status message in the status bar at the bottom of the screen. Any errors will appear in either the Build Error List or Content Error List views. For example, a successfully assembled document will read something like this in the status bar:



**Note:**

Make sure you save the REQ file before running the Build. Also, all attributes must be declared in the source XML of the REQ file (for example, the attribute "ID").

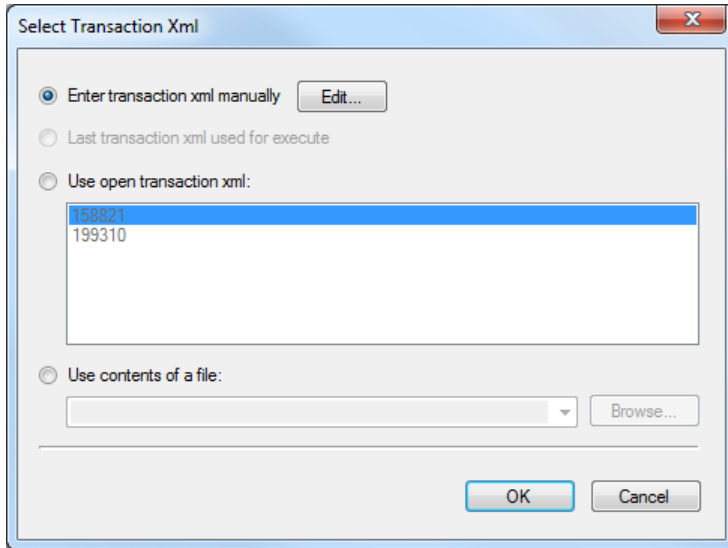
## Executing a Document

The *Execute Document* feature is used to assemble (or generate) a PDF output of the REQ file for testing purposes. This is used primarily to evaluate rules and create a document based on a particular data set. When using this feature you must define the source of the transaction data to use in creating the PDF output.

On the **Assembly** menu, click **Execute Document**. Next, select the source of the transaction data from the list of available options.

- **Enter transaction xml manually**
- **Last transaction xml used for execute**
- **Use open transaction xml** - Select this option to use an embedded external or internal transaction.
- **Use contents of a file** - Select this option to use an external transaction file.

The **Select Transaction XML** dialog box, opened when you select the *Execute Document* option, appears as follows:



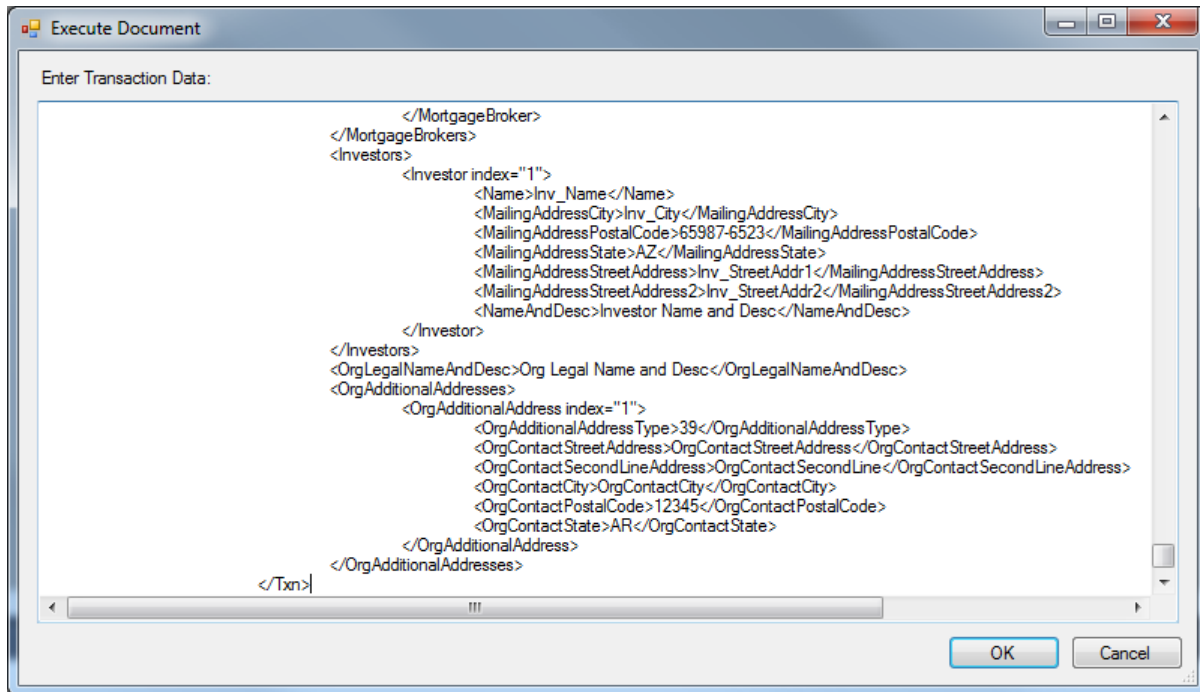
**Note:**

The **Last transaction xml used for execute** option is only available if you have previously executed a document.

Once the transaction data source is defined, the system then processes the REQ file and creates a PDF output. The resulting PDF(s) automatically open and can be reviewed, printed, and saved as a separate file.

### Entering Transaction XML Manually

When selecting the **Enter transaction xml manually** option on the Select Transaction XML dialog, the **Edit** button is enabled, allowing you to display the Execute Document window. Paste your transaction data directly in the window and click **OK**.



**Note:**

When using the **Edit** button to paste in a transaction, you also have the ability to make changes to the transaction XML prior to executing the document.

When returned to the **Select Transaction XML** dialog box, click **OK** to execute the document.

## Executing Document Package Rules

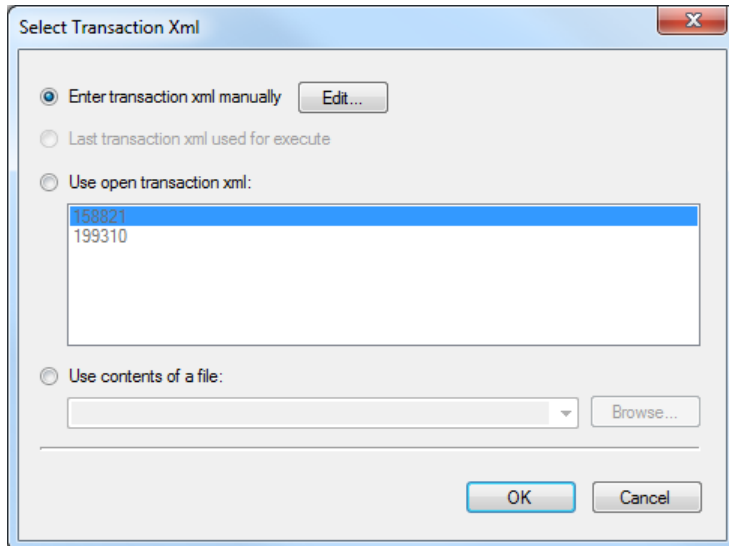
The *Execute Package Rules* option is used to assemble (or generate) a PDF output of the package rules associated with an REQ file (and transaction data) that contains a list of errors and additional information for testing purposes organized by document package rules.

Each document belongs to one or more packages, which is typically the same as the “line of business” and a phase. Packages and phases should be predetermined prior to the start of a release. Package rules allow you to select the package and phase, then have selection rules and/or suggestion rules as to when the document would be included in the package and phase. The selection and suggestion rules both have Business Logic and Product Technical Rules.

On the **Assembly** menu, click **Execute Document**. Next, select the source of the transaction data from the list of available options.

- **Enter transaction xml manually**
- **Last transaction xml used for execute**
- **Use open transaction xml** - Select this option to use an embedded external or internal transaction.
- **Use contents of a file** - Select this option to use an external transaction file.

The **Select Transaction XML** dialog box, opened when you select the *Execute Document* option, appears as follows:



Once the transaction data source is defined, the system then processes the REQ file and creates a PDF output. The resulting PDF(s) automatically open and can be reviewed, printed, and saved as a separate file.

## Working with Transactions

Within the Requirements Editor tool, you can import existing XML transaction files to modify as well as create new XML transaction files from scratch. You can create an external transaction file through the **File > New** menu or work with transactions for an open document in the **Transaction** menu.

The following section discusses transactions and the actions that can be taken related to transactions.

- About Transactions and Transaction Data
- Authoring Transaction Data
- Creating Schema-based Transactions
- Creating Document-based Transactions
  - Creating Document Rule-based Transactions
  - Creating Package Rule-based Transactions
- Importing Transactions
- Saving Transactions
- Renaming Transactions
- Validating Transactions

### Creating a new transaction based on read-only REQ files

Users can create a new external transaction based on a read-only REQ by either of the following methods:

1. Open a read-only REQ file and select **File > New**. Select **Expere External Transactions** and select the appropriate schema for your transaction type (see [Creating Schema-based Transactions](#)); or
2. Click **Transaction >** and either **Create transaction from document rules** or **Create transaction from package rules**. Save the transaction file.

## About Transactions and Transaction Data

The business rules that govern the selection and assembly of documents are driven by data. This data is supplied to the selection and assembly facility as an XML document referred to as the transaction data. Transaction data may be supplemented with values that are common across many transactions. Common data values may be delivered with the transaction data or may be determined by a policy rules program that modifies the transaction data. Policy rules are driven by transaction characteristics, such as the applicable law state or the type of loan and can also insert data values unconditionally. The policy program is optional and can be ignored if all values are passed in the transaction data. When the data is used for document selection and assembly it must be in a single XML document and that document must be valid according to a transaction schema.

If a document is authored to use data from a particular transaction schema it cannot use data from a different transaction schema. Therefore, data must be transformed to the schema used by the documents of interest. Documents are collected by line of business. In the current Expere IE release, documents for

the Commercial, Consumer Direct, Deposit, IRA, HSA and Home Equity are authored to use data from the ExpereTxn schema and documents for the Mortgage line of business are authored to use data from the VMPMISMO schema. While ExpereTxn is the preferred authoring schema, documents created in the future for other lines of business may employ ExpereTxn, VMPMISMO, or another schema.

**Note:**

See the Transaction folder on the content CD for sample lending transactions

## Authoring Transaction Data

A transaction based on document rules is used to create an empty set of transaction data containing all data points reference in the document PTRs. Once created, you can edit the transaction by entering data into a particular data element. This feature is used in part to assist in transaction testing and, as a best practice, the data (values) entered into a data element should represent a realistic example of actual transaction data as opposed to nonsense data.

Once authored, you will want to verify all necessary data is entered to test the document or section of the document. This involves both verifying the data needed to create an instance of the document as well as the data for the logic being tested.

Consider the following guidelines when authoring transaction data.

- TxnString: an alpha-numeric string
- Enumerations/Boolean: select an option from drop-down list. If selecting True, a 1 is inserted as the value. For False, a 0 is inserted.
- TxnMoney: only numeric values up to 2 decimal places (decimal places are not required).
- TxnInteger: only a whole number.
- TxnDate: enter in the format YYYYMMDD.
- TxnRef: These are just identifiers used to tie different collections/containers together, valid values should include both a number and a letter. For example, each Reg E Policy has an ID and each Deposit Account has a Reg E Policy ID Ref so that the applicable Reg E Policy can be tied to the Account. Both of these data points are defined as TxnRef, and to tie a given policy to an account the same value would be entered for the Reg E Policy ID and the Account's Reg E Policy ID Ref.

All paths contained in the document (REQ file) should appear in transaction editor. You can, however, add additional data points not already in the transaction. For example, if you have an existing transaction file (XML) and the REQ file changes to use additional data points you can manually add the new data points to the existing transaction XML.

To add data inputs within an existing container right click anywhere in the *Element Attributes* pane and select **Add Attribute**. A list will appear with all available data points at that location. Highlight the data point you want to add and click **Add**.

**Note:**

You can select multiple data points at the same time by holding Control while you select multiple and then click Add.

To add an entire new collection/container right click on the element that the collection is a child of in the *Transaction Element* pane, then select **Add Child Element** and select the element you wish to add.

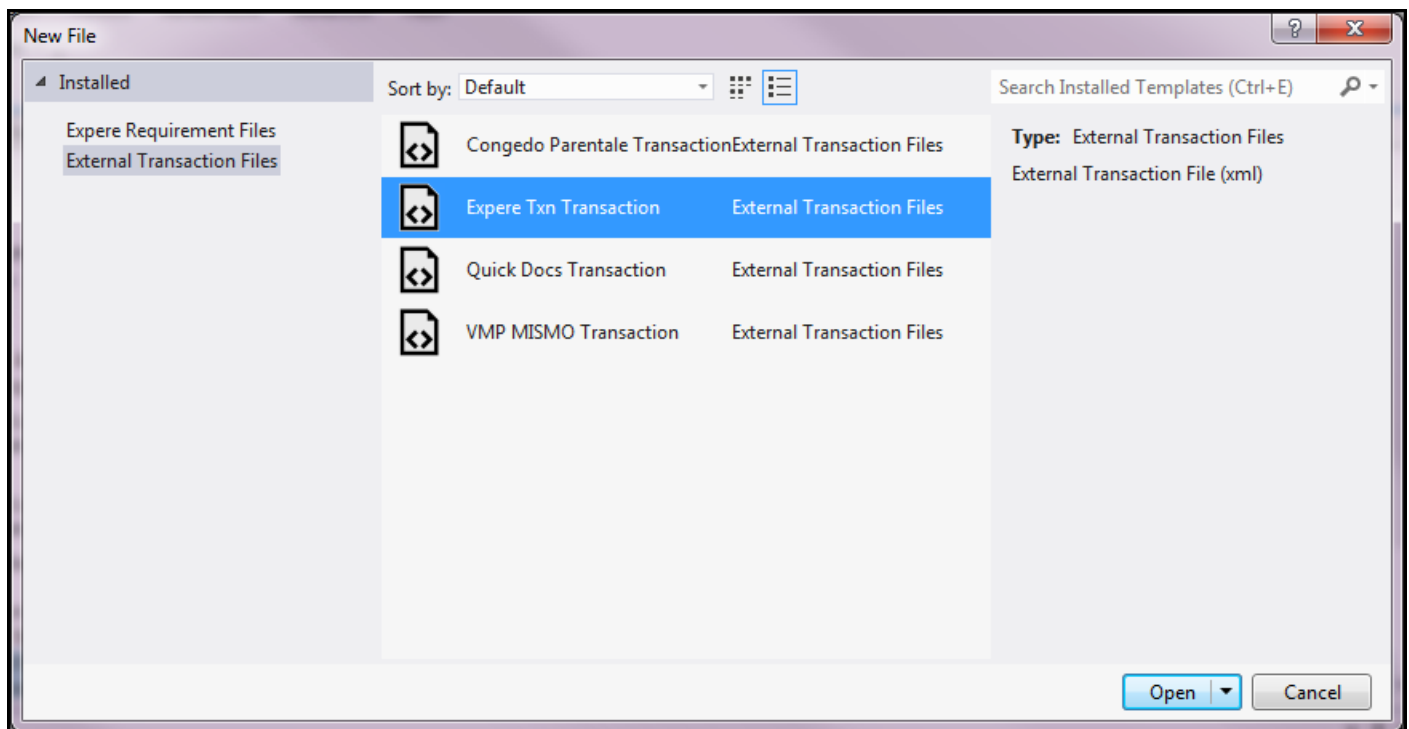
## Creating Schema-based Transactions

Transactions can be created independent of a document based on a schema. In contrast to creating a transaction based on a document (REQ), this method creates a clean external transaction file based on the defined schema. Once created, you can edit the transaction by entering data into a particular data element.

**Note:**

Users can also create a new external transaction file when opening a read-only REQ file.

In the **File** menu, select **New**. In the **New File** dialog box, select **External Transaction Files** and select the transaction type based on the underlying schema. This feature creates an external transaction file and opens the file in the transaction editor.



## Creating Transactions Associated with a Document

Transactions can be created based on either the document rules or package rules associated with the document (REQ) from the Transaction menu. In both cases, you have the option of embedding the transaction data within the associated REQ file or creating an external transaction as a separate XML file.

### About Embedded Transactions

Creating transaction data embedded in the REQ file writes the transaction data to the REQ and presents the data as a tabbed view, as an untitled transaction, in the REQ tab group view.

**Note:**

In the *Transaction* menu, click **Change Transaction Name** to enter a meaningful name for the transaction data.

When using an external transaction, the available options in the Transactions menu include:

- **Save Transaction As:** Choose this option to save the embedded transaction as an external transaction.
- **Change Transaction Name:** Choose the option to change the name of the embedded transaction (the name is presented as the tab title).
- **Validate Transaction:** Choose this option to run validation on the transaction.

### About External Transactions

Creating an external transaction writes the transaction data to the separate file that is saved independently from the REQ files from which it was generated. The output of this option is a unique file containing the transaction data associated with the REQ that is saved to the folder of your choosing on your local drive (or on a shared drive). Once saved, the transaction is presented as a document, as *Transaction Name.xml*, in the document view of the editor.

When using an external transaction, the available options in the Transactions menu include:

- **Validate Transaction:** Choose this option to run validation on the transaction.

## Creating Document Rule-based Transactions

A transaction based on document rules is used to create an empty set of transaction data containing all data points reference in the document PTRs. Once created, you can edit the transaction by entering data into a particular data element.

With the target REQ file open, in the **Transaction** menu select **Create Transaction from Document Rules**. You can elect to create the transaction as an embedded transaction or an external transaction. Once created, the transaction editor opens. From this point you can begin authoring and testing of the transaction data.

**Note:**

When viewing a Read-Only REQ file, users can create a new external transaction (see [Creating Schema-based Transactions](#)) or open and use an existing external transaction (**File > Open**).

## Creating Package Rule-based Transactions

A transaction based on package rules is used to create an empty set of transaction data containing all package rules reference in the document PTRs. Once created, you can edit the transaction by entering data into a particular data element.

With the target REQ file open, in the Transaction menu select **Create Transaction from Package Rules**. You can elect to create the transaction as an embedded transaction or an external transaction.

Once created, the transaction editor opens. From this point you can begin authoring and testing of the transaction data.

**Note:**

When viewing a Read-Only REQ file, users can create a new external transaction (see [Creating Schema-based Transactions](#)) or open and use an existing external transaction (**File > Open**).

## Importing Transactions

External transactions can be imported as an embedded transaction within an REQ file. This is an effective means of taking an external transaction that was modified or otherwise worked on outside the REQ file and associating it with the REQ file direction as an embedded transaction.

**Note:**

Transaction data imported into the tool will save as an embedded transaction along with the REQ file. Often though, you may want to save the XML File separately with another name (for example, if you import an XML file and modify it as a new transaction). To save an embedded transaction as an external transaction; in the *Transaction* menu, select **Save Transaction As** and provide the new name and location.

To import an external transaction, use the **Import Transaction** option in the *Transaction* menu. This option opens an *Open* dialog box where you can select a transaction from a folder. The imported transaction is then converted to an embedded transaction associated with an REQ file.

## Saving Transactions

The process for saving embedded and external transactions is slightly different in terms of the menu options used. The **Save Transaction As** option allows you to save an embedded transaction as an external transaction. This option works similarly to the **Save As** option in the **File** menu used to save external transactions.

To save an embedded transaction as an external transaction, use the **Save Transaction As** option in the *Transaction* menu. This option opens a *Save As* dialog box where you can provide a name and destination path for the transaction. You can also select the embedded transaction tab and right-click to open a shortcut menu providing *Save As*, *Change Transaction Name*, and *Validate* options.

**Note:**

To change the name of an embedded transaction, use the *Change Transaction Name* option in the *Transaction* menu.

To save an external transaction, use the **Save** or **Save As** options in the *File* menu. An external transaction (as a separate file) is opened, closed, and saved just like an REQ file from options available in the *File* menu.

## Renaming Transactions

In certain cases, you may want to change the name of an embedded transaction. The **Change Transaction Name** option allows you to modify the name of an embedded transaction as an external transaction.

To change the name of an embedded transaction within an REQ file, use the **Change Transaction Name** option in the *Transaction* menu. This option opens a dialog box where you can provide a new name for the transaction. Once a new name is entered, the transaction opens in a new tab. You can also select the embedded transaction tab and right-click to open a shortcut menu providing *Save As*, *Change Transaction Name*, and *Validate* options.

**Note:**  
This option is not available for external transactions.

## Validating Transactions

Transaction XML can be validated by selecting **Validate Transaction** in the *Transaction* menu. The feature performs validation against the schema associated with the document whether ExpereTxn, VMP MISMO, or any other schema that the document is based upon.

The results of the validation are presented in a *Validation Errors(nn)* window displayed directly under the document/tab. In the list of errors, select an error to apply focus to the related element in the transaction view.

Requirements Editor (Administrator) Quick Launch (Ctrl+Q)

FILE EDIT VIEW TRANSACTION ASSEMBLY TOOLS RENDITION WINDOW HELP

Other Windows ▾

AddendumToLoanApplication.req\* ▾ X

Outline Editor | Field Grid | Package Rules | Notes | Preview | PDF Outline | **Untitled Transaction X**

Transaction Elements

- LOAN
  - APPLICATION
    - BORROWER**
      - MORTGAGE\_TERMS
      - PROPERTY
  - CLOSING\_DOCUMENTS
    - INVESTOR
    - LENDER
    - LOAN\_DETAILS
    - MORTGAGE\_BROKER

Element Attributes: /LOAN/\_APPLICATION/BORROWER (\*\* indicates suppressed UI)

	Value	Options/Data Type
***_First Name		String
***_Last Name		String
***_Middle Name		String
***_Name Suffix		String
***_NonObligated Indicator		
_vDomesticPartnershipDescription		String
_vDomesticPartnershipIndicator		
_vHomesteadPartyName		String
_vOtherPartyWithHomesteadIndicator		
***NonObligatedBorrowerIndicator		

Validation Errors(11):

Description	XPath
Invalid Node 'LOAN'; Node does not exist in the schema.	/LOAN
Invalid Node '_APPLICATION'; Node does not exist in the schema.	/LOAN/_APPLICATION
Invalid Node '_MAIL_TO'; Node does not exist in the schema.	/LOAN/_APPLICATION/BORROWER/_MAIL_TO
Invalid Node '_POWER_OF_ATTORNEY'; Node does not exist in the schema.	/LOAN/_APPLICATION/BORROWER/_POWER_OF_ATTORNEY
Invalid Node 'MORTGAGE_TERMS'; Node does not exist in the schema.	/LOAN/_APPLICATION/MORTGAGE_TERMS
Invalid Node 'PROPERTY'; Node does not exist in the schema.	/LOAN/_APPLICATION/PROPERTY
Invalid Node '_CLOSING_DOCUMENTS'; Node does not exist in the schema.	/LOAN/_CLOSING_DOCUMENTS
Invalid Node 'INVESTOR'; Node does not exist in the schema.	/LOAN/_CLOSING_DOCUMENTS/INVESTOR
Invalid Node 'LENDER'; Node does not exist in the schema.	/LOAN/_CLOSING_DOCUMENTS/LENDER
Invalid Node 'LOAN_DETAILS'; Node does not exist in the schema.	/LOAN/_CLOSING_DOCUMENTS/LOAN_DETAILS
Invalid Node 'MORTGAGE_BROKER'; Node does not exist in the schema.	/LOAN/_CLOSING_DOCUMENTS/MORTGAGE_BROKER

Messages

Messages | PTR Editor | Build Errors | Content Errors

Ready

## Working with the DocViewer

Conceptual topic describing DocViewer integration use cases.

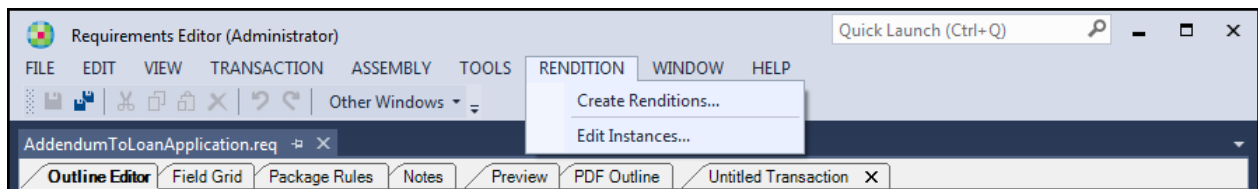
## Working with the Rendition Plug-In

The **Rendition** feature allows the author to publish Eform, Preprint and Menu (EPM) products from within the Requirements Editor. To enable the feature in the Requirements Editor, you must install the Requirements Editor Internal version and, using the *Advanced Install* option, select the *E-Forms Rendition* plug-in.

**Note:**

The Rendition feature is not available for generating Expere REQ-based document outputs.

Once installed, the feature is available by clicking the **Create Renditions** option in the *Rendition* menu when a document is opened for editing.



## Using Overrides

This topic describes best practices for using the standard field size override feature on standard, content-driven forms (EForms and preprint forms). Changes, or overrides, to the default field size restrictions on standard content is permitted in cases where the desired layout of a form does not accommodate the field size defined on the same form. The same adjustments to custom content can be applied at the discretion of the author.

The Requirements Editor provides content authors with a means to modify the field size restrictions (width and vertical height) on a form and all instances of the document. These overrides allow you to construct a form that deviates from the standard data classification rules, or field definitions, found in the [Single Source Tools](http://nasrvtx95802/SingleSourceTools/DataClassifications/Default.aspx) application (<http://nasrvtx95802/SingleSourceTools/DataClassifications/Default.aspx>).

**Note:**

The best practice for using field overrides applies to standard content only. For custom content, the best practice can be applied but the author is not restricted by the best practice in regards to custom content authoring.

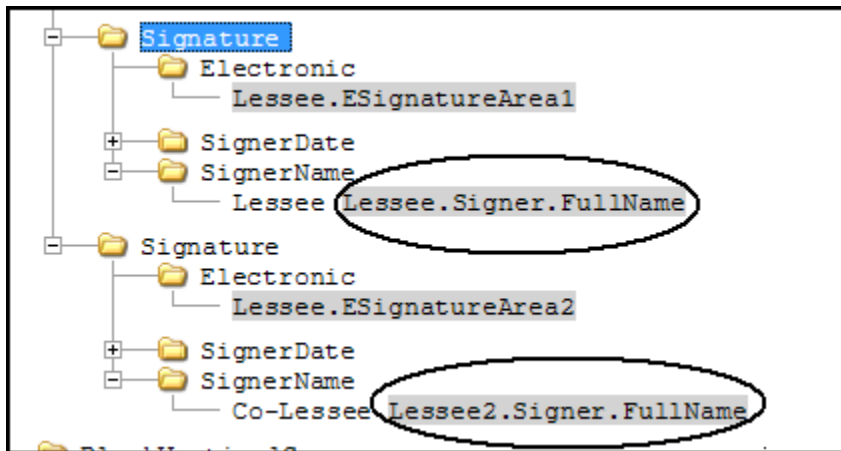
When considering using an override, the first option is to attempt to address content requirements through layout changes. If a layout change is not feasible, the override feature can be used to apply specified size values to a particular field on a form. In all cases, the use of overrides should be considered before creating new fields to accommodate size restrictions on certain forms.

The best practice for the use of overrides on a form can be summarized as:

Do not use the override option when...	A layout change in the document can accommodate the content requirements. In this case, the field in question in the flow and context of the form does not present any space limitations and a layout change can address the issue. Layout changes are typically tested through a visual check of the output document (preview or test PDF).
Use the override option when...	A layout change on the form is not feasible AND space limitations prevent you from using the standard field definition, then authors may handle the requirements through an override to fit the layout design.

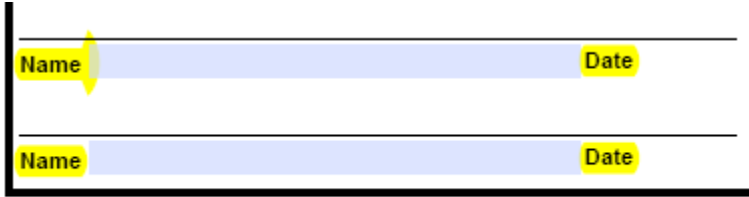
In the Requirements Editor, field overrides are implemented through the **Outline Editor** perspective within the **PTR Editor** view. With the target field selected in the Outline Editor, locate the *Class* setting in the PTR Editor toolbar. You may need to click the drop-down arrow to show the *Class* setting along with some other setting options. You can then apply the override by entering a value for the desired field size to the *Class* attribute. You can only apply one classification attribute per data name (*PublicName* or *GDDName*) and the value supplied must be one of the existing data classifications documented in the *Single Source Tools* application. Data classification values are assigned to a public name (or *GDDName*) and typically include values such as *DollarNN*, *TextNN*, *Initial*, *NumberNN*, *PercentNN*, and so on. You will find these and other examples in the *Single Source Tools* application and guidance for each defined in the *Wolters Kluwer Financial Services StyleGuide*.

In this example, an override has been defined for the **Name / Date** signature fields in a signature section. The standard form required markup changes and modification to allow a two text changes. The signature line required a change from **Name / Date** to **Lessee / Date Signed** and **Co-Lessee / Date Signed** respectively. The standard data classification for the lessee (public name=*Lessee.Signer.FullName*) and co-lessee (public name=*Lessee2.Signer.FullName*) is defined as *Text29* but in order to make the text change in the signature area we are going to override that value with a data classification of *Text24* (lessee) and *Text22* (co-lessee).



The field is assembled and the output generated consistent with the established override. In this example, the lessee and co-lessee values are now *Text24* and *Text22* respectively (an override of the default data classification value of *Text29*) to support the a width that allows the signature area text change to fit in the form layout.

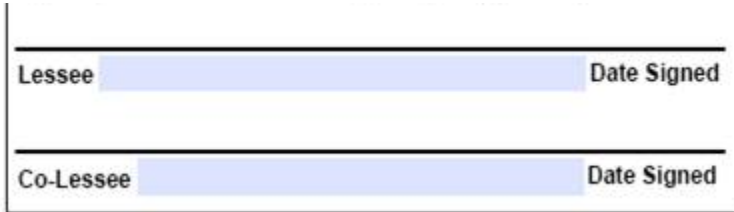
The signature area in the original form looked like this:



Name	Date
Name	Date

**Notice to Lessee.** (1) Do not sign this Lease before you read it or if it contains any blank spaces; (2) You are entitled to a completely filled-in copy of this Lease.

After the override is applied, the signature area on the form looks like:

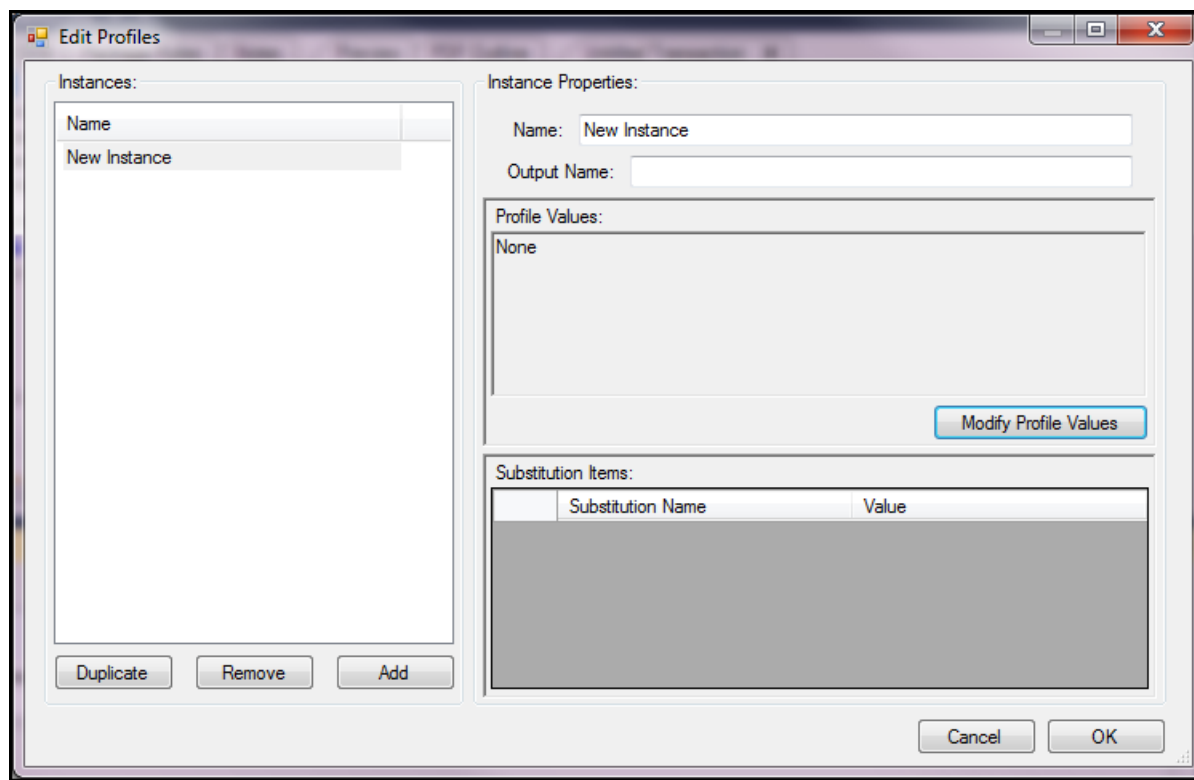


Lessee	Date Signed
Co-Lessee	Date Signed

## Creating and Editing Instances

Enter topic summary (Optional) - remove all text if not used

Choosing the **Edit Instances** option from the *Rendition* menu brings up a dialog box that allows you to add, copy, remove, and edit instances associated with a given document. In addition to the Rendition menu, you can open the instance editor by clicking the **Edit Instances** button on the Rendition toolbar.



When an EPM REQ file is created that will produce static document instances, those instances need to be defined using profile values and substitution name values. One instance definition will be created for each static document to be produced.

The instance name should be the form number, for example NDaS-SI-SD. The Output Name should be the E-Form file name if the instance will be producing an electronic form. For the preprint or paper documents the full form name should be used as the Output Name.

The profile values are set for each instance by selecting the instance you are defining from the instance list on the left (or create a new one by clicking on the Add button) and then clicking on the Modify Profile Values button in the middle of the screen.

The profile categories that have been used in the document will be displayed and you will be able to select the value you need for the instance you are defining. If you haven't used a profile that you will need to set for the instances (ex. GoverningJurisdiction) you can add it to an empty section so that it will appear in the Modify Profile Values selection. The list of values will indicate which values have been used in the outline. You can choose values that haven't been used if you are setting up the instances before the full content of the REQ file has been authored. You can scroll up or down to view additional values by clicking on the up or down arrow at the top and bottom of the list.

**Note:**

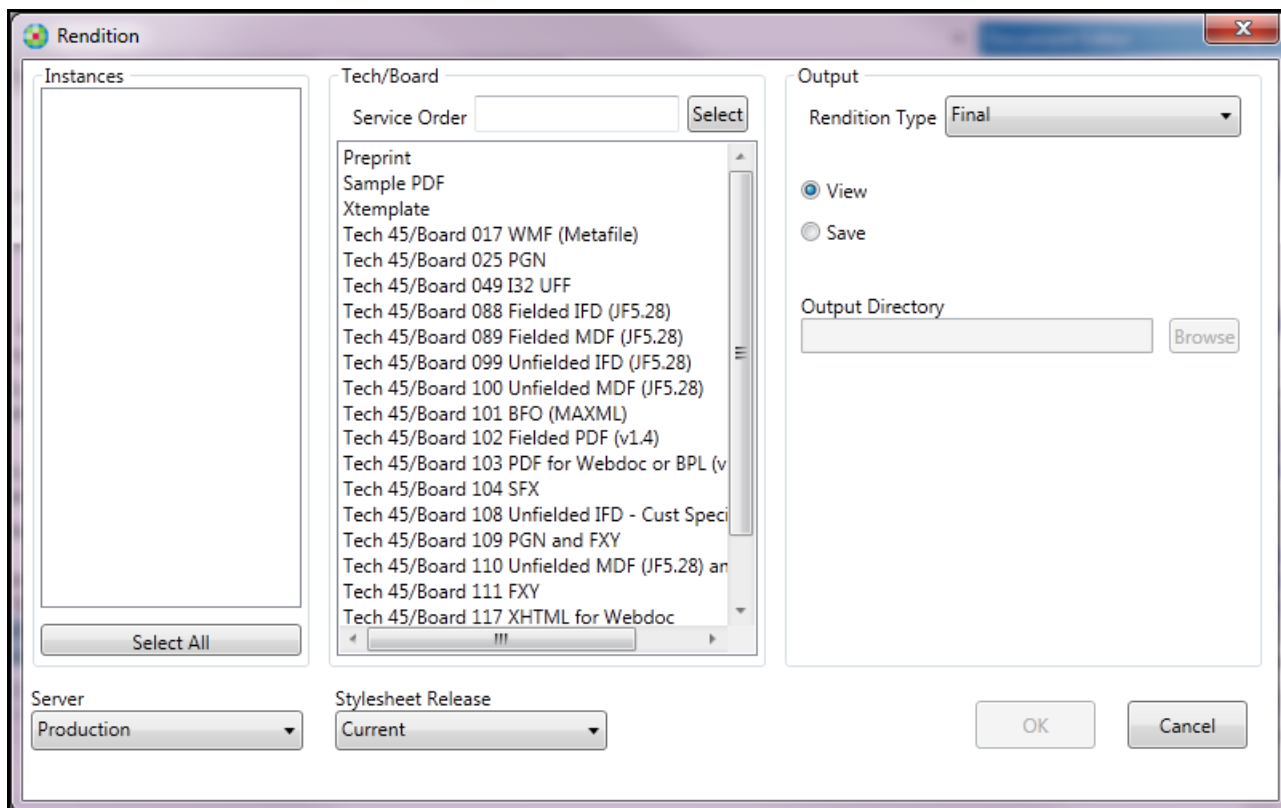
For an REQ file that will produce either an Electronic or Preprint version make sure to set the Format profile appropriately for each of the static instances.

In some cases it may be faster to create your first Instance and then use the Duplicate button found on the Edit Profiles screen below the Instances list. This option copies all of the values (Name, Output Name,

Profile values and Substitution Values) for the instance you have selected and creates a new instance with the same name followed by (2). You can rename the new instance and modify the values as needed.

## Creating a Rendition

Choosing the **Create Renditions** option from the *Rendition* menu brings up a dialog box that allows you to enter all the information necessary to produce a PDF rendition for review or to take the process all the way to completion and produce the final eform or preprint PDF that will be used for fulfillment.



In the Instances list, you can select one instance, multiple instances, or all instances of the form to produce. Select multiple instances by holding down the Ctrl key while making your selection or click Select All to select all of the instances in the list. The selection of the Tech/Board(s) indicates which format(s) the document will be produced in for final fulfillment. A Service Order number is used to determine which Tech/Boards to select based off information stored in SAP. If a Service Order number is not available or you are creating a document that doesn't require a Service Order (such as preprint documents), you can manually select the Tech/Boards required. Multiple items can be selected by holding down the Ctrl key while making your selection.

There is an option in the lower left hand corner of the window that allows you to change the Server used to produce the documents from Production (the default value) to Development. You would only change this setting to Development if you were running tests on pending changes to the stylesheets that have not yet been pushed to the Production server.

There are 4 options to choose from for Rendition Type:

- Final - used to create a PDF rendition for the Attorney/CA to review for content, style, and column or page break issues. All content appears in black.
- CustomerProof - used to create a PDF rendition for an external customer to review for approval of a Menu document. The color coding indicates what content is standard (blue) and what content is based off the customer's response to a worksheet question (pink).
- ComplianceReview - used to create a PDF rendition for the Attorney/CA to review the profiles applied to the content and the field names used. The profile applied to the content will display in black before the beginning of the content that it is applied to. The content appears in color to make it easier to differentiate between the profiles and the content.
- Transformation - used to create the final Tech/Board outputs once the PDF rendition created with one of the previous options has been approved.

You then have the option to view or save the generated product. The View option creates the files and opens them in Adobe Reader for immediately review. The Save option creates the files and saves them to a designated folder on the local computer or mapped drive.

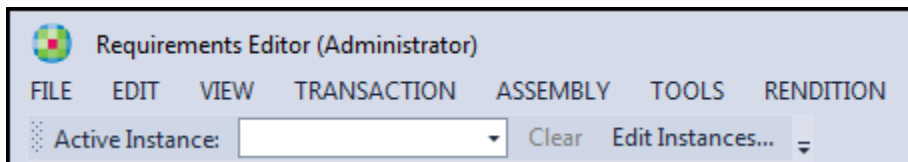
**Note:**

For the Transformation option only the PDF rendition is opened for viewing or saved to the location specified. The actual Tech/Board files are placed out on the server.

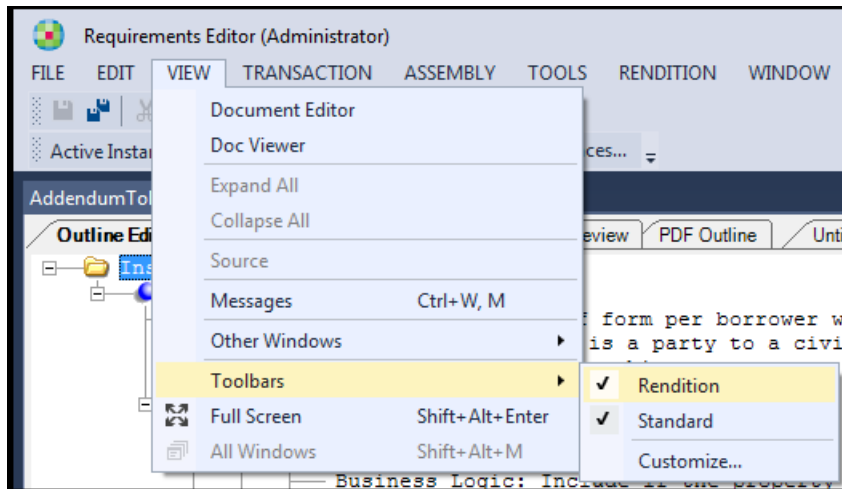
Once the rendition settings are defined, click **OK**. At this point the request is submitted for processing and a status window appears showing the progress of the request. If multiple instances were selected, each instance is displayed as it is completed in the Detail section. Once all instances are complete the Details section displays a status of *Complete*.

## Using the Rendition Toolbar

As you are authoring files using the Rendition plug-in you can enable the Rendition toolbar. The toolbar provides one-click access to select an active instance, clear the instance selector, and add or edit instances.



To enable the toolbar, in the *View* menu select *Toolbars* and click **Rendition**.



## Using Text Boxes (Boxing)

Authors can add a text box (borders) on any section within an REQ file by adding the *Boxing* attribute to a section. Boxing is intended primarily for use with notices to create a uniform standard for all borders (top, bottom, left, right) around the notice text in accordance with the *Wolters Kluwer Financial Services Standards Style Guide*.

When the Boxing option is enabled (Boxing=Yes), border widths are defined in the BoxingLineWidth attribute and applied to all four sides of the text box along with appropriate padding. The width values can be set by entering a desired value in points (1pt, 1.5pt, 2pt).

**Note:**

This option is only available when using the Rendition plug-in.

1. With a Document level element, right-click and select **Add Attribute/Boxing**.
2. Select either Yes or No; the default selection is Yes.
3. Within the element where Boxing is enabled, right-click and select **Add Attribute/BoxingLineWidth**.

## Working with the Document Editor

The **Document Editor** is a unique authoring perspective providing an editable design view of a requirements file (REQ). This perspective opens as a separate window and displays the contents of active requirements file in an author, or design, view. This perspective provides an alternative to the Outline Editor that displays a requirements files in a text, or XML tags, view. Consider this perspective as the "author" mode whereas the Outline Editor perspective is a more traditional "text" mode for XML editing.

**Note:**

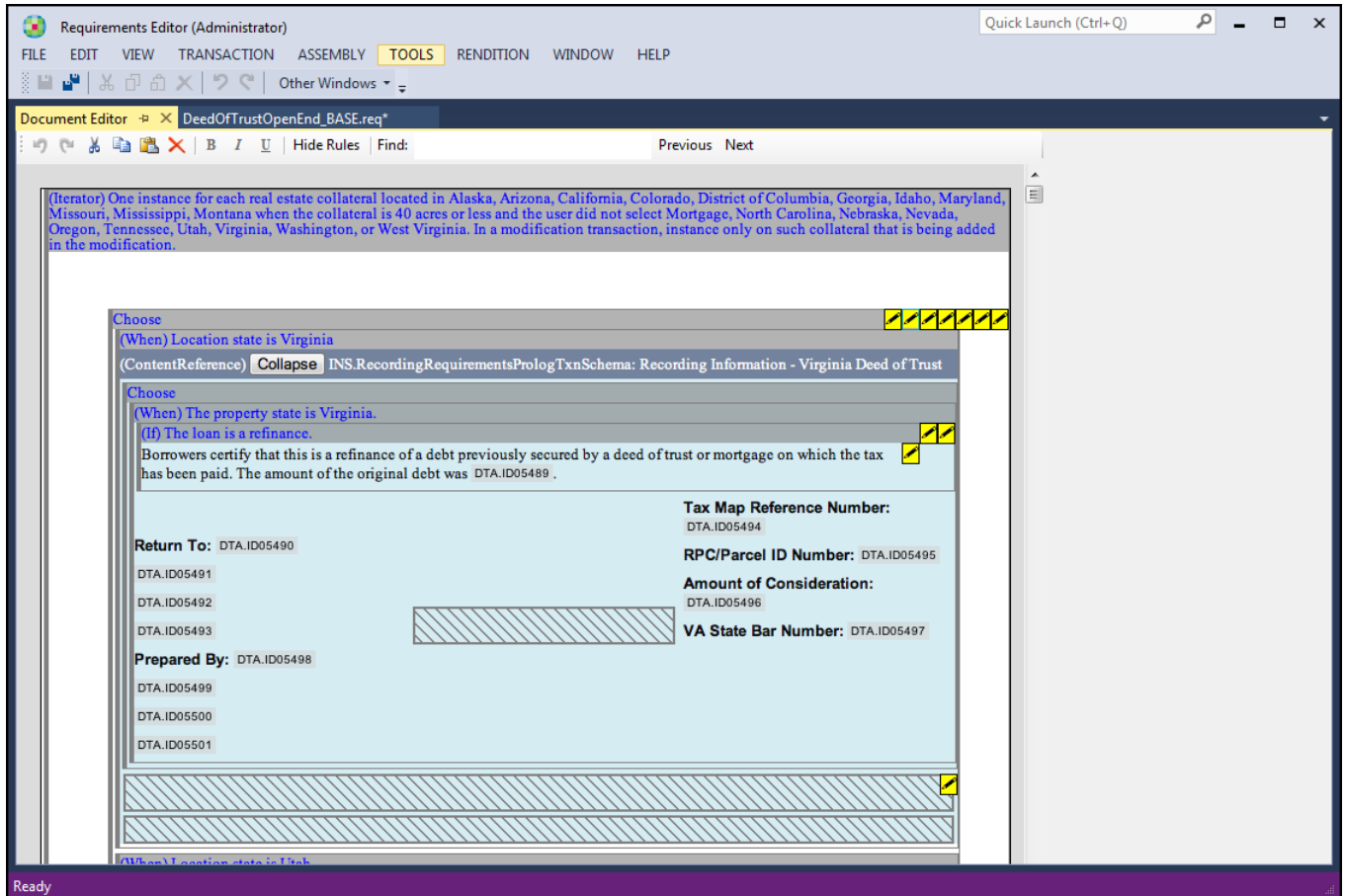
The Document Editor is installed as an optional feature as part of the *Requirements Editor advanced installation* program.

This perspective provides a more visual means to author and edit document. As you author a document in the Document Editor, the document remains available in the Outline Editor and any changes made within either perspective are synchronized to update the corresponding perspective.

**Note:**

This perspective is only available for dynamic documents.

The Document Editor perspective is available from the **View** menu when a dynamic requirements files is open in the Outline Editor perspective. To view a dynamic document in author mode; in the *View* menu, click **Document Editor**. In this example, a general *DeedOfTrust* document is displayed in the Document Editor.

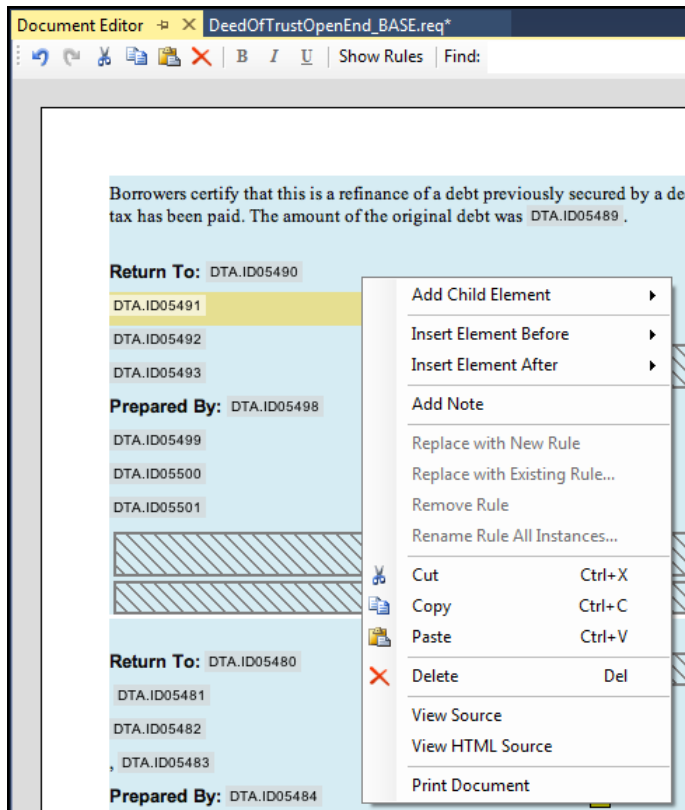


## Using the Document Editor

As mentioned, the Document Editor is a unique authoring perspective providing an editable design view of a requirements files. The Document Editor toolbar provides standard copy/cut/paste options, basic formatting options, a hide/show rules option, and a document find (search) option. The toolbar is opened when the perspective is opened; it is not available outside the Document Editor as a toolbar option.



In addition to the toolbar, the shortcut menu (available through a right-click) is available to assist in working with elements, attributes, and other authoring tasks in a similar manner as it is used in the Outline Editor. Most of the authoring functionality provided in the Outline Editor is available in the Document Editor. You can modify text, add/remove elements, define attributes, manage rules, and view the document source in both XML and HTML.



All changes made to a document in the Document Editor are saved in the requirements file (the REQ.xml) and therefore synchronized immediately in both the Document Editor and Outline Editor perspectives. In fact, when you select an element, or make any changes, in one perspective, the corresponding perspective also updates to select the active element. The author view uses color codes to indicate text sections, element tags, background, as well as to distinguish between parts of the document structure. When you click on a section in the document, the selected section becomes active and the section changes color to indicate it is active, or selected.

## Document Editor Toolbar Features OPEN

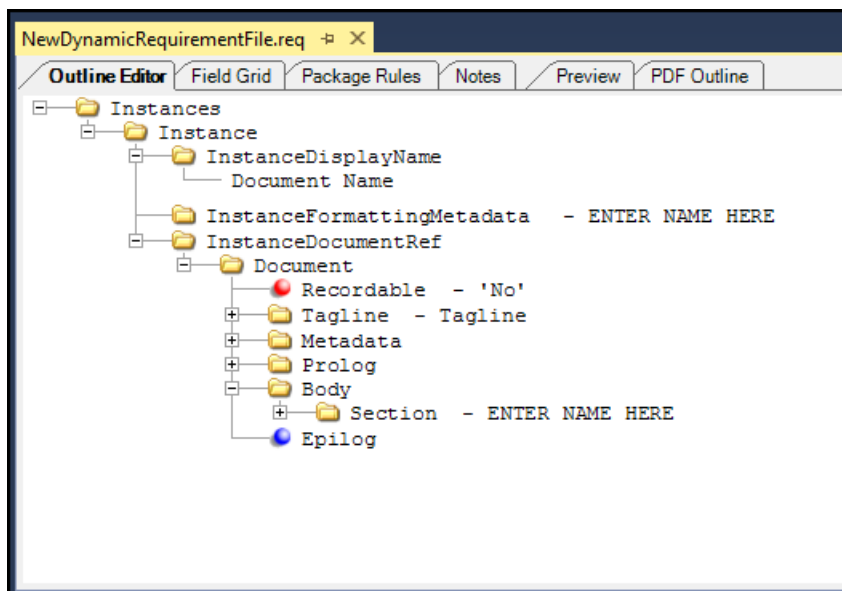
## Working with Dynamic Requirements Documents

This section describes some of the key authoring activities in creating, editing, and producing dynamic documents. For an overview of authoring dynamic documents, see [Authoring Dynamic REQ Files](#).

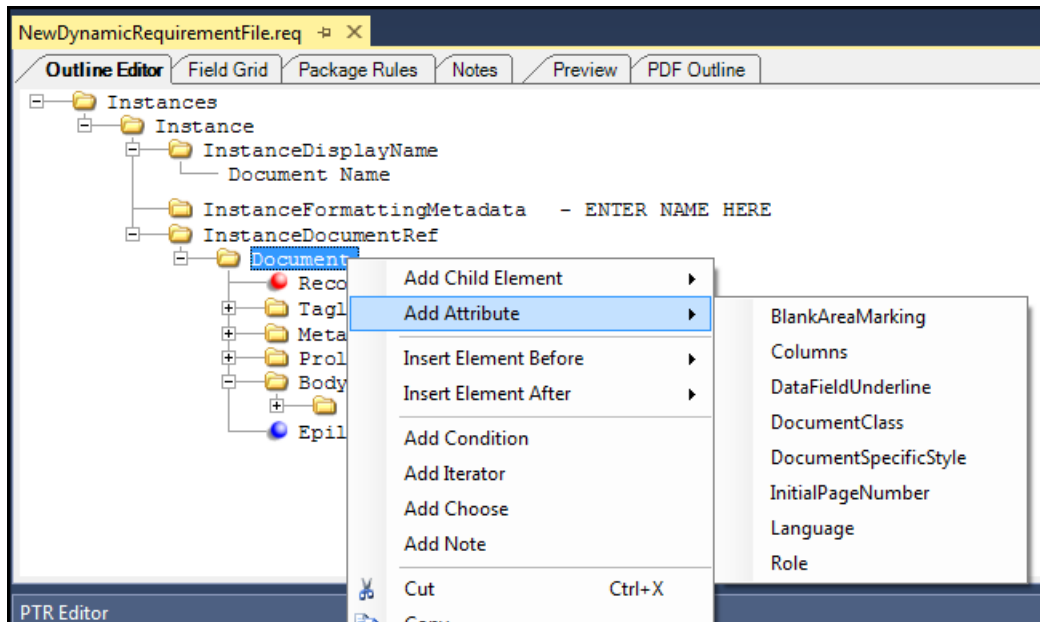
### About Dynamic Document Structure

A typical dynamic requirements file resembles the following structure.

The **Document** element is positioned inside the **InstanceDocumentRef** element which is nested with the overall instance structure of the requirements file. The **Document** element contains document content and allows a number of attributes and child elements to support authoring.

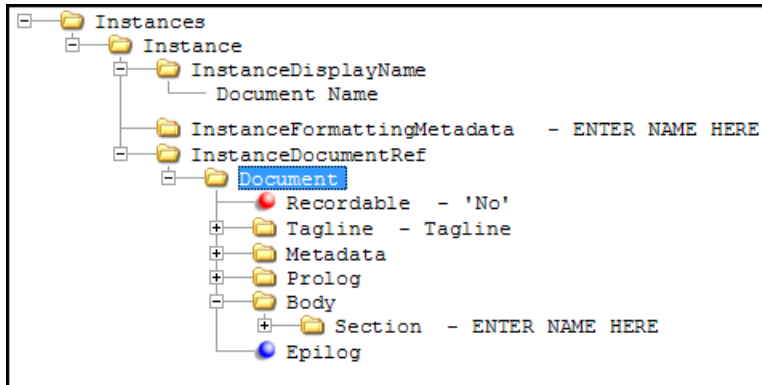


The following attributes are available on the element. To view available attributes, select the element and right-click. In the shortcut menu, select **Add Attributes**.



- **BlankAreaMarking:** Overrides the insertion of the “This area intentionally left blank.” language. The StyleGuide standard and default setting includes this language in any vertical blank space that exceeds 1.5” in the body of the document (it will not appear at the end of the document if there is more than 1.5” of blank space). Set this value to **No** if the document you are creating should not print this language. An example of when this language should not be used is on a model form where additional text is not usually allowed.
- **Columns:** Overrides the StyleGuide standard and default of two (2) columns for the document layout. By setting the value for this attribute to one (1) the document content will span the entire width of the page.
- **DataFieldUnderline:** Adds document level underlining functionality in the Requirements Editor. The attribute is used to apply underlines to all data fields within an entire document and is exclusively used with the Rendition plug-in.
- **DocumentClass:** Currently has no impact on the document.
- **DocumentSpecificStyle:** Currently used to indicate which stylesheet should be used for Non-Runtime publishing.
- **InitialPageNumber:**
- **Language:**
- **Recordable** - Determines if the document is recordable.
- **Role** - The value **SmallBody** denotes one-column format for **Document** content.

The **Document** element typically contains the following children.



- Tagline
- Note (the **Note** element must be added, it is not a default element of the **Document** node)
- Metadata
- Prolog
- Body (comprising sections)

## Iterators

This section explains how to create an iterator at the Instance Level (Instance element) in a requirements file. Iterators allow authors to repeat content multiple times for containers of a collection (for example, /Txn/Beneficiaries or /Txn/Beneficiaries/Beneficiary). Iterators can be used to print name and address for each /Txn/Beneficiaries/Beneficiary, or for each /Txn/Beneficiaries/Beneficiary[Type='4'] identified in a document. In general, the guidelines for creating an iterator include:

- An REQ file should only have one Iterator, regardless of the number of instances.
- The same node cannot be selected multiple times; users should iterate on the first Owner of the collateral that fits the requirements. The scope will be both Owner and Collateral; the PTR's underneath this Iterator must accommodate the potential for either of those.

Before creating an iterator, verify the **Iterator** is a parent to the **Instance** element and that the Business Logic within the **Iterator** explains all pertinent scenarios.

**Note:**

In the PTR for the **Iterator**, separate each scenario with a pipe ("|").

## Tagline

Authors are required to insert the tagline information into a new document; it is typically used as the footer of the document. The tagline is a common .REQ file used by all documents. The tagline file is pulled into a new document by using a *Content Reference*. Once inserted, the tagline appears as blue highlighted text.

- The tagline is typically a common .REQ file.
- The tagline file is pulled into a new document by using a Content Reference.

**Note:**

Adding this element to the document triggers the creation of a tagline on the document. The children of this element also control whether a barcode is printed on the document, what to print in the optional data fields on the tagline, whether to add page numbers to the tagline, and whether to include an initials line in the tagline.

### Tagline Separator Line

**TaglineSeparatorLine** is an attribute of the **Document** element that gives you the ability to suppress printing the Tagline separator line.

### Usage

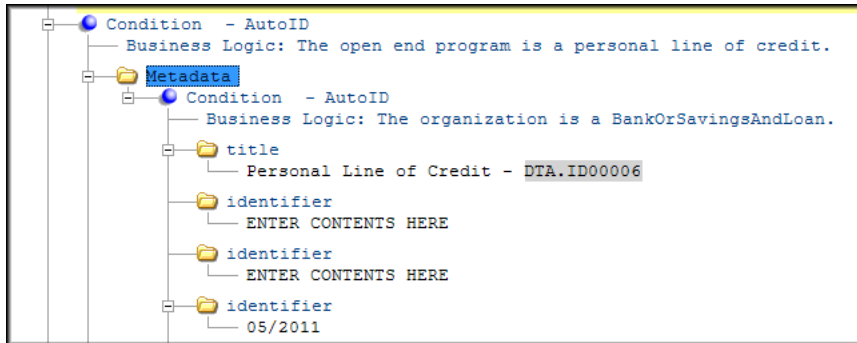
<b>Attribute Name</b>	TaglineSeparatorLine
<b>Values</b>	Yes, No - Default is Yes
<b>Logic</b>	<ul style="list-style-type: none"> <li>■ If attribute is not present -or- TaglineSeparatorLine=Yes, then continue to honor printing the tagline separator line as it does today.</li> <li>■ If TaglineSeparatorLine=No, suppress printing the tagline separator line.</li> </ul>
<b>Additional Information</b>	By default, if the Tagline and MetaData elements are not authored, the default behavior is to print a tagline separator line at bottom of each page. Also, presence of Tagline and/or MetaData elements will print a tagline separator line. In either scenario, adding TaglineSeparatorLine=No element will be honored and suppress printing of the tagline separator line.

### Metadata

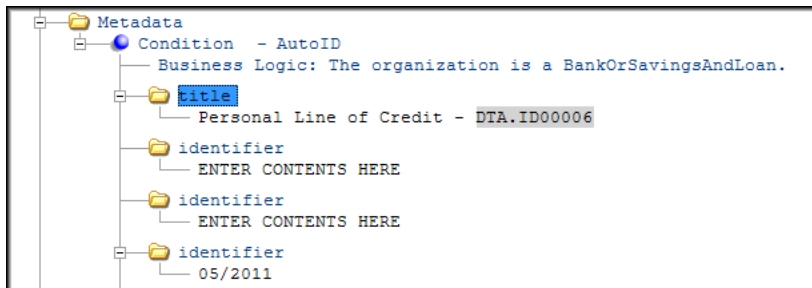
The document level metadata resides beneath the common tagline information. The Metadata element serves as a container for information about this document. Information includes publisher name, copyrights, document titles, and form numbers. Users must add the tagline information into a new document.

- The tagline is typically a common .REQ file.
- The tagline file is pulled into a new document by using a Content Reference.

You will not always have multiple titles, or conditions before the title. You must have a title, a publisher, and rights. The trademark and copyright symbols must be cut and pasted off a Word document, at this time, as you can not create these symbols in the tool.



The Title element adds a title and subtitle to a Document or Section. Add a title to a Formal Paragraph or Table. The Stylesheet produces section titles in various ways depending on the level of the section. You may only have one title that will print for each section. You may add conditions before any title element to determine which title to print.



A special kind of data item is used: a substitution item; this allows user to complete the information that changes from document to document with values entered into the instance definition. For example, the state abbreviation needs to print for the data item highlighted below in the title element.

The tagline consists of the following:

- Title followed by a dash and then a data item with a substitution name of State Abbreviation if this is a state specific document. This may not be the same as the document title. Refer to the markup to determine what should be used for the title in the tagline.
- FormInformation, which contains the child elements of FormNumber and RevisionDate is used instead of the Identifier element that appears when you create a new .REQ file. FormNumber is the form number. If the form number includes the state abbreviation the same substitution name is used that was used in the title. There may be multiple FormInformation elements. This would occur for Fannie Mae documents where we include the Fannie Form number and revision date along with the WKFS one. RevisionDate is made up of a data item with a substitution name of Revision Date or Control Date. The date will be supplied by the instance definition.

**Note:**

Fannie/Freddie documents that have an agency revision date that should be entered as part of the FormNumber.

- Publisher is usually Bankers Systems TM or VMP®, or both (cobranded).
- Rights is always Wolters Kluwer Financial Services, if we are copyrighting the document the © symbol is included followed by the first year we filed a copyright registration (current year if this

is a brand new document) and then a substitution item for Copyright Year (year the revision is being done, usually the current year).

- **HeaderFormNumber** - This information is optional and is only used currently for our Retail Sales contracts. It prints an additional form number and revision date at the top of the first page.
- **IntDiscFinancialInstitutionDefinedField** - In lieu of the spacing constraints in the tagline, this element was added so that a user can pass FI information instead of using a barcode. The user will be able to place in a maximum of 28 characters before the information becomes truncated. Requirements Editor will arrange the information to be displayed correctly in the second row of the tagline.

**Note:**

This only impacts the Integrated Disclosures documents.

- **identifier[1]** - To accommodate the maximum character count for the Loan ID on the Integrated Disclosure documents, the width of the columns have been modified for the first instance of the identifier to meet Integrated Disclosure requirements. You can place up to 30 characters into the [field] without the information flowing to the second row.

**Note:**

This only impacts the Integrated Disclosures documents.

## Prolog

The **Prolog** element is used to define the document title (DocTitle) and a table containing the parties on the document along with the document summary (descriptive information about the document ex. Loan Date, Loan Number, etc.).

The Title element allows you to add a title and subtitle to a document or a title to a section. A title can also be added to a Formal Paragraph or Table. The stylesheet produces section titles in various ways depending on the level of the section.

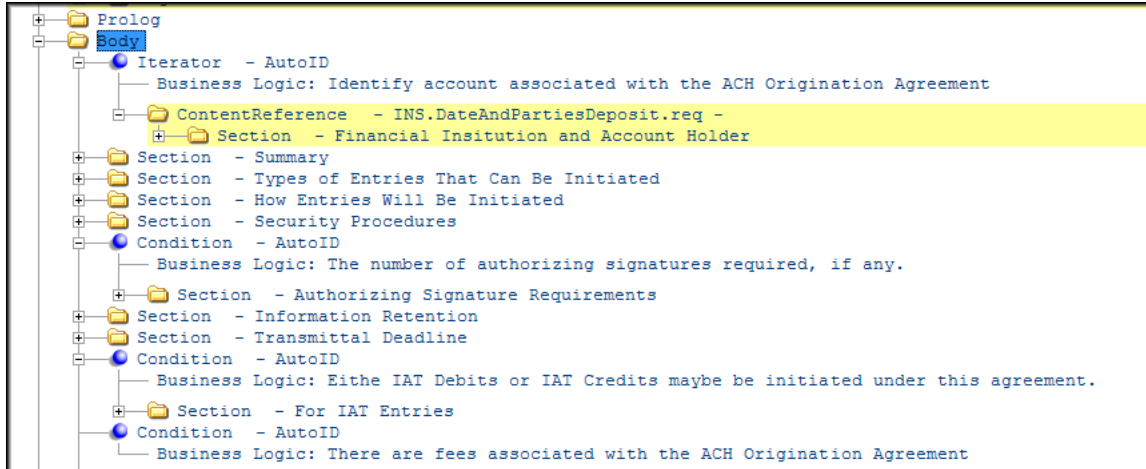
## Body (Section)

The **Body** element contains all the message data and contains various sections, paragraphs, and tables within the .REQ.

Typical child elements within a **Body** element consist of the following:

- **Section**
- **Para**
- **Title**
- **Table**
- **Condition/Choose/Iterator**
- **Note**

■ **Content Reference**



## Section

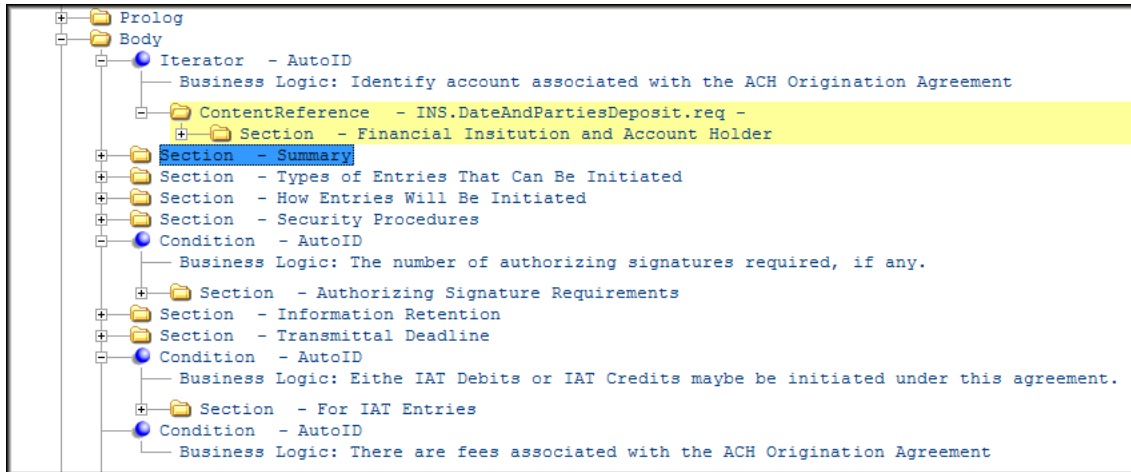
Add a **Section** element directly under the **Body** or under another **Section** (as a subsection of that other Section). It cannot contain text directly, but it can have a **Para** child that does contain text. It typically will have a title.

A **Section** child (subsection) of a **Section** will be indented inside that “parent” **Section**. Use a **Section** for anything that is a distinct topic within the document that can be titled. Name the main sections, like “Definitions” “Defaults” etc.

In the Outline Editor, give each section a name that is meaningful to the author. This “title” can be used to create Internal References to that section and is different from the actual section title under the title element, as it will not appear on the final document. The actual section heading that appears on the printed document appears under the Title element. In this example, the titles are the same, but they do not have to be identical.

Typical child elements within a **Section** element consist of the following:

- **Condition/Choose/Iterator**
- **Title**
- **Table**
- **List**
- **Note**
- **Para**
- **Signature**
- **Title**



## Section Level Attributes

The following attributes are available on the **Section** element and offer some control over how the stylesheet will format that section:

- **Boxing** - When set to **Yes**, the entire content of the section appears in a box. The content is kept together when setting the boxing to **Yes**.
- **BoxingColor** - When setting the **Boxing** attribute to **Yes**, the **BoxingColor** attribute specifies that the box is red instead of the default of black.
- **FeatureText** - Identifies text that is included in a document based on the selection made by the customer on one of our Menu worksheets.
- **KeepTogether** - When set to **Yes**, the entire content of the section (also available on other elements) will not break across columns or pages.
- **KeepWithNext** - When set to **Yes**, the stylesheet keeps the **Section** content with the next object at the same level or higher, typically the next **Section** (also available on other elements). An example of where this may occur is with a Notice that needs to appear directly above the **SignatureSection**.
- **Language** - Set to 'es' if this section only is in Spanish. This will change the font used in this section so that the Spanish accents are supported in all Eform outputs.
- **Number** - Set the value of this attribute to override the standard stylesheet setting for **Section** numbering. If the stylesheet is configured to number all sections and this attribute is set to **No** for a **Section**, that **Section** will not be numbered.
- **Span** - For documents that have a stylesheet that will produce two column layout documents this attribute can be used to make the **Section** (also available on other elements) span across the full width of the page. This is commonly used for **Tables**.

**Note:**

Any switch from two column to one column starts on a new page, but any switch from one column to two column can occur in mid-page, unless the two column follows a **PageBreak** element.

- **SuppressPeriod** - Suppress the period from printing by setting this attribute on the **Section** element to **Yes**; for example: when a title needs to end with a question mark.

## Viewing Dynamic Document Source XML

The **Source** option allows users to open the .XML code for an .REQ in a text editing application (typically Notepad).

- 1.
- 2.

## Working with Elements/Attributes

When working in the Outline Editor, users can utilize the a right-click mouse menu to perform many of the actions found in the Requirements Editor menus. However, certain actions can only be performing using the right-click menus, including the following:

- Adding child elements
- Adding attributes
- Adding rules
- Expanding and collapsing elements

### Moving items using Drag/Drop

Use caution when moving content in the Outline Editor. To move content, click to select the desired element; drag the element to the desired location. Note that all child elements and attributes of that parent also move with it.

**Note:**

- Before releasing the mouse button, watch closely for the dotted lines which indicate the level where your content will be placed. Be careful not to move your content underneath the wrong element.
- Note that users cannot drop on/from a read-only node.

### Cut/Copy/Paste

The Requirements Editor allows users to copy content from within a document or from another document.

Note that to link content so that changes on the source document appear on the destination document, use the Content Reference option.

**Note:**

Use the **Content Reference** option to link content so that changes on the source document appear on the destination document, .

The **Cut**, **Copy**, and **Paste** commands allow you to transfer text, data or objects within the .REQ file. Cut, Copy, and Paste functionality is virtually identical.

## Undo and Redo

The **Undo/Redo** options allow users to either erase the last change made to the .REQ (**Undo**) or reverse the effect of the undo action (**Redo**).

**Note:**

The limit to undo is 1,000 operations. The number of text changes inside a cell on one of the grid tabs is unlimited, but once the user exits that cell the undo history is gone.

## Child Elements

The **Add Child Elements** option allows users to add elements to a parent element. When creating an effective file following the rules of XML, child elements organize the information. A root element is a container for all other elements. While you must have a root to create a viable XML file, it is the child elements that add functionality to the page. Consider the root element a filing cabinet and child elements are the drawers to that cabinet. Add as many child elements to the root as necessary.

## Attributes

The **Add Attributes** option allows users to add attributes to element. Attributes are used to specify additional information about the element. It may help to think of attributes as a means of specializing generic elements to fit your needs. An attribute for an element appears within the opening tag. If there are multiple values an attribute may have, then the value of the attribute must be specified.

## Inserting Elements Before/After

The **Insert Element Before** and **Insert Element After** options allow users to insert elements before or after a specific element within the Outline Editor. An XML element is everything from (including) the element's start tag to (including) the element's end tag. An element can contain the following:

- other elements
- text
- attributes
- a mix of the above

## Using Rules

This section explains how to add rules to an .REQ file within the Outline Editor. The rules consist of the following:

- **Conditions**
- **Choose**
- **Iterators**

See **Adding a Condition, Iterator, or Choose** below for detailed information on using these rules.

## Viewing Rules

The **View Rules** option allows users to view rules from the following:

- Other documents
- All documents
- The current document

## Notes

Use the **note** element to add additional information concerning the desired element. This information typically indicates the following:

- which states are using the content (matching what the profile is set to)
- what state statute requires the text or special formatting.

Notes consist of the following:

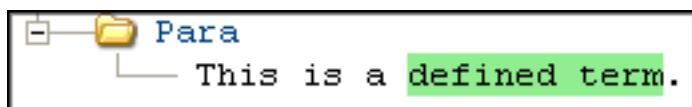
- **Analysis**
- **Private**
- **Citation**
- **Coding**
- **Datacollection**

Notes are not converted into content; however, they may be useful to those reviewing the .REQ files. Only Citation notes are viewable by the end customer. When authoring a new .REQ file, typically only Citation Notes and Private Notes are used. Citation notes include the cite dictating the content or format. Private Notes can be used to document notes about a discussion that resulted in the content appearing a certain way, or a reminder for the author or reviewer to handle the content a specific way.

## Defined Terms

Define a key term typically the first time it is used. Tag only the location where the term is being defined as a Defined Term. If appearing in other locations, it is authored in title case (with the exception of pronouns). In the tool, highlight the text that is considered a key term and right click to add a **Defined Term Tag**.

Based on the stylesheet, text within a **Defined Term** tag appears in italics and with double quotes around it. In the Outline Editor, the text will appear with green highlighting.



## Emphasis

Emphasis is given when a legal requirement dictates the appearance of text. The following types of emphasis are available:

- **Bold** - Bolds the text within the emphasis element.
- **Italics** - Makes the text within the emphasis element italics.
- **Underline** - Underlines the text within the emphasis element.
- **Strikethrough** - Use this option when a statutory requirement exists, and no other way exists to meet the requirement. It places a strikethrough on the text within the emphasis element.
- **ALL CAPS** - Use this option rather than typing the content in all caps. It changes all the text within the emphasis element to ALL CAPS.
- **Small Caps** - Use this option when possible in the place of All Caps (not all Eform outputs support Small Caps at this time). It changes all text within the emphasis element to Small Caps.
- **Font Size Small** - This option reduces the font size for all text within the emphasis element to be two point sizes smaller than the base font size of the document.
- **Font Size Large** - This option increases the font size for all text within the emphasis element to be two point sizes larger than the base font size of the document.
- **Custom Font Size** - Rather than setting the font size in relation to the base font size this allows you to explicitly set the font size required. This will be used when a regulation or statute requires a specific font size (ex. 16pt). The stylesheet will use the appropriate font to achieve the true height value that is entered.
- **Custom Font Color** - In instances where font color requirements should exist as something other than the standard Black (Red). This option allows the user to set the font color to Black, Red, or White. (The Black and White options would typically only be used when the text appears in a shaded table cell and the user wants to override the font color the stylesheet uses by default.)

- **Title Font** - This option changes the font use for body text from a Serif font to SansSerif (the same font used for titles in the document). This is used for captions for data fields or other areas where the appearance of a title (without the bolding or period) is required.
- **KeepTogether** - In certain cases content breaks from one line of a document to the next. An example of this is words like fee(s) where the (s) ends up on the next line. This occurs with the § symbol being separated from the statute number. Applying this **KeepTogether** option on the text that should stay together will force all of the text to move to the next line.

Do not use the following for future content development even though current content exist that use these. Use the above emphasis attributes instead.

- **Critical** - for content that has major compliance implications. Based on the stylesheet , it is formatted as bold and Small Caps.
- **Emphatic** - for content that deserves added attention and which is important for compliance. Based on the stylesheet . It is formatted with a bold typeface.
- **Informational** - content that deserves some added attention, but there are no significant compliance risks if the emphasis is not noticed. Based on the stylesheet , it is formatted with italics.

## Data

Insert a rule to print data or reserve space for a static data item; for example: if creating a sentence, right-click inside the sentence and select **Add Rule**. Upon creating a rule, add values for the **Public Name**, **GDD Name**, or a **Substitution Name** by adding this information in the **PTR Editor** tab.

### Note:

- By default, adding a rule to an element automatically displays the **PTR Editor** tab. For more information on using the PTR Editor, see [Product Technical Rule \(PTR\) Reference](#).
- To access the **GDD Name**, **Public Name**, and **Substitution Name** for a data item that you have just inserted, click outside the para and then (single click) back onto the data item. The **GDD Name**, **Public Name**, etc. will appear with a white background instead of being grayed when that data item is “active.”
- If the rule comes from a content reference, then **Public Name**, **GDD Name** and **Substitution Name**, as well as the PTR and business logic, are read only.

The inserted data rule appears in the Outline Editor view with the Data ID that has been assigned; or if a **Public Name** or **GDD Name** has been added it will display that name.

### Note:

A DTA ID is required to be unique to the Public Name/GDD Name/Substitution Name assigned to it.

If copying text that contains a data field and then changing the **Public Name/GDD Name/Substitution Name**, the following error appears:

- This data rule exists in more than one location. Changing the PublicName will update all locations. Do you wish to continue?

If you need a new data name click **No**, remove the data element, and insert a new one with a new DTA ID assigned.

The **Public Name/GDD Name/Substitution Name** should also only appear once in the REQ file. If attempting to create a new data field with a Public Name/GDD Name/Substitution Name that already exists, the following error appears:

- The PublicName <x> already exists for rule DTA.<x> and cannot exist on multiple rules. Would you like to reuse this rule?

Click **Yes** to use the same data field in multiple places.

## Data Attributes - Appearance

The following attributes allow you to override the standard appearance for that data item. These attributes are accessible from the dropdown menu on the **PTR Editor** tab on the far right of the screen.

- **Underline** - You can set the **Underline** attribute that will apply for this data item in the **PTR editor** tab; this overrides the default for the selected stylesheet. The following are standards for when a field should not get an underline even when the default for the stylesheet is to underline fields:
  - **SignerName**, **SignerTitle**, and **SignerDescription** fields below the Signature line
  - All fields that appear in the tagline of the document
  - Fields that are the only content within a table cell

Consumer Reporting Agency Name	Mailing Address	Telephone Toll-Free	Web Address

- If the field is more than three lines and is preceded and followed by text (if part of a sentence):

<b>Purpose of Addendum</b>
This Aircraft Security Agreement Addendum ("Addendum") is a supplement to a Security Agreement, dated _____, ("Security Agreement") between Debtor and Secured Party which secures a loan ("Loan") from Secured Party to _____ ("Borrowers") dated on _____. This Addendum includes additional terms, covenants and representatives.

- If the field is not preceded by and followed by text (if not part of a sentence) and extends to more than one line.

**Note:**

A field that appears with only a period following it should be considered a sentence for purposes of determining whether to underline.

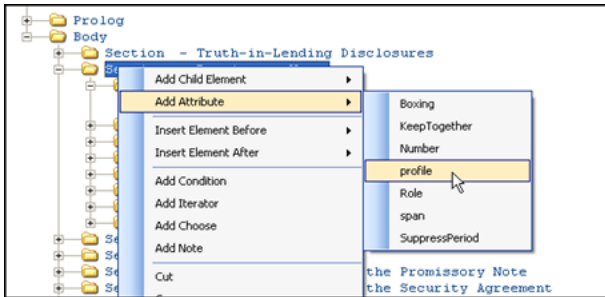
- The attorney/CA can also decide to remove an underline in other areas to improve the cosmetic appearance of the document.
- **Alignment** - Setting the alignment for a data item overrides the standard alignment applied by the stylesheet. (Percents are right aligned, Dollar amounts are left aligned, etc.) Typically this is used when there is a column of dollar amounts that are being added to show a total and we want all of the dollar amounts to be right aligned instead of left.
- **Prefix and Suffix** - Adding a value for the **Prefix** or **Suffix** attributes inserts that entry immediately before or after the data item in the document output. This is used when a calculation is being done on the data items and we add a +, -, or = as the **Prefix** value.

## Profiles

Based on the instance definition, this determines whether to include a particular element (typically a **Section**) in a document. Right-click on the element which you want the profile to apply and choose **Add Attribute** and **Profile**.

**Note:**

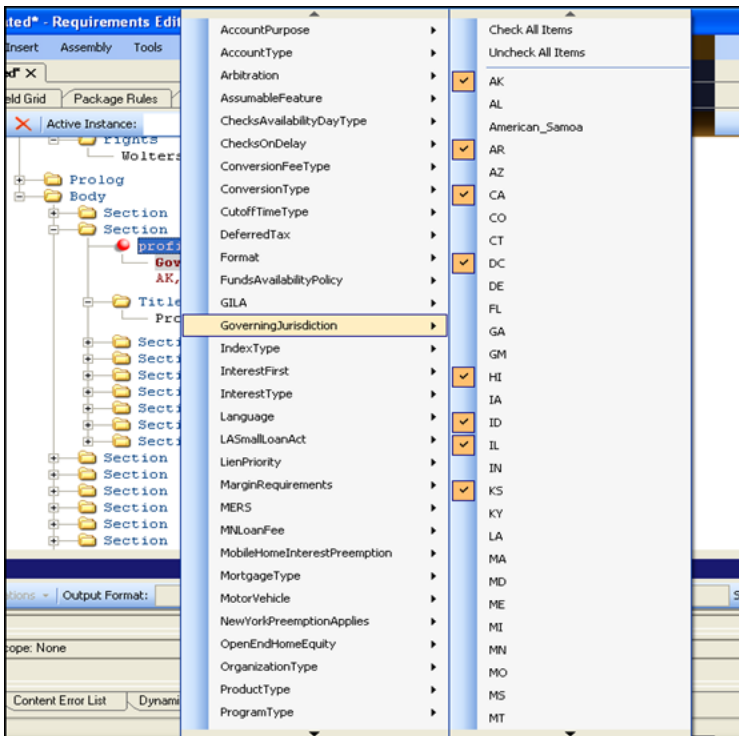
If specific values for a profile are not selected, by default all of the values apply. For example, not setting the value for Governing Jurisdiction for a section includes the content for all jurisdictions.



After adding the profile attribute, choose which profile category you want to apply. To do this right click on the profile attribute and a list of categories will be displayed.

Hover over the desired category to display a list of available values in another window. Click to choose all the values that apply for this section and a checkmark will appear to the left of the value. An option exists at the top of the Governing Jurisdiction value window to **Check All Items** and **Uncheck All Items**. These options allow you to select/deselect all states except one. You can choose to **Check All Items** and then deselect the one state that is not needed.

If more than one profile category applies, set additional values. After setting all of the profile values, click in the main Outline Editor window and the profile windows will close.



The selected profile attributes now appear within the Outline Editor.

**Note:**

To locate content within an REQ file that has sections that vary by Governing Jurisdiction: insert the variations in alphabetical order by state abbreviation starting with the variation that covers the most states.

## Paragraphs

Use a **Para** element to add a piece of text to the document. A **Para** is usually found within a **Section** or **Formal Paragraph**.

## Formal Paragraphs

A **FormalPara** can contain a title like a section, but is used when the content of the FormalPara is not significant enough to warrant a true **Section** (something that would appear in an index or Table of Contents); for example: content that does not exist independently for reuse in multiple REQ files.

The **FormalPara** content will not indent from the parent element and appears as **Section** content.

## Emphasis Elements and Tags

When legal requirements dictate how text must appear, you can add the required emphasis, such as: Bold, Italics, Underline, Strikethrough, Small Caps (small capital letters) and All Caps (All capital letters).

There are also font size options, which can make selected text 2 points larger or smaller than the current base font.

**Note:**

To display or change the base font size (and other display options) in the Requirements Editor, select View > Tree View Options.

In addition, you can add XML tags for defined terms, document references, superscript, and Xref targets.

**Note:**

Keep in mind that emphasis options are configured in the stylesheet. These options are used to ensure consistency across a set of documents. For example, if the Title Font option is set to 12 point Arial with bold typeface in the stylesheet, that is how all titles will display, provided that the Title Font option is used on all documents.

## Emphasis Menu Options

The table below describes the emphasis and XML tag options.

### Properties

This section displays the **Properties** information.

Emphasis Menu Options	
Menu Option	Description & When to Use
Bold	The selected text is set bold typeface.
Italics	The selected text is set in italics.
Underline	Underlines the selected text. Due to the limitations of E-forms, avoid using underline with non-runtime content.
Strikethrough	Strikes the selected text, for example: Your account will automatically renew unless you prevent it. Use strikethrough only when there is a statutory requirement and there is no other way to meet it.
All Caps	The selected text is set in capital letters, for example: BORROWER NAME. Use this option instead of typing the text in all capital letters.
Small Caps	The selected text is set in small capital letters, for example: BORROWER NAME. Use this option instead of All Caps whenever possible.
Level: Informational	Do not use. Use Italics instead.
Level: Emphatic	Do not use. Use Bold instead.
Level: Critical	Do not use. Use All Caps instead.
Font Size: Small	Sets the font size of the selected text 2 point sizes smaller than the base font size.
Font Size: Large	Sets the font size of the selected text 2 point sizes larger than the base font size.
Custom Font Size	Sets the font to the custom size specified in the dropdown menu.
Title Font	Use to set the selected text to the title font.
Keep Together	Use to allow text to wrap in an element, but not allow the text to break between lines, such as in citation references and checkboxes.
Wrap in 'DefinedTerm' Tag	Use defined term tags only when a term is used and defined for the first time.
Wrap in 'DocumentReference' Tag	Use a document reference for text that refers to an external document, such as the Wall Street Journal.
Wrap in 'Superscript' Tag	The selected text displays slightly smaller and higher than other text. The service mark (SM) in the following example is set in superscript: Initial Interest <sup>SM</sup> Adjustable Rate Rider.
Wrap in 'Hyperlink Tag	The selected text display as an active Web link (text is underlined and displays in blue) when the document is assembled.
Wrap in 'XrefTarget' Tag	Use Xref target tags to refer to a specific sentence or word in another section of the same document.

## Using Emphasis Elements

This is an example procedure. It describes how to add bold and italic emphasis elements to a document name in a dynamic REQ file. It shows you how to determine when multiple emphasis elements have been applied to a block of text, and how to add further emphasis or delete emphasis as needed.

After reading this procedure, you will understand the basic rules for using emphasis and you will be able to apply the required emphasis to the documents that you edit and author.

**Note:**

When you apply multiple emphasis tags to the same word or block of text, spacing errors, such as an extra space or a missing space, can occur. Always check your changes on the Preview tab before saving them.

Emphasis (and XML tags) can be applied to text in the following elements:

- Para
- Title
- Document Name

## Deleting Emphasis Elements

You can delete emphasis tags when necessary. All emphasis applied to the text between emphasis tags is removed when the tags are deleted.

A common instance when it is necessary to delete emphasis tags is when a block of text, such as The Wall Street Journal is tagged for bold and italic emphasis. To be correct, referenced external documents, such as The Wall Street Journal should be wrapped in Document Reference tags, which include the bold and italic styles.

To delete emphasis tags, right-click the first tag and select **Delete**.

## About Defined Term, Document Reference, and Superscript Tags

This section explains how and in what circumstance to use **Defined Term**, **Document Reference**, and **Super Script** tags. Each tag has its own style settings and you add the tags the same way (and from the same menu) that you add emphasis tags.

Like the emphasis options, these tags can be applied to text in a dynamic REQ file. Unlike the emphasis elements (where **Bold**, **Italics**, **Underline**, etc. appear as an **Emphasis** tag), each of these elements has its own tag.

**Note:**

The procedure for using **Xref** and **Xref Target** tags is described later in this section.

The options for applying **Defined Term**, **Document Reference**, and **Super Script** tags appear at the bottom of the emphasis menu.

XML tags (and emphasis options) can be applied to text in the following elements:

- Para
- Title

- Document Name

*Defined Term Tags:*

Use defined term tags only when a term is used and defined for the first time. When the term is used again later in the document, use title case (initial capitals) to denote it is a defined term. In the generated document, the text inside the defined term tags displays inside quotation remarks and is formatted with italics, for example: “Current Index.”

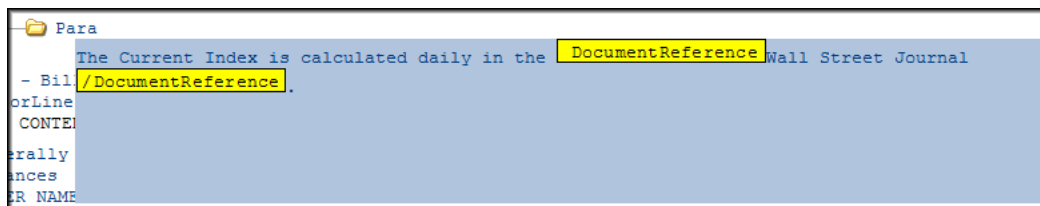
When you open an editing box, defined terms appear with green tags; see the graphic below.

The screen shot also shows two ways of effectively and easily defining a term—one with the term at the beginning of the sentence, and the other with the term at the end.

*Document Reference Tags:*

Use document reference tags for text that refers to an external document, such as the Wall Street Journal. Apply the tag only to the first occurrence of the reference. The text inside the tags is formatted in bold italics, for example: **Wall Street Journal**.

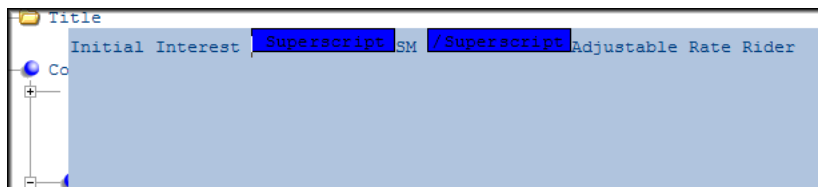
When you open an editing box, document references appear with yellow tags, see the graphic below.



*Superscript Tags:*

Use Superscript tags when for service marks and trademarks. The text inside the tags displays higher and slightly smaller than the base font.

When you open an editing box, text set in superscript appears with blue tags; see graphic below.



## Using Defined Term, Document Reference, and Superscript Tags

This is a sample procedure. It describes how to add the required Defined Term, Document Reference, and Superscript tags to the document text listed below:

"Beginning with the first Change Date, my interest rate will be based on an Index. The Index is the average of interbank offered rates for one-year U.S. dollar-denominated deposits in the London market ("LIBOR"), as published in The Wall Street Journal. The most recent Index figure, available as of the date

45 days before each Change Date, is called the Current Index. Other interest rates, such as those offered in the Initial Interest Adjustable Rate Rider are not valid."

This procedure explains how to apply and use these tags when editing and authoring documents.

**Note:**

Do not apply additional emphasis between **Defined Term**, **Document Reference**, and **Superscript** tags. It will cause errors.

The style sheet dictates which emphasis elements, such as **Bold** and **Italics**, these tags use. For more information about which tags use which styles, see the table in the section **About Emphasis Elements and Tags**.

## Using Xref and Xref Target Tags

This is a sample procedure. It describes how to use cross-reference (Xref) tags to make a general reference to another section in the same document. This procedure also describes how to use Xref target tags to reference a specific piece of information, such as an address, in another section.

When a document is created, the author names the various sections of the document. When inserting an Xref you can choose from a list of section names as the target of the reference.

You choose how you how you want the reference to display: by section number, section title, or both number and title. Note that you may have to make minor changes to the text so that it flows properly and the references make sense.

After reading this procedure, you will understand how to apply Xref and Xref target tags and you will be able to use them as needed in the documents that you edit and author.

**Note:**

Like emphasis tags, Xref and Xref target tags only can be added in Para, Title, and Document Name elements.

## Implementing Barcodes as an Ancillary Output Option

Implementing a barcode involves identifying the need to apply a barcode to a document, defining requirements such as appearance, labeling, and the barcode type to apply. Then you must submit the request with the instructions for the barcode and the transaction data to process the barcode instructions.

Barcodes are only applied to documents generated by the Expere IE and are added to all generated pages; that is, each generated page in the document and document within a package. The ability to selectively apply barcodes to specific pages in a document is not supported. The barcode object can be submitted as part of the **AncillaryOutputObjects** on a document generation request to generate barcodes on **all** documents in a requested package or can be submitted on the **DocumentSet** or **Document** object on a document generation request to generate unique barcodes on a **document by document** basis.

In the ExpereAPI, barcodes are implemented as part of the **AncillaryOutputOptions** object on several operations used to generate documents, including:

- [Generate](#)

- [SelectGenerateAndSave](#)
- [AssembleDocumentSet](#)

Using the **DocumentSet** object, to apply a unique barcode to individual documents, they are implemented on the following operations:

- [Generate](#)
- [GenerateAndSave](#)
- [AssembleDocuments](#)

To implement barcoding on a document through these API calls, you must first define the parameters of the barcode in the **Barcode** object and then submit that object on the **AncillaryOutputOptions** object submitted in the request. Following is an example of an **AncillaryOutputOptions** submitted on a request that will generate a barcode.

```
<exp:AncillaryOutput>
  <!--Zero or more repetitions:-->
  <exp:AncillaryOutputOption>
    <exp:OutputType>Barcode</exp:OutputType>
    <exp:Barcode>
      <exp:Type>barcode128</exp:Type>
      <exp:Pattern>Page_%page%_of_%totalpages%_%date%_Doc_%docsequence%_of_%docsequencetotal%</exp:Pattern>
      <exp:UseCoverPage>True</exp:UseCoverPage>
    </exp:Barcode>
  </exp:AncillaryOutputOption>
</exp:AncillaryOutput>
```

```
<exp:AncillaryOutputOptions>
  <exp:OutputType>Barcode</exp:OutputType>
  <exp:Barcode>
    <exp:Type>barcode128</exp:Type>
    <exp:Pattern>Page_%page%_of_%totalpages%_%date%_Doc_%docsequence%_of_%docsequencetotal%</exp:Pattern>
    <exp:UseCoverPage>True</exp:UseCoverPage>
  </exp:Barcode>
</exp:AncillaryOutputOptions>
```

The barcode is rendered on a generated document, as follows:



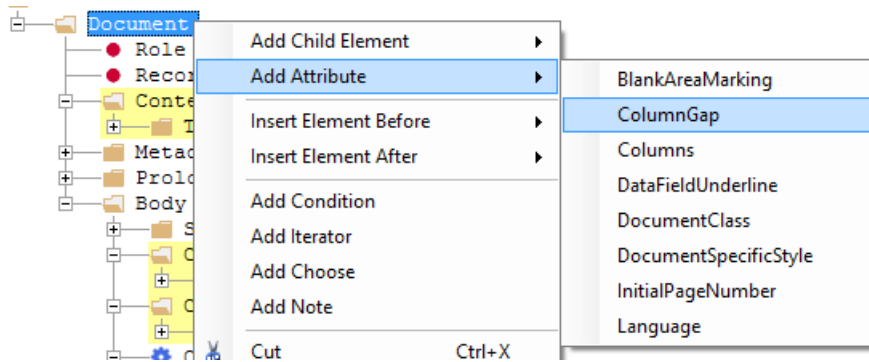
## Modifying Space Between Columns

This procedure explains how to resize the amount of space between two columns in a document.

**Note:**

The ColumnCap attribute should only be used when the standard column gap of 22pts is not sufficient, per compliance requirements.

1. Open the document within Requirements Editor and locate the **Document** element.
2. Right-click the **Document** element and select **Add Attribute > ColumnGap**.



The **ColumnGap** attribute is added to the **Document** element.

3. Double-click the **ColumnGap** attribute to place it in Edit mode and enter the desired width in inches(in) or points (pt) [ex. *1.5in*]

## Working with Static Requirements Documents

This section describes some of the key authoring activities in creating, editing, and producing static documents. For an overview of authoring static documents, see [Authoring Static REQ Files](#).

### Exporting an XTemplate

Static requirements files, or static documents, are written and prepared in advance for particular processes or lending transactions and brought into the Requirements Editor through an XTemplate (layout-driven) image. On occasion, an error may exist with an XTemplate or an update may be required. In these cases, the author will export the XTemplate for a static document and apply any required updates to the XTemplate. The **Export xTemplate** option (**Static > Export XTemplate**) option the author to send an XTemplate to a folder location where the XTemplate can be picked up by the author to make changes to image in the I-32 design tool.

### Export an XTemplate

The static document subject to revision through editing of the XTemplate is known and open in the Requirements Editor.

1. Open the target static document in the Requirements Editor.
2. In the **Static** menu, click **Export xTemplate**.
3. In the **Save As** dialog box, enter the file name or select the file to overwrite and save the export by clicking **Save**.

At this point, the user should make any image changes in I-32 and import the XTemplate back into the Requirements Editor.

### Importing an XTemplate

Static requirements files, or static documents, are written and prepared in advance for particular processes or lending transactions and brought into the Requirements Editor through an XTemplate (layout-driven) image. On occasion, an image change is applied to an XTemplate and that image must be imported into the Requirements Editor to update a static document. The **Import xTemplate** option (**Static > Import XTemplate**) allows the author to import an updated XTemplate file associated with a static document.

### Import an XTemplate

The static document from which the revised XTemplate will be imported is known and open in the Requirements Editor.

1. Open the target static document in the Requirements Editor.
2. In the **Static** menu, click **Import xTemplate**.
3. In the **Synchronize with xTemplate** dialog box, verify the **xTemplate Path** is accurate; if not, click **Browse** to navigate to the desired location.
4. Click **OK**.

## Adding Continuation Pages to Static Documents

This feature is exclusive to static requirements files.

The **Continuation Page Setup** option provides a means to create an overflow page for additional information that would not fit in a **Form View** text box/field.

1. In the Static menu, click **Continuation Page Setup**.
2. In the *Continuation Page Setup* dialog box, select the *Continuation Page* option and choose the last page number to apply to the continuation page.

Continuation pages are supported on the last page of the active document only.

3. Click **OK**.

## Working with Merge Sessions

Within the Requirements Editor tool, you can merge content distributions to your existing repository, allowing you to selectively retain any modifications you have made to your content.

### New Merge Session

A new Merge session can be created by selecting **New Merge Session** from the **File** menu.

### Open Merge Session

An existing/previous session can be opened by selecting **File > Open Merge Session**.

The previous session file has an extension of **.msession**. When opening an existing Merge Session, identify the msession file located in the folder created during the [New Merge Session](#).

Name	Date modified	Type	Size	L
...	6/23/2015 1:32 PM	File folder		
...	6/23/2015 1:32 PM	File folder		
...	6/23/2015 1:32 PM	File folder		
...	6/23/2015 1:32 PM	File folder		
...	6/23/2015 1:32 PM	File folder		
normalized	6/23/2015 1:30 PM	File folder		
RetailBU.msession	6/23/2015 1:32 PM	MSESSION File	1,487 KB	
ContentReferencesCatalog.xml	6/23/2015 1:31 PM	XML File	2,208 KB	

### Close Merge Session

To close your Merge session without closing Requirements Editor, select **File > Close Merge Session**.

## Merge Session Windows

This section introduces you to the individual windows that are present within a Merge session.

There are four major Merge Session windows: [Merge Session Window](#), [Shared Content Diffs Window](#), [Content Item Diffs Window](#), and [REQ Diffs Window](#). Each window can be accessed from the Requirements Editor **Edit** menu. Example: **Edit > Shared Content Diffs**.

Repository analysis is a complex process and relies on all Merge Session windows to be comprehensive. It is recommended that you organize your window layout so that you can view all pertinent windows simultaneously during the reconciliation process. The following example displays The Merge Session windows in a functional layout.

6917 Merge Session - Requirements Editor (Administrator)

FILE EDIT VIEW MERGE TOOLS WINDOW HELP

Other Windows

**Merge Session**

Name	6917
Results Location	C:\TFS\Expere_scrum\Rep
<b>Repositories</b>	
Incoming	C:\TFS\Expere_scrum\Rep
Existing	C:\TFS\Expere_scrum\Rep
Common Base	C:\TFS\Expere_scrum\Rep
<b>REQ Document Counts</b>	
Total Documents	40
Total Base Documents	28
Total Custom Documen	12
Added	3
Unmodified	6
Modified	6
Deleted	13
Invalidated	1 (1 Remaining)
Spared	3
Impacted	6 (6 Remaining)
Orphaned	2 (2 Remaining)
<b>REQ Change Counts</b>	
Total Changes	76 (7 Conflicts - 69 Automati
Incoming	36 (36 Remaining)
Existing	17 (17 Remaining)
Both	16 (16 Remaining)
Conflicts	7 (7 Remaining)
<b>Shared Section Counts</b>	
Total Shared Sections	14
Total Base Shared Sec	11
Total Custom Shared S	3
Added	0
Unmodified	3
Modified	1
Deleted	7
Invalidated	0 (0 Remaining)
Spared	1
Impacted	1 (1 Remaining)
Orphaned	1 (1 Remaining)
<b>Content Item File Counts</b>	
Total Content Items	59
Total Base Content Iter	53
Total Custom Content It	6
Added	1
Unmodified	46
Modified	4
Deleted	2
Invalidated	0 (0 Remaining)
Spared	1
Impacted	4 (4 Remaining)
Orphaned	1 (1 Remaining)
<b>Content Item Change Counts</b>	
Total Changes	4 (1 Conflicts - 3 Automatic)
Incoming	2 (2 Remaining)
Existing	0 (0 Remaining)
Both	1 (1 Remaining)

**Total Content Items**  
Total Content Items (Total Base + Total Custom)

**REQ Diffs**

+ Added (3)   
 = Unmodified (6)   
 ✖ Modified (6)   
 ✖ Deleted (13)

✖ Invalidated (1)   
 ★ Spared (3)   
 ! Impacted (6)   
 ✖ Orphaned (2)

6917

- ORG.WoltersKluwerFinancialServices.Root
  - ORG.Child01
    - Added\_Invalidated\_Dynamic
    - Deleted\_Orphaned\_Dynamic
    - Modified\_Impacted\_Both\_Dynamic
    - Modified\_Impacted\_Conflict\_Dynamic
    - Modified\_Impacted\_Existing\_Dynamic
    - Modified\_Impacted\_Incoming\_Dynamic
    - Modified\_Incoming\_Dynamic
    - Reference\_File\_1
    - Reference\_File\_2

**Shared Content Diffs**    **Content Item Diffs**

+ Added (0)   
 = Unmodified (3)   
 ✖ Modified (1)   
 ✖ Deleted (7)

✖ Invalidated (0)   
 ★ Spared (1)   
 ! Impacted (1)   
 ✖ Orphaned (1)

6917

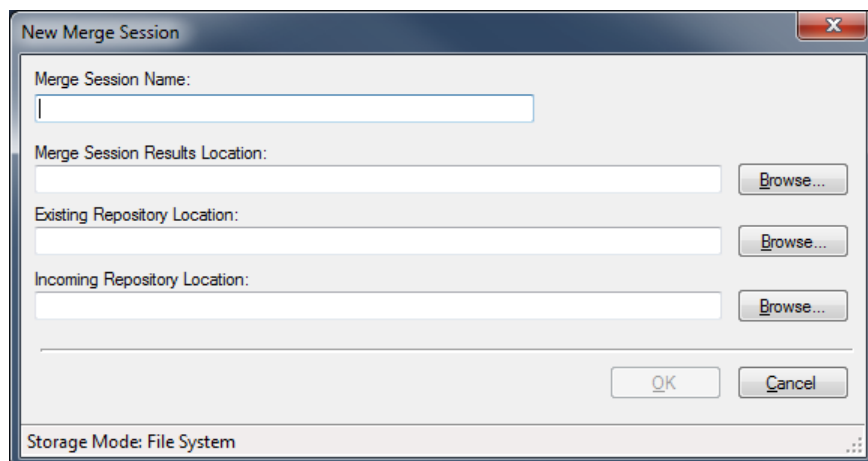
- ORG.WoltersKluwerFinancialServices.Root
  - ORG.Child01
    - Reference\_File\_1.Modify this section
      - Modified\_Impacted\_Both\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.Chil
      - Modified\_Impacted\_Conflict\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.C
      - Modified\_Impacted\_Existing\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.C
      - Modified\_Impacted\_Incoming\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG
      - Modified\_Incoming\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.Child01)
    - Reference\_File\_2.Do not modify
      - Modified\_Impacted\_Conflict\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.C
      - Modified\_Impacted\_Incoming\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG
      - Modified\_Incoming\_Dynamic (ORG.WoltersKluwerFinancialServices.Root\ORG.Child01)

Ready

All windows are dockable and can be configured to suit your personal preference. For more information, refer to [About the Interface](#).

## New Merge Session

After selecting **New Merge Session** from the **File** menu, the New Merge Session dialog is displayed.

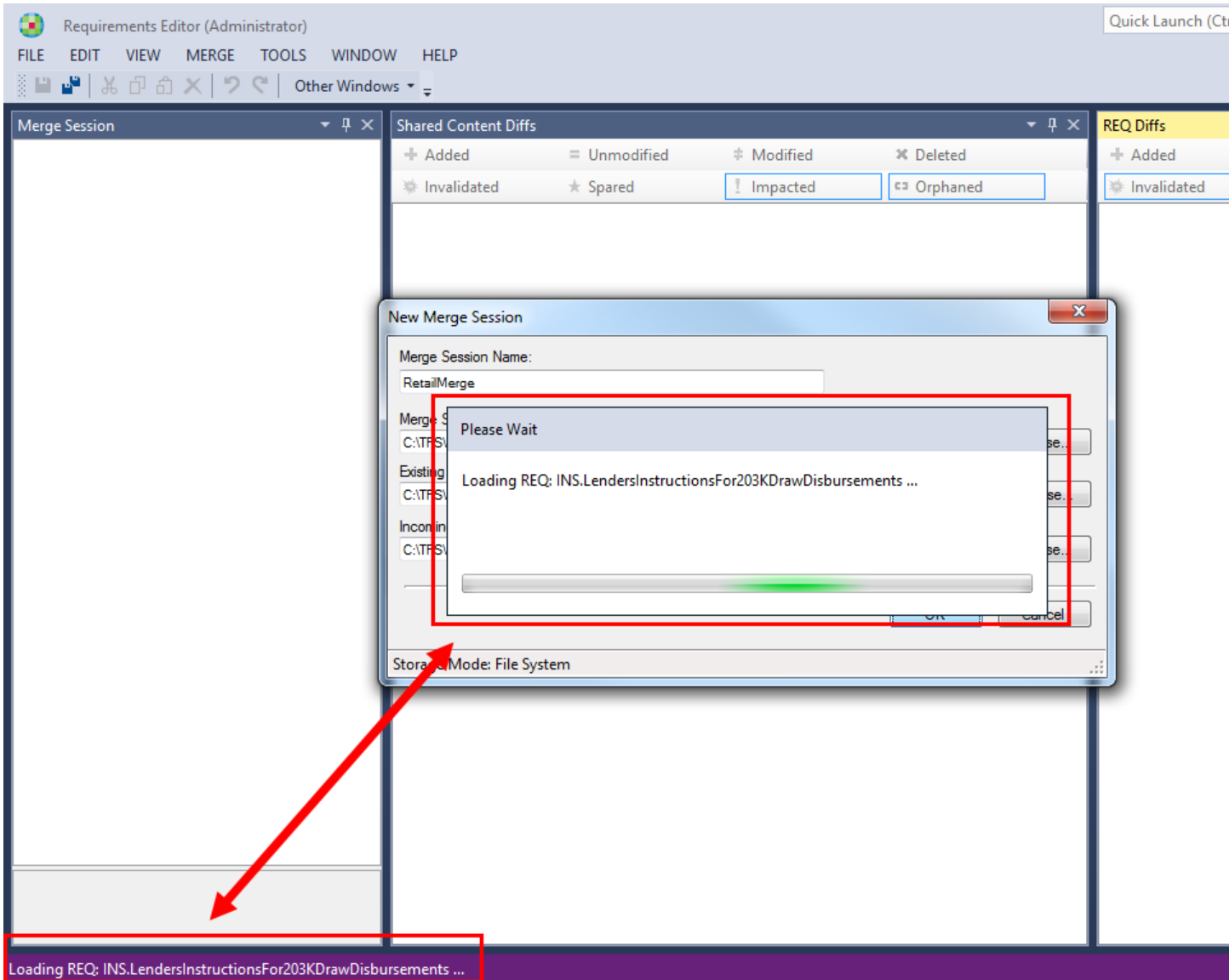


## New Merge Session Dialog

Field	Description
Merge Session Name	The name of the Merge session. The name you enter becomes the name of the folder created in the Merge Session Results Location.
Merge Session Results Location	The location where you want the new merge session to be saved. The session file (*.msession), the analysis files and folders and the Merge Results will be placed in this location.
Existing Repository Location	The location of the existing custom repository. This location must contain a valid Organizations.xml file.
Incoming Repository Location	The location of the new repository to be merged to the existing. This location must contain a valid Organizations.xml file.

When all fields are completed, click **OK** to begin the Merge analysis.

During the analysis process, a progress window is displayed. The progress is mirrored in the Requirements Editor status bar.



**Note:**

The time required to analyze depends on size the of the repositories and the number of customizations and changes.

## Merge Session Window

After creating a Merge session (see [New Merge Session](#)) or opening an existing session, The Merge Session window is displayed, providing access to Merge session details and Diffs for Shared Content and Content Items.

Merge Session	
<b>Merge Session</b>	
Name	Merge201506
Results Location	C:\temp\Merge\
<b>Repositories</b>	
Incoming	C:\TFS\Expere_scrum\Repo\BasePlusUsBankMortgage\Be
Existing	C:\TFS\Expere_scrum\Repo\BasePlusUsBankMortgage\Be
Common Base	C:\TFS\Expere_scrum\Repo\BasePlusUsBankMortgage\Be
<b>REQ Document Counts</b>	
Total Documents	1203
Total Base Documents	995
Total Custom Documents	208
Added	225
Unmodified	584
Modified	154
Deleted	32
Invalidated	0 (0 Remaining)
Spared	156
Impacted	46 (46 Remaining)
Orphaned	6 (6 Remaining)
<b>REQ Change Counts</b>	
Total Changes	514 (37 Conflicts - 477 Automatic)
Incoming	388 (388 Remaining)
Existing	89 (89 Remaining)
Both	0 (0 Remaining)
Conflicts	37 (37 Remaining)
<b>Content Item File Counts</b>	
Total Content Items	60
Total Base Content Items	59
Total Custom Content Items	1
Added	0
Unmodified	56
Modified	1
Deleted	2
Invalidated	0 (0 Remaining)
Spared	1
Impacted	0 (0 Remaining)
Orphaned	0 (0 Remaining)
<b>Content Item Change Counts</b>	
Total Changes	0 (0 Conflicts - 0 Automatic)
Incoming	0 (0 Remaining)
Existing	0 (0 Remaining)
Both	0 (0 Remaining)
Conflicts	0 (0 Remaining)
<b>Total Content Items</b>	
Total Content Items (Total Base + Total Custom)	

The Merge Session window provides a detailed summary of your session, as well as provides detailed information regarding the number of documents remaining to be reconciled. This includes the following expandable sections within the tab:

- **Merge Session** - displays the Name of the session and the location in which the results were saved
- **Repositories** - displays the locations of the Incoming, Existing and Common Base repositories
- **REQ Document Counts** - displays the total number of each type of REQ (Total Documents, Added, Modified, Deleted, etc)
- **REQ Change Counts** - displays the total number of changes within the REQs compared to the Base documents

- **Content Item File Counts** - displays the number of specific Content Items (Total Content Items, Added, Unmodified, etc)
- **Content Item Change Counts** - displays the number and types of Content Items changed

**Note:**

Clicking a heading in the Merge Session window displays brief help for that section in the bottom pane of the window.

The Merge Session window is the parent controlling window for the session. If the it is closed, any open session windows will remain open, but will display not information.

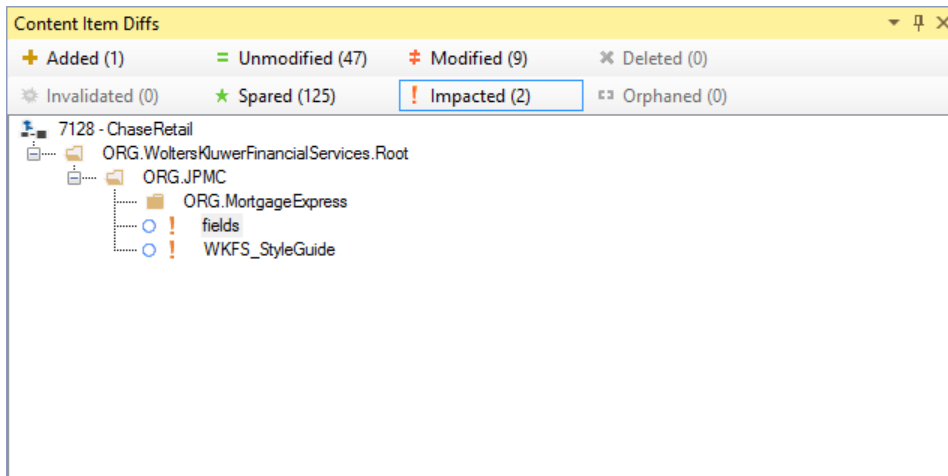
**Note:**

If the Merge Session window is closed during a session, it will not automatically be displayed when the session is opened again (**File > Open Merge Session**). Requirements Editor retains window configuration when it is closed.

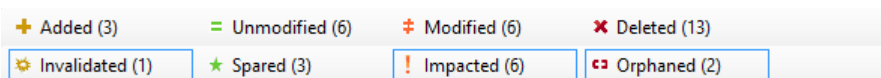
Items Remaining: Numerous reported documents within the Merge Session window contain an indication of how many documents must require attention. Under the REQ Document Counts in the above example, notice there are 46 documents remaining to be addressed. As you address each document, this number updates so that you can easily see how many more must be addressed.

## Content Item Diffs Window

The **Content Item Diffs** tab displays all repository-wide common items not in REQ format analyzed during the Merge session.

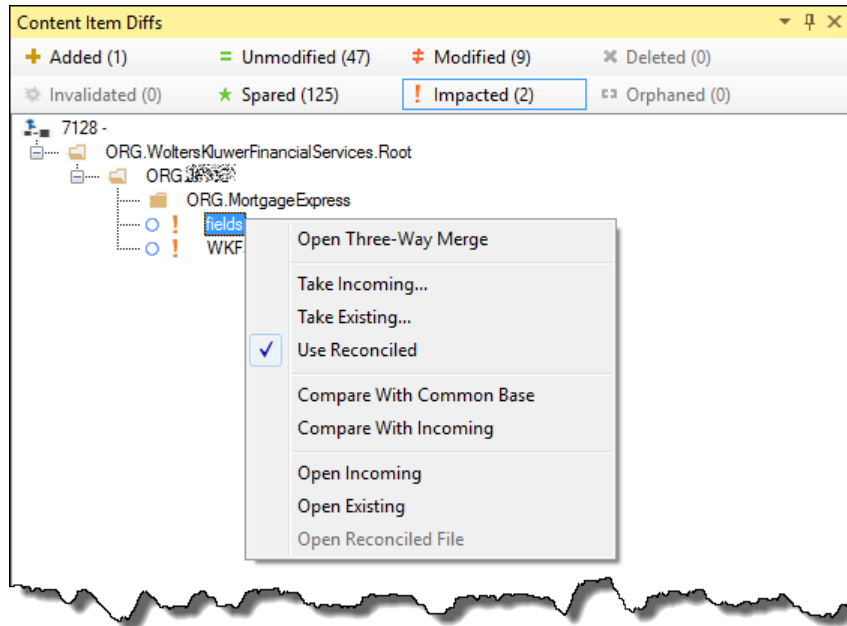


At the top of the tab are buttons that allow you to filter the items displayed within the tab, as follows:



For more information, refer to [Merge Window Filters](#).

Right-clicking any Content item in the **Content Item Diffs** tab allows you to open the selected item's Incoming or Existing version, or compare it with the Common Base version.

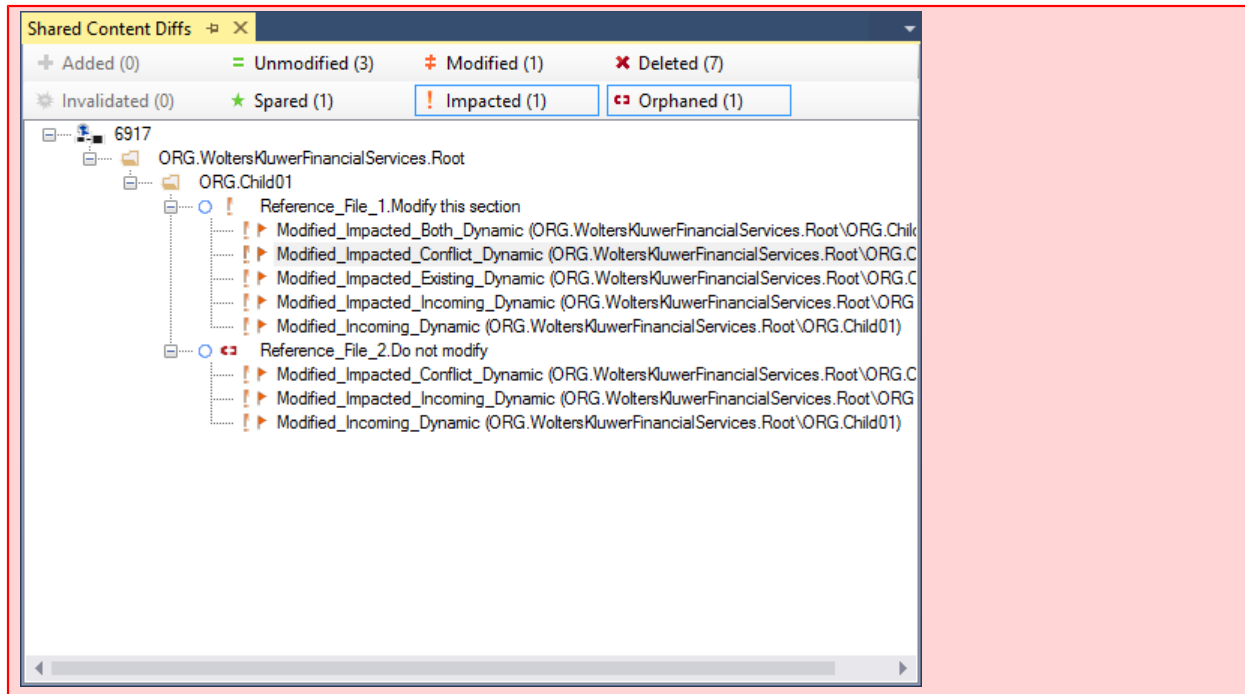


Context menus are designed to help you perform the most common tasks when working with specific windows. For more information, refer to [Context Menu Items](#).

## Shared Content Diffs Window

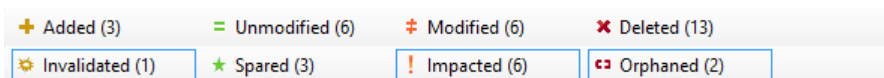
The **Shared Content Diffs** tab

**Comment [kelly.mankenber]:**Update with multiple section example



The **Shared Content Diffs** tab displays all referenced content that has changed. Because this is a shared ContentReference, you must decide if you want to take the referenced content changes or break the connection and keep your changes. In most cases, the changes in the referenced content should be taken in order to keep the link with the referenced content.

At the top of the tab are buttons that allow you to filter the items displayed within the tab, as follows:



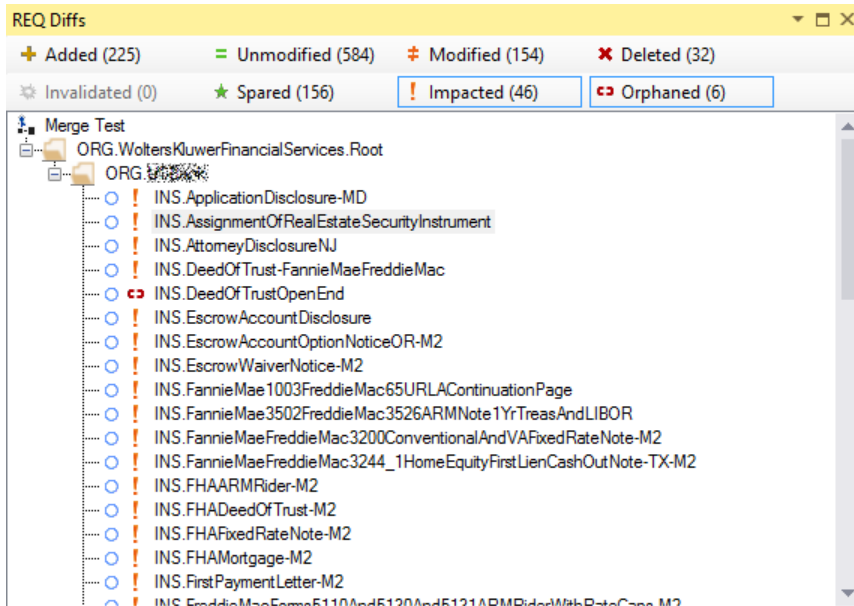
For more information, refer to [Merge Window Filters](#).

### Context Menu Items

Context menus are designed to help you perform the most common tasks when working with specific windows and documents. For more information, refer to [Context Menu Items](#).

## REQ Diffs Window

The REQ Diffs window displays all differences between the Incoming and the Existing REQ documents.

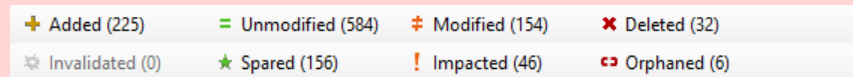


The blue Progress Status icon (○) indicates the associated document needs to be reconciled.

### Filters

#### Comment [kelly.mankenbergl]:

At the top of the tab are buttons that allow you to filter the items displayed within the tab, as follows:

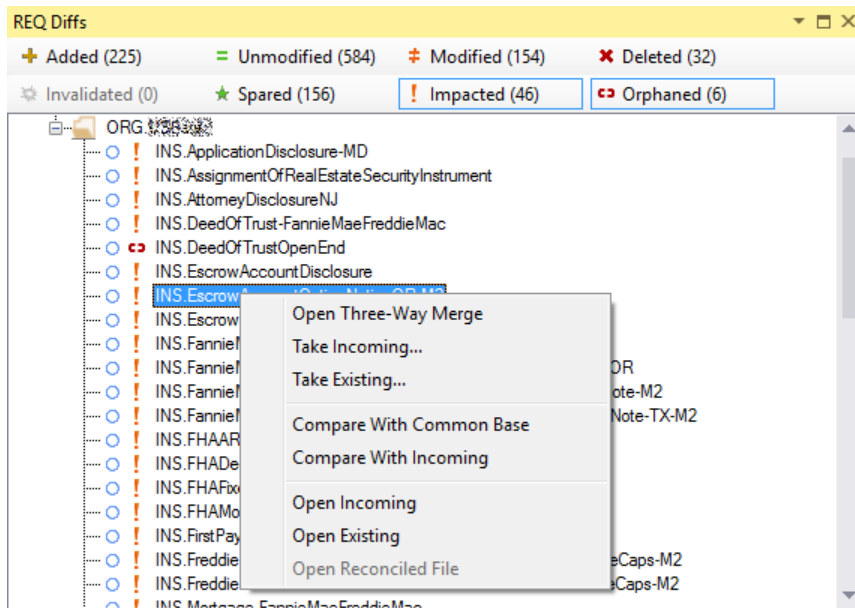


For more information, refer to [Merge Window Filters](#).

### Context Menu Items

Context menus are designed to help you perform the most common tasks when working with specific windows and documents. For more information, refer to [Context Menu Items](#).

**Figure 1.** Example of REQ Diffs Context Menu



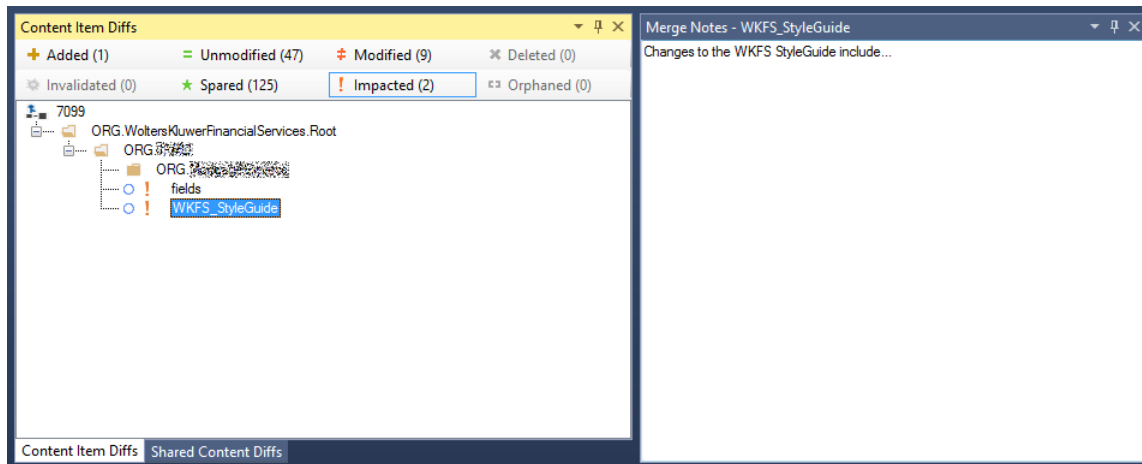
## REQ Diff Editor

This section discusses the various tabs found within the REQ Diff Editor when you open a document during a Merge session.

In the REQ Diff Editor, multiple Diff Docs can be opened simultaneously. The Incoming and Existing windows are shared among them and populated according to the active Diff Doc window having focus. If focus is set to something other than a Diff Doc window, the Incoming and Existing windows are cleared of their contents to reflect that.

## Merge Notes Window

The Merge Notes windows allows you to associate custom notes to any node within the repository windows. The following example shows a note within the Merge Notes window associated with the WKFS\_StyleGuide node of the Content Item Diffs window.



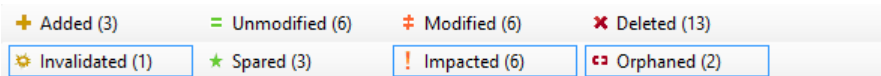
Notes are saved automatically when you select a different node within one of the Merge Session windows.

## Merge Window Filters

### Comment [Kelly.Mankenberg]:

Merge Sprint 7

At the top of the Merge Window tabs are buttons that allow you to filter the items displayed within the tab, as follows:



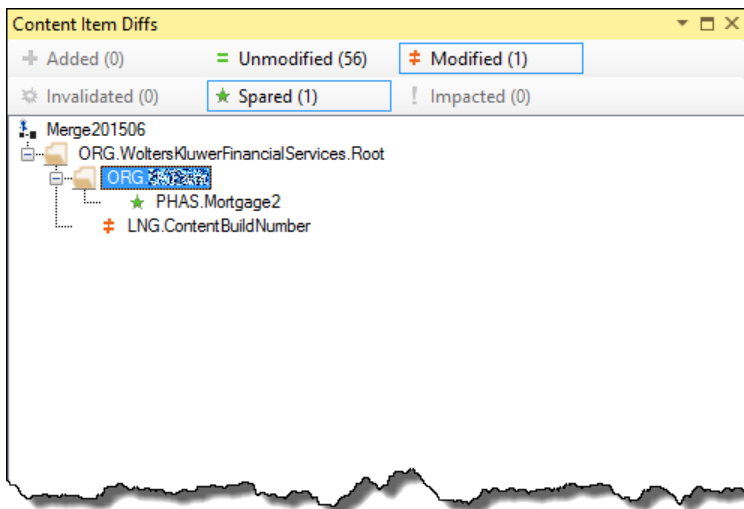
Comment [kelly.mankenberg]: Top row is for the WKFS (Incoming) repository. Bottom row is for the existing repository. Top buttons correlate with the button directly below.

- Added
- Unmodified
- Modified
- Deleted
- Invalidated
- Spared
- Impacted
- Orphaned

**Note:**

- The top row of filter buttons is associated with the base repository. The bottom row of filter buttons is associated with the child custom organizations.
- Filter buttons apply only to the root document.
- A filter button is disabled if there are no content items that can be associated with the filter. In the previous example, the **Added** button is disabled, indicating there were no Content Items added by the Incoming repository.
- Not all filters are present in all Merge windows. Example: Deleted and Orphaned are present in the REQ Diffs window, but not in the Shared Content Diffs window.
- For information on the icons used within the filter buttons, refer to [Icons](#).





By default, no filters are selected. You can select as many of the filters simultaneously as desired. In the following example, the Modified filter was selected causing the PHAS Mortgage2 item to display, and the Spared filter was selected causing the LNG.ContentBuildNumber item to display.



**Icons**

Each filter button contains an icon that is associated with the status of the items within the window. For example, the **Modified** button contains a symbol ( + ) associating the icon with modified items. Following is a list of all icons/filter buttons used within Merge windows.

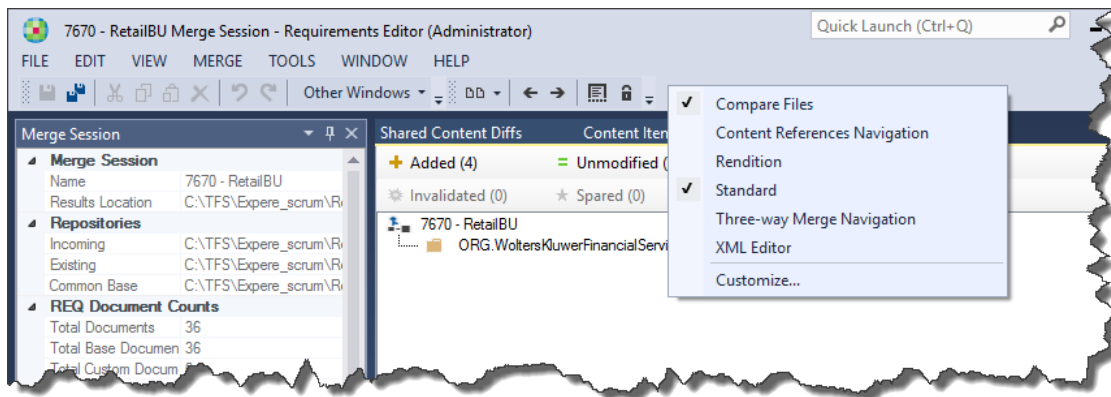
- Added
- Deleted
- Impacted
- Invalidated (1)

-  Modified
-  Orphaned
-  Spared
-  Unmodified

## Merge Toolbars

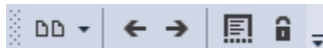
The following sections provide details for the buttons found within the toolbars accessible to the Merge windows only.

Toolbars can be displayed by selecting **View > Toolbars**, or by right-clicking an empty section of the toolbar area and selecting the desired toolbar.


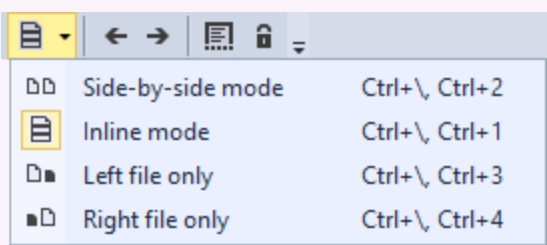






## Compare Files

This toolbar applies to the 2-way Diff Window.

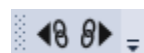


## Compare Files Toolbar Buttons


Button	Description
Compare View 	Allows you to select the layout of the two documents being compared. 
Previous Difference 	Navigates to the previous identified difference
Next Difference 	Navigates to the next identified difference
Ignore trim white space 	Trims the white space from being displayed in the viewed documents
Auto-synchronize Views 	Enables both document windows to move together as you scroll


### Content Reference Navigation

This toolbar applies to the 2-way Diff Window.



## Compare Files Toolbar Buttons





Button	Description
Previous Content Reference 	Navigates to the previous content reference

Button	Description
Next Content Reference 	Navigates to the next content reference

### Three-way Merge Navigation

This toolbar applies only to the Three-way Merge functionality ([REQ Diff Editor](#)) for REQ files only.

### Three-way Merge Navigation Toolbar Buttons

Button	Description
Previous Difference 	Navigates to the previous identified difference
Previous Conflict 	Navigates to the previous identified conflict
Next Conflict 	Navigates to the next identified conflict
Next Difference 	Navigates to the next identified difference

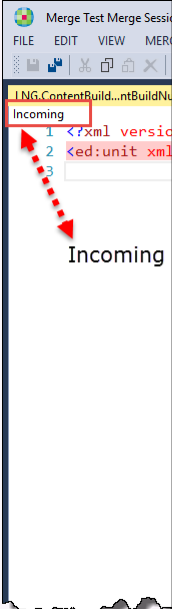
### Context Menu Items

**Comment [Kelly.Mankenberg]:**

Merge Sprint 7

Requirements Editor Merge windows have contextually based (right-click) menus. Menu items that appear when right-clicking are dependent on the window you are currently working in and the document you have selected.

**Comment [kelly.mankenberg]:**make a comparison check list

Menu Item	Description	Available in
<p>Compare With Common Base</p>	<p>Opens the selected document with a view of the Incoming and Existing versions.</p>  <p>For more information, refer to <a href="#">Compare with Common Base</a>.</p>	<ul style="list-style-type: none"> <li>- Shared Content Diffs Window</li> <li>- Content Item Diffs Window</li> </ul> 

Menu Item	Description	Available in
Compare with Incoming	Opens the selected document with a view of the Incoming and Existing versions. For more information, refer to <a href="#">Compare with Incoming</a> .	
Delete from Repository	Deletes the selected document from the existing repository.	<ul style="list-style-type: none"> <li>- Shared Content Diffs Window</li> <li>- REQ Diffs Window</li> </ul>
Open Incoming	Opens the incoming version of the selected document.	<ul style="list-style-type: none"> <li>- Shared Content Diffs Window</li> <li>- Content Item Diffs Window</li> <li>- REQ Diffs Window</li> </ul>
Open Existing	Opens the existing version of the selected document.	<ul style="list-style-type: none"> <li>- Shared Content Diffs Window</li> <li>- Content Item Diffs Window</li> <li>- REQ Diffs Window</li> </ul>
Open Reconciled File	Opens the reconciled version of the selected document.	<ul style="list-style-type: none"> <li>- REQ Diffs Window</li> </ul>
Open Three-Way Merge	Allows you to compare the selected document in the Incoming, Existing and RQDIF windows. For more information, refer to <a href="#">Three Way Merge</a> .	<ul style="list-style-type: none"> <li>- REQ Diffs Window</li> </ul>

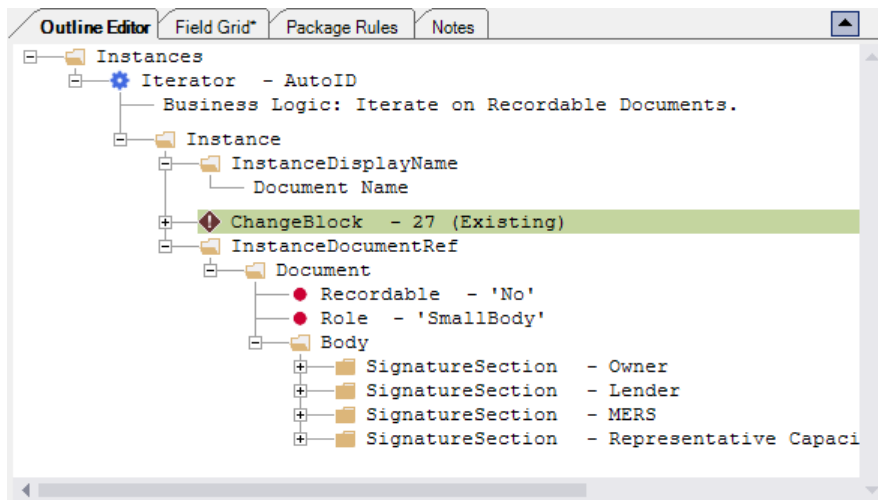
Menu Item	Description	Available in
Take Incoming	Accepts all changes in the incoming version of the selected document	- REQ Diffs Window
Take Existing	Accepts the existing version of the selected document	- REQ Diffs Window

## REQ Diff Editor

This section discusses the various tabs found within the REQ Diff Editor when you open a document during a Merge session.

### Outline Editor Tab

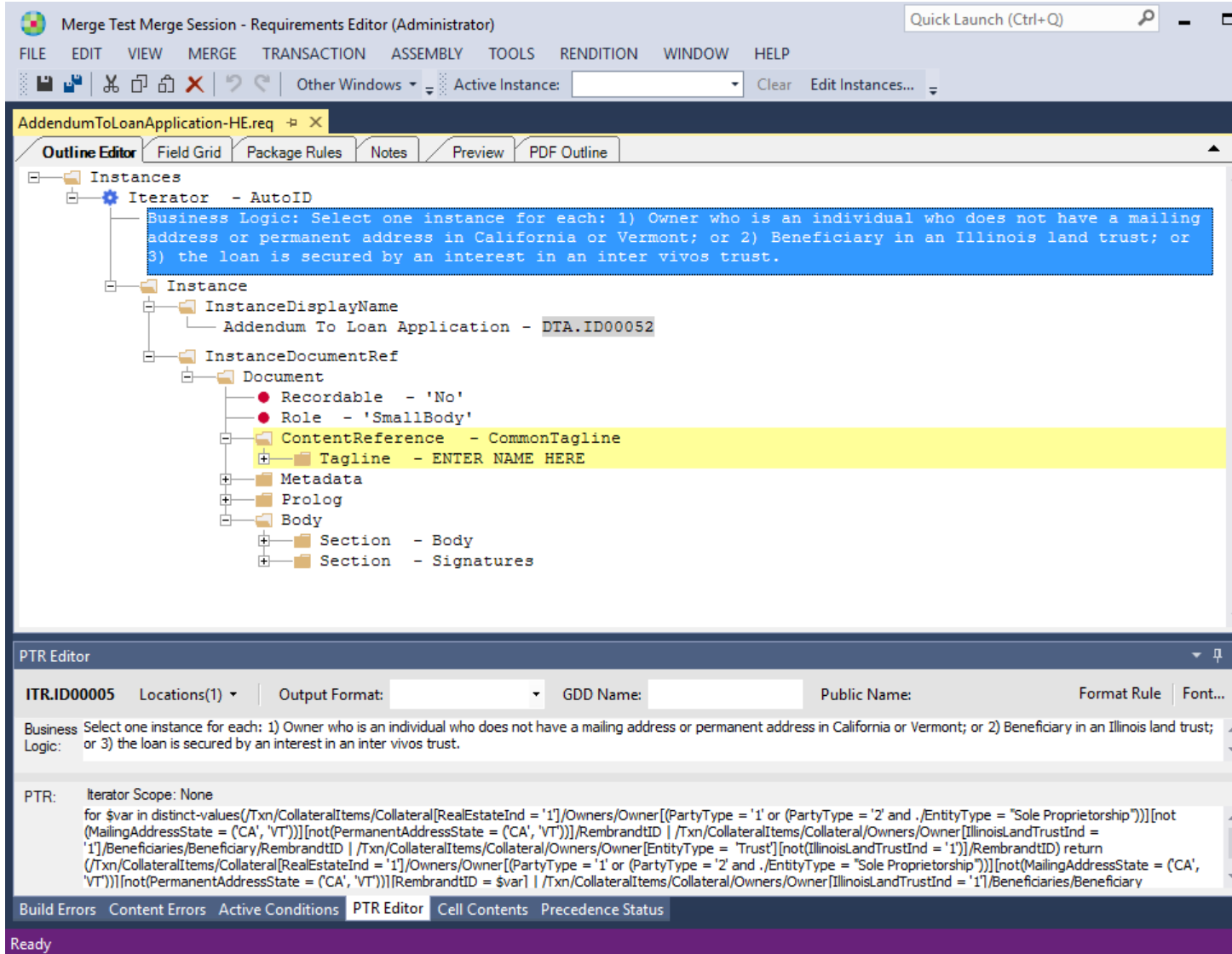
The **Outline Editor** tab is used as the main view to author documents; both in authoring new and editing existing requirements files. This perspective provides an XML Editor experience where the requirements file is presented in an outline based on the XML structure of the file.



Authors utilize this perspective and the main view of a requirements files for authoring and testing. It is here where you can build the file and edit it's contents by adding and removing elements and attributes to adjust the content of the document and to apply formatting requirements to the document.

This window allows you to browse through the document's hierarchy, expand and collapse nodes, as well as edit PTR data within the PTR Editor window.

If the PTR Editor is open and the selected node has associated PTR information, the information is automatically populated within the PTR Editor Window.



## Field Grid Tab

The **Field Grid** perspective allows the author to scroll through all data points (DTAs) and identify any invalid rules or missing GDD names/Public names. This is helpful after the PTRs are complete to be able to quickly scroll through all of the PTRs and identify possible validation issues. Potentially invalid rules will be highlighted in yellow, but note that some rules that will not validate in the field grid are actually valid rules. The Outline Editor perspective is much more accurate for validating PTRs.

	Rule Name	Business Logic	PTR
▶ 1	DTA.ID00005	Checkbox if someone other than the borrower has a homestead interest in the property securing the loan. Map to blank	
2	DTA.ID00006	Checkbox if someone other than the borrower has no homestead interest in the property securing the loan. Map to blank	
3	DTA.ID00039	The address of the collateral.	PRINT ancestor::Collateral/Addresses/Address/StreetAddress ", " ancestor::Collateral/Addresses/Address/City ", " ancestor::Collateral/LocationState " ancestor::Collateral/Addresses/Address/PostalCode
4	DTA.ID00040	Checkbox if borrower is a party to a civil union or registered domestic partnership. Map to blank	
5	DTA.ID00041	Checkbox if borrower is not a party to a civil union or registered domestic partnership. Map to blank	
		Name of someone other than the borrower who is the	

The **Field Grid** perspective allows the author to scroll through all data points (DTAs) and identify any invalid rules or missing GDD names/Public names. This is helpful after the PTRs are complete to be able to quickly scroll through all of the PTRs and identify possible validation issues. Potentially invalid rules will be highlighted in yellow, but note that some rules that will not validate in the field grid are actually valid rules. The Outline Editor perspective is much more accurate for validating PTRs.

The perspective is available by clicking the **Field Grid** tab in the document pane. The Field Grid is available for both dynamic and static documents.

The following values are used to define a data field and are presented in the Field Grid:

- **Rule Name:** A unique ID for each data field entered.
- **Business Logic:** A plain English description of the purpose of the data field, or rule. This is not used for EPM content.
- **Product Technical Rule (PTR):** The PTR represents the processing instructions, or code, that defines the behavior of the rule. This is not used for EPM content.
- **GDD Name:** The data name used for VMP tech/board and E-Forms content.
- **Public Name:** The name used to define the rule for the BSI tech/board and E-Forms content.
- **Substitution Name:** The data name used when completed with values from the Edit Instance screen.
- **Enterprise Name:** This value is not used in current authoring practices.

If a property is editable, clicking in the desired cell will allow you to edit the content.

Outline Editor		Field Grid	Package Rules	Notes	Preview	PDF Outline
	Rule Name	Business Logic			PTR	
▶ 1	DTA.ID00005	Checkbox if someone other than the borrower has a homestead interest in the property securing the loan. Map to blank				
2	DTA.ID00006	Checkbox if someone other than the borrower has no homestead interest in the property securing the loan. Map to blank				
3	DTA.ID00039	The address of the collateral.			PRINT ancestor::Collateral/Addresses/Address/City	PRINT ancestor::Collateral/Addresses/Address/Post
4	DTA.ID00040	Checkbox if borrower is a party to a civil union or registered domestic partnership. Map to blank				

## Package Rules Tab

The **Package Rules** perspective allows the author to define rules to apply to an associated package (also referred to as Auto-Selection and Auto-Suggestion rules) that determine when the document should be selected or suggested based on the transaction data. In this perspective, an author can create new package rules or modify existing ones. A document can have Auto-Selection and Auto-Suggestion rules for each phase that it is in. A separate PTR is written for each selection and suggestion rule.

Package Rule PTRs always start with “IF”, include “THEN” and “SELECT” and end with “ENDIF”. The rules within the Package Rule follow the same Keyword conventions as Condition and Data PTRs. There can be multiple IFs in a Package Rule. For example, the following rule selects the document any time either of the two conditions were met. :

```
IF /Txn/Deposit/IRAPPlans/IRAPlan[Type = '1'] THEN
SELECT
ELSEIF /Txn/Deposit/IRAPPlans/IRAPlan[SelfDirectedInd = '1'] THEN
SELECT
ENDIF
```

The perspective is available by clicking the **Package Rules** tab in the document pane. The Package Rules perspective is available for both dynamic and static documents. This is an example of the field grid perspective in a dynamic document.

Outline Editor   Field Grid <b>Package Rules</b> Notes   Preview   PDF Outline				
	Packages	Selection - Business Logic	Selection - Product Technical Rule	
▶ 1	PKG.HomeEquityApplication PKG.Mortgage2Upfront	Select when: 1) the loan is secured by real estate and there is an individual owner of the real estate who lives in a state other than CA or VT; or 2) the loan is secured by a beneficial interest in an Illinois land trust; or 3) the loan is secured by an interest in an inter vivos trust.	<pre> IF (/Txn/CollateralItems/Collateral[RealEstateInd = '1']/ [(PartyType = '1' or (PartyType = '2' and /EntityType = " [not(MailingAddressState = ('CA', 'VT'))][not(Permanent/ VT))]) or (/Txn/CollateralItems/Collateral/Owners/Owner[ '1'] or (/Txn/CollateralItems/Collateral/Owners/Owner[E (IllinoisLandTrustInd = '1'))]) THEN   SELECT ENDIF           </pre>	
*				

## Notes Tab

The **Notes** perspective allows the author to document activities in the requirements file for both internal-only and external (public) view. The perspective presents a grid of pre-defined values that can be entered by the author to provide reference, guidance, and summary information about authoring activity and changes made to a requirements file. Notes provided in the **Notes** perspective are not converted into content. However, they may be useful to those reviewing requirements files.

**Note:**

The **Note** element can be used to provide documentation at the element level.

Notes are not converted into content; however, they may be useful to those reviewing requirements files. Only Citation notes are viewable by the end customer. When authoring a new .REQ file, typically only Citation Notes and Private Notes are used. Citation notes include the cite dictating the content or format. Private Notes can be used to document notes about a discussion that resulted in the content appearing a certain way, or a reminder for the author or reviewer to handle the content a specific way.

The perspective is available by clicking the **Notes** tab in the document pane. The Notes perspective is available for both dynamic and static documents. This is an example of the notes perspective in a dynamic document.

Outline Editor   Field Grid   Package Rules   <b>Notes</b>   Preview   PDF Outline						
	Author	Note Type	WKFS Revision	Static Form ID	Jurisdiction Type	Jurisdiction
▶ 1	WKFS	Public	08/2012		All States	AK, AL, AR, AZ, CA, CO, CT, DC, DE, FL, GA, HI, IA, ID, IL, IN, KS, KY, LA, MA, MD, ME, MI, MN, MO, MS, MT, NC, ND, NE, NH, NJ, NM, NV, NY, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WV, WY
2	WKFS	Public	09/2012		All States	AK, AL, AR, AZ, CA, CO, CT, DC, DE, FL, GA, HI, IA, ID, IL, IN, KS, KY, LA, MA, MD, ME, MI, MN, MO, MS, MT, NC, ND, NE, NH, NJ, NM, NV, NY, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WV, WY
3	WKFS	Public	08/2012		All States	AK, AL, AR, AZ, CA, CO, CT, DC, DE, FL, GA, HI, IA, ID, IL, IN, KS, KY, LA, MA, MD, ME, MI, MN, MO, MS, MT, NC, ND, NE, NH, NJ, NM, NV, NY, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WV, WY
4	WKFS	Public	08/2012		All States	AK, AL, AR, AZ, CA, CO, CT, DC, DE, FL, GA, HI, IA, ID, IL, IN, KS, KY, LA, MA, MD, ME, MI, MN, MO, MS, MT, NC, ND, NE, NH, NJ, NM, NV, NY, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WV, WY

## Preview Tab

The **Preview** perspective allows the author to view a requirements files in the format that most closely resembles how the file will appear as a generated PDF and includes a listing of all the data points (DTAs) contained in the file as well as any invalid rules or missing GDD names/Public names.

The perspective is available by clicking the **Preview** tab in the document pane. The Preview perspective is available for dynamic documents only. This is an example of the preview perspective in a dynamic document.

Outline Editor   Field Grid   Package Rules   Notes   **Preview**   PDF Outline

---

## Addendum To Loan Application

---

<b>Lender</b>	<b>Owner</b>
DTA.ID00408	DTA.ID00413
DTA.ID00409	DTA.ID00414
DTA.ID00410	DTA.ID00415
DTA.ID00411	DTA.ID00416
DTA.ID00412	DTA.ID00417
	DTA.ID00418
	DTA.ID00419
	DTA.ID00420
	DTA.ID00421, DTA.ID00422 DTA.ID00423

<b>Lender</b>	<b>Owner</b>
DTA.ID00424	DTA.ID00429
DTA.ID00425	DTA.ID00430
DTA.ID00426	DTA.ID00431
DTA.ID00427	DTA.ID00432
DTA.ID00428	DTA.ID00433
	DTA.ID00434
	DTA.ID00435
	DTA.ID00436
	DTA.ID00437, DTA.ID00438 DTA.ID00439

### PDF Outline Tab

The **PDF Outline** perspective allows the author to create a PDF version of the requirements file as it appears in the **Outline Editor**. Included in the PDF outline are all metadata, document information, body sections, package rules, instances, and the field grid. This perspective is often used in testing a requirements file. The document opens in an integrated Adobe Acrobat Reader as a PDF.

The perspective is available by clicking the **PDF Outline** tab in the document pane. The PDF Outline is available for dynamic documents only. This is an example of the PDF outline perspective in a dynamic document.

Outline Editor | Field Grid | Package Rules | Notes | Preview | **PDF Outline**

---

CONFIDENTIAL INFORMATION  
Do not copy or distribute without permission from Wolters Kluwer Financial Services.

---

## Addendum To Loan Application Document Requirements

### Table of Contents

Metadata.....	1
Document Information.....	1
Body.....	6
Signatures.....	7
Package Rules.....	12
Instances Iterator.....	12
Field Grid.....	13

**Key**

- Sections: ***bold italic text***
- Title of sections: Print: **bold text**
- Paragraph: Print: normal-weight text
- Data Item: **[DTA.IDxxx....bold underline text]**
- Condition: **COND.IDxxx:....red text**
- Iterator: **ITR.IDxxx:....red text**
- Note: <Note> citation: or coding:
- List: <List> **Sequenced or Unsequenced**

## Instance Rules Tab

The **Instance Rules** perspective allows the author to create unique instances of the document for a transaction. For example, a unique instance might be used to take a piece of data and apply it for different scenarios in the same form. When certain options are fulfilled, specific documents are selected for the transaction based on the defined instance rules.

The perspective is available by clicking the **Instance Rules** tab in the document pane. The Instance Rules perspective is available exclusively for static documents and is displayed when a static requirements file is opened. This example is from a single instance document.

## Instance Display Rules Tab

The **Instance Display Rules** perspective enables the author to identify cases when the form includes more than one variation (as distinguished from a copy of a form). Each static requirements file has an Instance Display Name and that name appears in a list of documents generated by a transaction. If the document is a multiple instance document, the instance display name is followed by a data item that determines which type of identifying information should print after the instance display name. The

additional identifying information is typically an account number, borrower name, or some other piece of data that will distinguish one instance of the document from another.

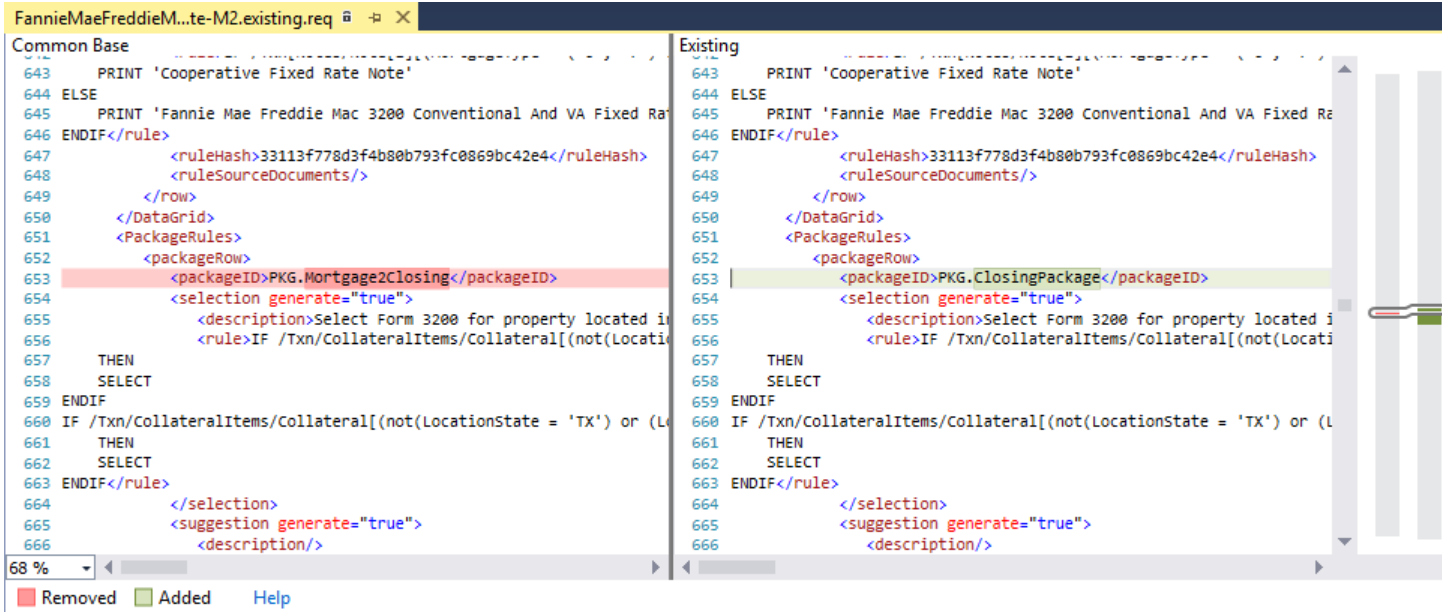
The perspective is available by clicking the **Instance Display Rules** tab in the document pane. The Instance Display Rules perspective is available exclusively for static documents and is displayed when a static requirements file is opened.

Form View		Field Grid*		Package Rules		Instance Rules		<b>Instance Display Rules</b>		Notes	
		<b>Business Logic</b>				<b>PTR</b>					
▶	1	Print HUD 92005 Veterans Affairs 26-1852 Description of Materials.				PRINT "HUD 92005 Veterans Affairs 26-1852 Description of Materials"					

## Working with Merge Session Documents

### Compare with Common Base

Selecting the context menu item **Compare with Common Base** from the [REQ Diffs Window](#) allows you to compare the existing customized version of the selected document with the Common Base version.



```

Common Base
643 PRINT 'Cooperative Fixed Rate Note'
644 ELSE
645 PRINT 'Fannie Mae Freddie Mac 3200 Conventional And VA Fixed Ra
646 ENDIF</rule>
647 <ruleHash>33113f778d3f4b80b793fc0869bc42e4</ruleHash>
648 <ruleSourceDocuments/>
649 </row>
650 </DataGrid>
651 <PackageRules>
652 <packageRow>
653 <packageID>PKG.Mortgage2Closing</packageID>
654 <selection generate="true">
655 <description>Select Form 3200 for property located i
656 <rule>IF /Txn/CollateralItems/Collateral[(not(Locati
657 THEN
658 SELECT
659 ENDIF
660 IF /Txn/CollateralItems/Collateral[(not(LocationState = 'TX') or (L
661 THEN
662 SELECT
663 ENDIF</rule>
664 </selection>
665 <suggestion generate="true">
666 <description/>

Existing
643 PRINT 'Cooperative Fixed Rate Note'
644 ELSE
645 PRINT 'Fannie Mae Freddie Mac 3200 Conventional And VA Fixed Ra
646 ENDIF</rule>
647 <ruleHash>33113f778d3f4b80b793fc0869bc42e4</ruleHash>
648 <ruleSourceDocuments/>
649 </row>
650 </DataGrid>
651 <PackageRules>
652 <packageRow>
653 <packageID>PKG.ClosingPackage</packageID>
654 <selection generate="true">
655 <description>Select Form 3200 for property located i
656 <rule>IF /Txn/CollateralItems/Collateral[(not(Locati
657 THEN
658 SELECT
659 ENDIF
660 IF /Txn/CollateralItems/Collateral[(not(LocationState = 'TX') or (L
661 THEN
662 SELECT
663 ENDIF</rule>
664 </selection>
665 <suggestion generate="true">
666 <description/>

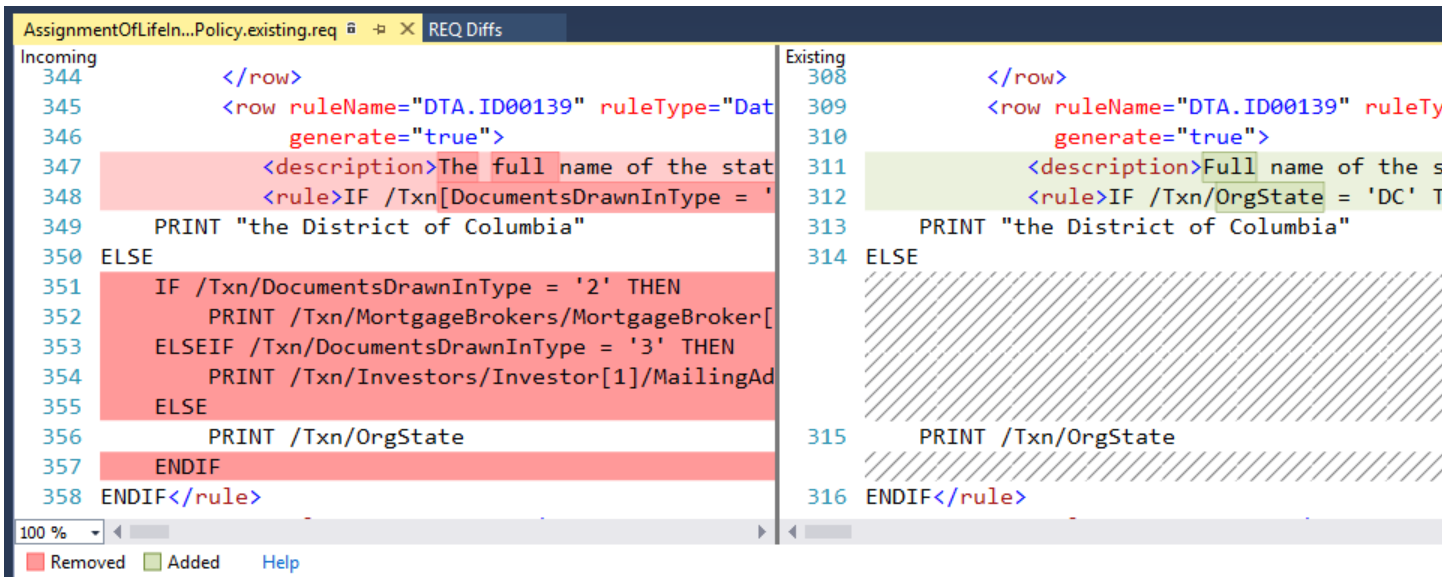
```

The selected document is displayed with the Common Base version on the left side of the window and the Existing version on the right.

The synchronized scroll bars on the right side of the window provide a visual clue as to locations within each version that contain identified differences.

## Compare with Incoming

Selecting the context menu item **Compare with Incoming** from the [REQ Diffs Window](#) allows you to compare the existing customized version of the selected document with the Incoming version.



```

AssignmentOfLifel...Policy.existing.req REQ Diffs
Incoming
344 </row>
345 <row ruleName="DTA.ID00139" ruleType="Dat
346 generate="true">
347 <description>The full name of the stat
348 <rule>IF /Txn[DocumentsDrawnInType = '
349 PRINT "the District of Columbia"
350 ELSE
351 IF /Txn/DocumentsDrawnInType = '2' THEN
352 PRINT /Txn/MortgageBrokers/MortgageBroker[
353 ELSEIF /Txn/DocumentsDrawnInType = '3' THEN
354 PRINT /Txn/Investors/Investor[1]/MailingAd
355 ELSE
356 PRINT /Txn/OrgState
357 ENDIF
358 ENDIF</rule>

Existing
308 </row>
309 <row ruleName="DTA.ID00139" ruleTy
310 generate="true">
311 <description>Full name of the s
312 <rule>IF /Txn/OrgState = 'DC' T
313 PRINT "the District of Columbia"
314 ELSE
315 PRINT /Txn/OrgState
316 ENDIF</rule>

```

The selected document is displayed with the Incoming version on the left side of the window and the Existing version on the right.

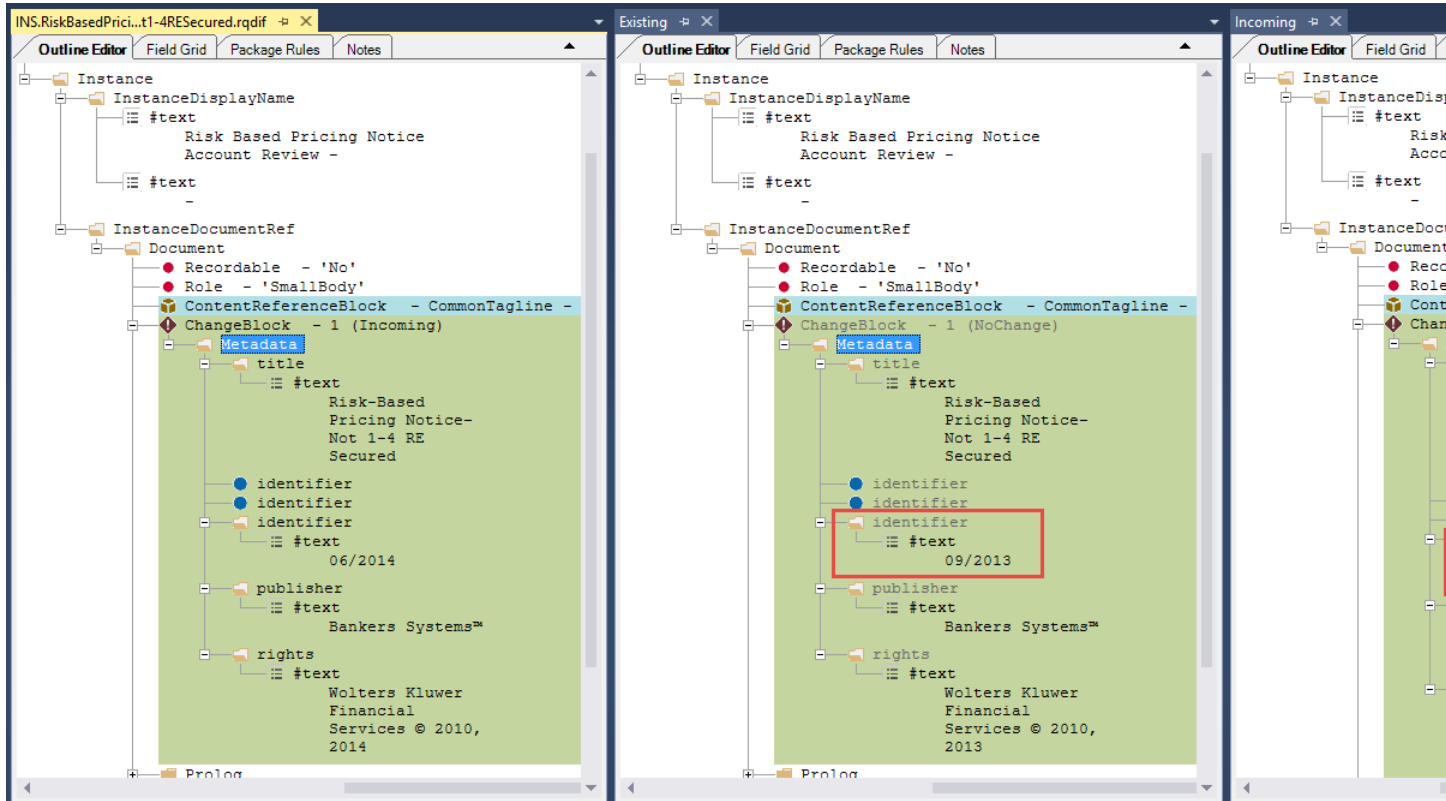
The synchronized scroll bars on the right side of the window provide a visual clue as to locations within each version that contain identified differences.

### Three Way Merge

Selecting Compare with Common Base from the content menu of the REQ Diffs window allows you to view the differences of each version of the document in the following windows:

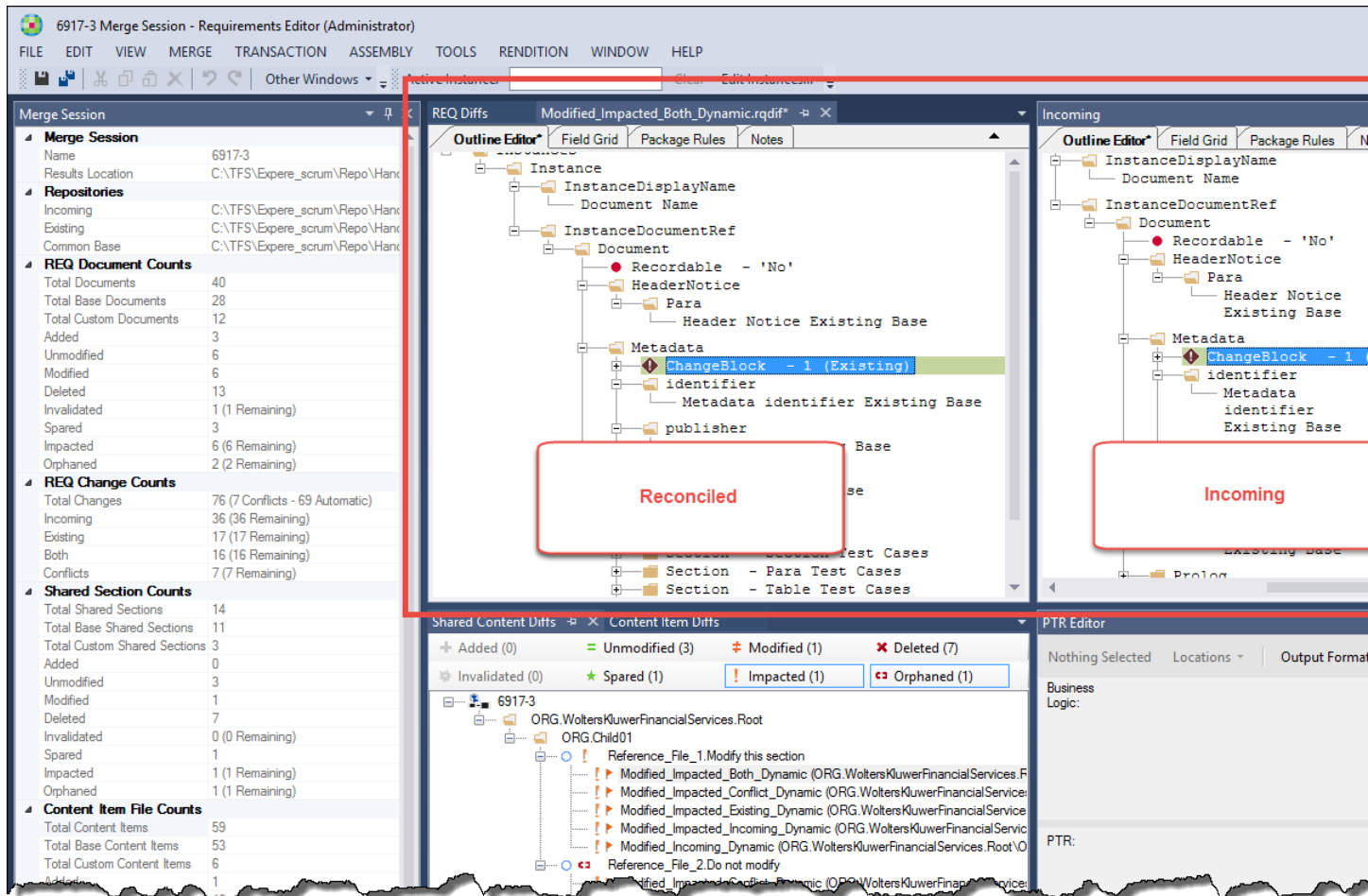
- **Incoming:** shows the incoming version of the document
- **Existing:** shows the existing version of the document
- **rqdif window:** displays the selected document and dynamically changes based upon the items selected or rejected in the Incoming and Existing windows

**Figure2. Example of Three-Way Merge**



## Accepting and Rejecting Changes

To accept or reject changes within the **Outline Editor** and **Field Grid** tabs, it is helpful to have all three versions displayed. The following example shows the Reconciled, Incoming and Existing versions in a side-by-side layout.



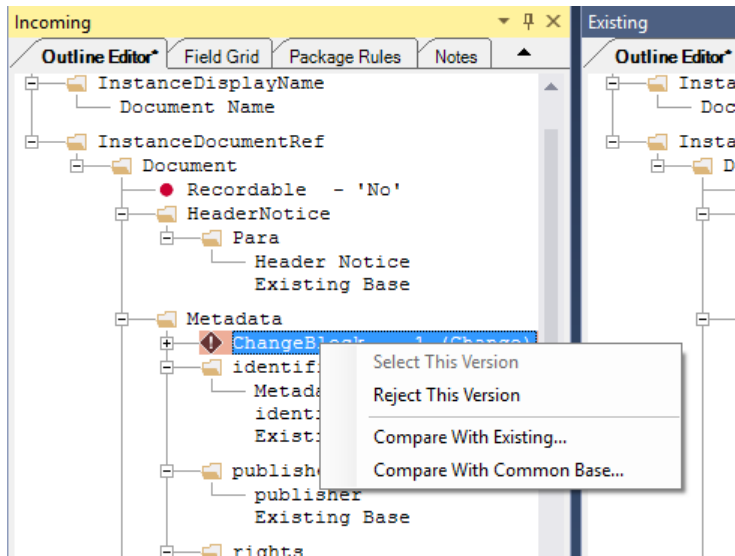
The screenshot displays the Requirements Editor interface during a merge session. On the left, a 'Merge Session' pane shows details for session 6917-3, including repository locations and document counts. The main workspace is divided into three vertical panes:

- Left Pane (Reconciled):** Shows the 'Reconciled' version of the change. A red box highlights the 'ChangeBlock - 1 (Existing)' in the 'Metadata' section.
- Middle Pane (Incoming):** Shows the 'Incoming' version of the change. A red box highlights the 'ChangeBlock - 1' in the 'Metadata' section.
- Right Pane (Existing):** Shows the 'Existing' version of the change. A red box highlights the 'ChangeBlock - 1' in the 'Metadata' section.

At the bottom, a 'Shared Content Diiffs' pane shows a tree view of content items, including 'ORG.Child01' and 'Reference\_File\_1.Modify this section'. A 'PTR Editor' pane is also visible on the right side.

The features built into Merge functionality allows you to select whichever versions of the selected change you wish. This is done by right-clicking the Incoming or Existing instance of the change and selecting **Select This Version** or **Reject This Version**. When selecting a version from the Existing or Incoming windows, the associated ChangeBlock in the RQDIF window displays the result of your selection (see table below).

Note that when a version is selected, the **Select This Version** option is disabled and the **Reject This Version** option is enabled, allowing you the change your selection.



The following table describes the possible options for selections in the Incoming and Existing versions, and their impact on the Reconciled version.

Existing Version	Incoming Version	Reconciled Version
No selection	No selection	Takes the CommonBase version
Select This Version	Select This version	Takes both versions
Reject This Version	Reject This Version	Accepts the default action. If an automatic change, it automatically takes the existing or incoming, based on the instance. If a conflict, it defaults to the existing version.
Select This Version	Reject this Version	Takes existing version
Reject This Version	Select This Version	Takes incoming version

**Comment [Kel.Mankenberg]:** a few rows have been commented in the table above. They appeared to be nonapplicable to the topic.

When finished selecting the versions to include, right-click the Reconciled version and select **Mark As Resolved**. This disables all selection options from the Existing and Incoming versions.

**Note:**

If you need to change your selections after you have marked the Reconciled as resolved, you can right-click the Reconciled version and select **Mark As Pending**. This makes all versions selectable.

## Merge Menu

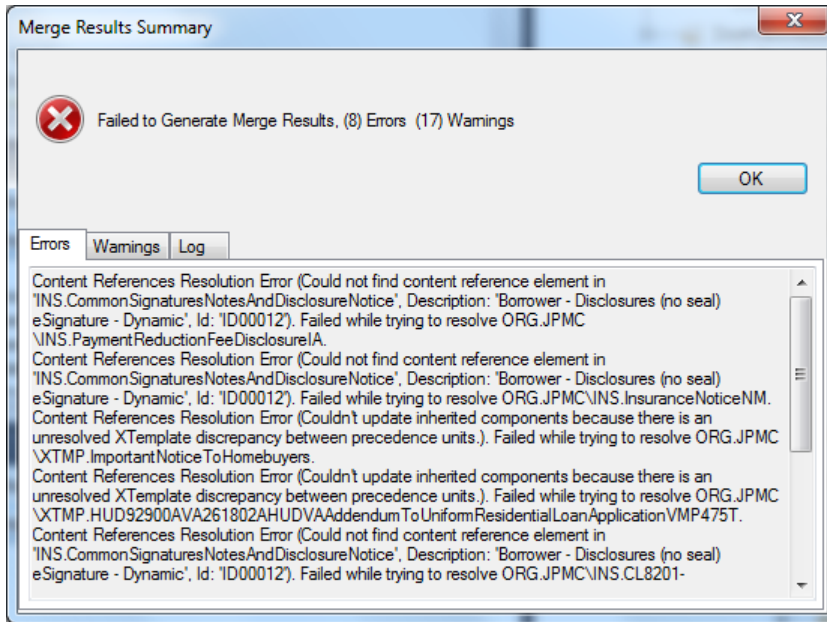
The **Merge** menu provides access to the following commands:

- Generate Results
- Generate Impact Analysis Reports

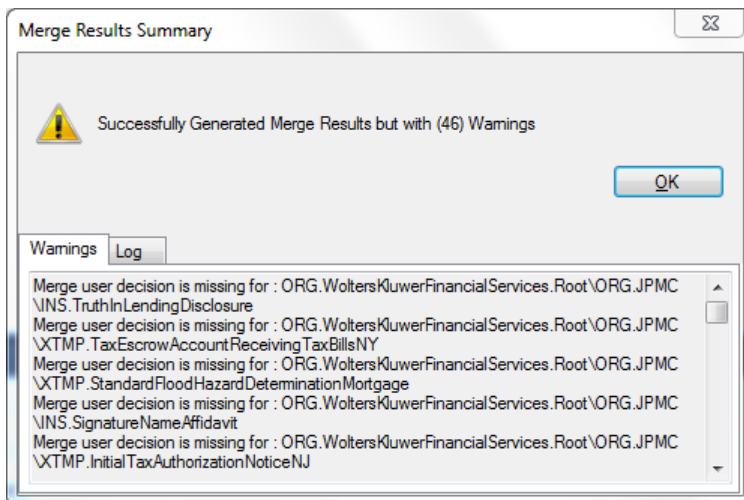
### Generate Results

The **Merge > Generate Results** command allows you to generate your final repository from all selected changes.

If there are items within the merge session that require attention, you will be prompted with a summary dialog.

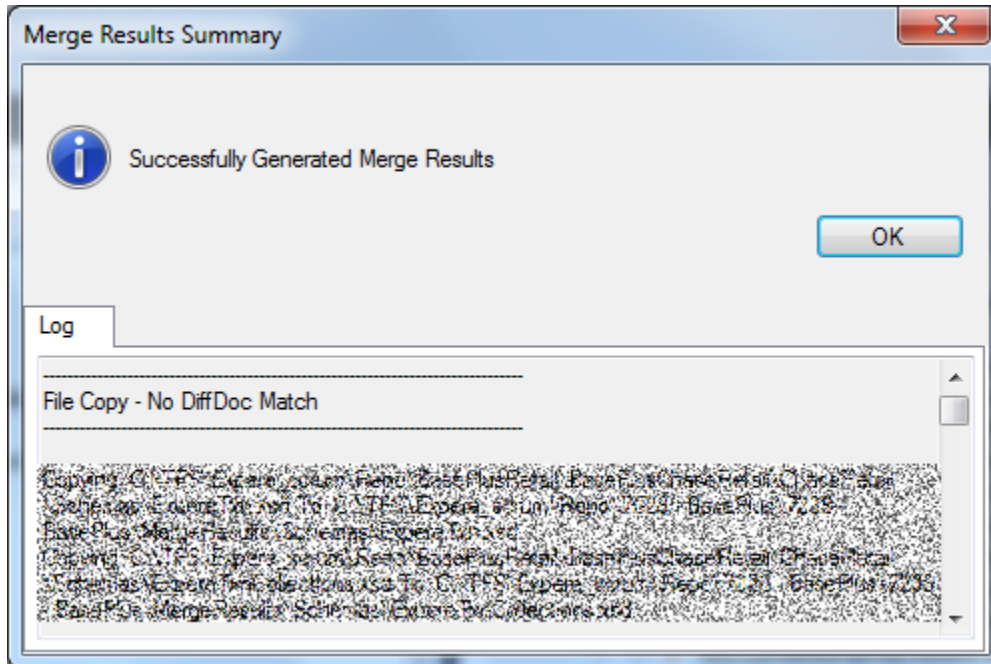


If the merge is successful but user decision items remain, a dialog similar to the following is displayed:



The tabs in the dialog display the errors and warnings found during the merge process and allow you to copy and paste the results into a text editor for reference when further working in your Merge Session.

When all errors and warnings are dealt with, a message is displayed indicating successful generation of the merged repository. Customer information in the following example has been masked.



### Generate Impact Analysis Report

Selecting **Merge > Generate Analysis Report** generates a multi-page HTML analysis report containing Change Counts (summaries) and REQ Change Details. The report is displayed in your web browser upon successful completion.

Change Counts enumerate the changes to REQ Documents, Shared Sections and Content Items, as shown in the following example.

## Impact Analysis Report

### Change Counts

#### REQ Documents

REQ DOCUMENT COUNTS		
Documents	Count	Remaining
Total	963	
Total Base	555	
Custom	408	
Added	11	
Modified	110	
Deleted	10	
Invalidated		
Spared	354	0
Impacted	49	
Orphaned	5	

REQ CHANGE COUNTS		
Change	Count	Remaining
Total	4969	0
Incoming	192	
Existing	4224	
Both	259	
Conflicts	294	
Automatic	4675	

#### Shared Sections

SHARED SECTION TOTALS		
Section	Count	Remaining
Total	130	
Base	84	
Custom	46	

SHARED SECTION CHANGE COUNTS		
Change	Count	Remaining
Added	46	
Modified	42	
Deleted	0	
Invalidated		
Spared	32	
Impacted	14	
Orphaned	0	









#### Content Items

CONTENT ITEM FILE COUNTS		
Files	Count	Remaining
Total	184	
Total Base	57	
Custom	127	
Added	1	
Modified	9	
Deleted	0	
Invalidated		
Spared	125	
Impacted	2	
Orphaned	0	

CONTENT ITEM CHANGE COUNTS		
Change	Count	Remaining
Total	4969	
Incoming	192	
Existing	4224	
Both	259	
Conflicts	294	
Automatic	4675	

The REQ Change Details section provides links to HTML pages containing details about the changes of individual REQ changes.

## REQ Change Details

ADDED	INVALIDATED
<a href="#">INS.VeteransAffairsOriginationStatement-DF</a>	
<a href="#">INS.ThirdPartyFeesDisclosure-DF</a>	
<a href="#">INS.ResidentialLoanFeeAgreement-LA</a>	
<a href="#">INS.ResidentialLoanFeeAgreement-LA-DF</a>	
<a href="#">INS.MortgageLoanCommitment-DF</a>	
<a href="#">INS.LockAgreementVA</a>	
<a href="#">INS.LockAgreementMN</a>	
<a href="#">INS.IntentToProceed</a>	
<a href="#">XTMP.InitialTaxAuthorizationNoticeNI-DF</a>	
<a href="#">INS.FloodInsuranceDisclosureMA</a>	
<a href="#">XTMP.AuthorizationToReleaseSSNVerification</a>	
DELETED	ORPHANED
<a href="#">INS.HighCostHomeLoanNoticeToBorrowerFL</a>	
<a href="#">INS.FannieMae3535InterestOnlyARMNote1YearLIBORIndex10YearInterestOnlyPeriod</a>	<a href="#">ORG </a>
<a href="#">INS.FairLendingActHighCostHomeLoanPrepaymentPenaltyDisclosureFL</a>	
<a href="#">INS.FannieMae3153FixedARMRiderWS1YearLIBOR10YearInterestOnly</a>	<a href="#">ORG </a>
<a href="#">INS.FannieMae3271InterestOnlyPeriodFixedRateNote</a>	<a href="#">ORG </a>
<a href="#">INS.HighCostHomeLoanNoticeToPurchasersAndAssigneesFL</a>	
<a href="#">INS.FannieMae3530And3531InterestOnlyAdjustableRateNoteOneYearTreasuryAndLIBORIndexWithRateCaps</a>	<a href="#">ORG </a>
<a href="#">INS.FHAFixedRateNote0115</a>	<a href="#">ORG </a>
<a href="#">INS.ResidentialLoanFeeAgreementLA</a>	
<a href="#">INS.NoticeOfAssignmentSaleOrTransferOfServicingRights</a>	
MODIFIED	IMPACTED
<a href="#">INS.ServicingDisclosureStatement</a>	
<a href="#">INS.FHAARMNote</a>	<a href="#">ORG </a>
<a href="#">INS.LockInAgreementMS</a>	
<a href="#">INS.ApplicationDisclosureNI</a>	
<a href="#">INS.FannieMaeFreddieMac3150PUDRider</a>	<a href="#">ORG </a>
<a href="#">INS.EscrowAccountDisclosureOH</a>	
<a href="#">INS.CounselingNoticeMD</a>	<a href="#">ORG </a>
<a href="#">INS.BorrowerOptInDisclosureMA</a>	
<a href="#">INS.TangibleNetBenefitWorksheetMA</a>	

## Org.Filename Details

### Notes

[← BACK TO REPORT](#)

Area	ID	Name	Action	Notes
DataGridRule	name:Lender.FullNameStree tCityStateZip:fuid:1:page:1		Automatic	
Section	COND.ID00026	Collateral Document Acknowledgment Block	Automatic	

## General Authoring

This section describes tasks that are performed in both dynamic and static document authoring.

### Working with Tables

This section demonstrates how to create and work with table structures in the outline editor perspective.

#### Create a Table

Authors typically create tables from within a Prolog, Body, or Section parent element.

1. Select the parent element, right-click and select **Table**.
2. The **Create Table** dialog appears. Determine the number of **Rows** and **Columns** in the respective fields.
3. In the **Table Borders** section, perform the following:
  - To add a border to the table, click the **Location** dropdown and select one of the following:
    - None
    - All
    - Bottom
    - Top
    - Sides
    - Top and Bottom
  - Click the **Weight** dropdown and select the desired thickness:
    - 0.5
    - 1.0
    - 1.5
    - 2.0
    - 2.5
    - 3.0
4. In the **Cell Properties** section, determine the following:
  - Font Size
  - Background Color
  - Horizontal Alignment

- Vertical Alignment

5. The **Cell Borders** section allows users to determine the thickness of cell borders; select one of the following:

- None
- Grid Lines
- Specific Sides

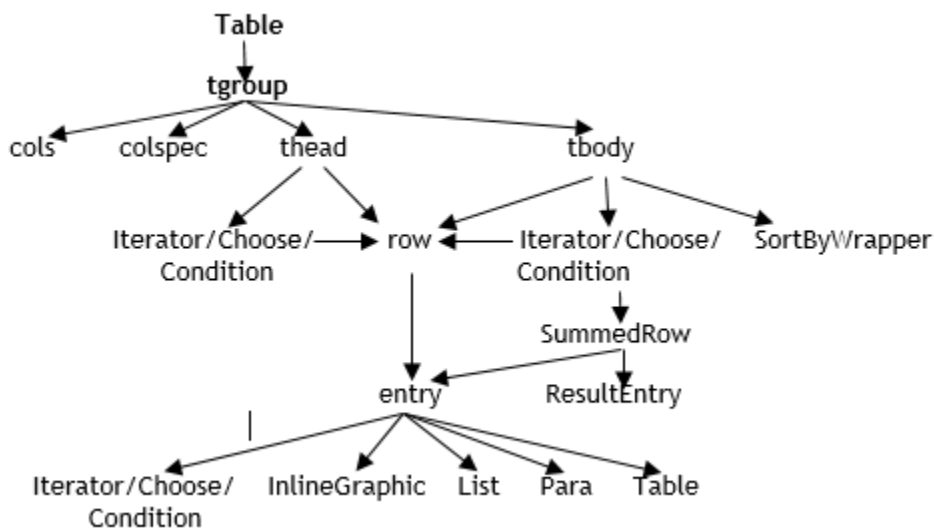
## Table Elements and Attributes

The Table element may be used to for both layout and data (tabular) tables. Layout tables are used to structure the page layout and for specific spacing requirements. Data tables are used to present information in a tabular format with cell contents that map to logical column and row headers.

**Note:**

If you have purchased the Tagged PDF Outputs add-on license, refer to [Accessibility in Tagged PDF Outputs Implementation Guide](#) for details on how to author tables to generate proper accessibility tags.

Typical child elements within a **Table** element consist of the following:



### Table Elements

Element	Description
table	<p>The <b>Table</b> element contains attributes that defines the overall structure of the table such as frame/borders, border weight, font size, etc. A <b>Table</b> element may contain multiple <b>tgroup</b> elements and may be added to an entry element to create a nested table structure.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p><b>Note:</b>            If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring tables.</p> </div>
tgroup	<p>The <b>tgroup</b> element is the container to define the table contents. Add the <b>tgroup</b> element as a child of the <b>Table</b> element. Child elements and attributes may be added to the <b>tgroup</b> element to define number of columns, column width, alignment, etc. Add the <b>thead</b> element as a child to further define the table columns. Add the <b>tbody</b> element as a child to define the cell contents. Note: multiple <b>tgroup</b> elements may be added to a given table.</p>
thead	<p>The <b>thead</b> element is the container for rows that are the column headings.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p><b>Note:</b>            If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring the <b>thead</b> element.</p> </div>
tbody	<p>The <b>tbody</b> element is the container for the body of the table. Add the <b>row&gt;entry</b> element as a child of the <b>tbody</b> to content for each row/cell.</p>
Iterator/Choose/Condition	<p>These elements may also be used as a child of the <b>thead</b> and <b>tbody</b> table elements.</p> <p>See <a href="#">Using Rules</a> for information on how to author these elements.</p>
row	<p>The <b>row</b> element is the container for the all cells in the row which are represented as entries.</p>
entry	<p>An <b>entry</b> element may be added as a child of the <b>row</b> element. The <b>entry</b> element defines the data/content for the table. Each entry under a row represents the content for each column; the first entry is the first column, the second entry is the second column, etc.</p>
SortByWrapper/SummedRow/ResultEntry	<p>See <a href="#">Defining the Number of Rows (SortByWrapper Element)</a> for further information.</p>

### Table Element Attributes

The following attributes are the most common available for table elements and offer some control over how the stylesheet will format the content:

Element	Attribute	Description
table	FontSize	This attribute allows the user to specify the font size explicitly (measured in points). This may be needed when a regulatory rule requires a specific font size be used (i.e. 16pt).
	frame/FrameWeight	These attributes allows the user to specify the frame (borders) and border weight explicitly for the entire table. If the frame is not authored, the default applied is none. The following options are available for the frame attribute: top, bottom, topbot, all, sides, none. The FrameWeight attribute may be specified in either points or inches (i.e. .05in, 1.5pt)
	ReadOrder	<div style="border: 1px solid green; padding: 5px;"> <p><b>Note:</b> If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring the <b>ReadOrder</b> element.</p> </div>
tgroup	cols	Specify the total number of columns for the table.
tgroup/colspec	colwidth	Specify a colspec element for each column to further define the column details. Add the colwidth attribute to each colspec element to define the amount of space each column will reserve. Enter the desired value using inches (1in) or use the asterisk (1.*) to evenly divide the columns based on the total page width. Each colwidth may be different but the total combined colwidths cannot exceed 7.5 inches.

Element	Attribute	Description
row	RowHeight	This attribute allows the user to specify the height of a row explicitly (measured in inches).
entry	align	Controls the horizontal alignment of the cells contents. Possible values: left, right, center, justify, char. Default alignment is left.
	BackgroundColor	Sets the background color or shading of the cell. The following color options are available: Black, Gray, Red, LightGray, IntDiscGray, MetadataValue. Selecting 'MetadataValue' will use the color specified in the stylesheet element InstanceFormattingMetadata/ TableCellBackgroundColor element.
	TopBorder/RightBorder/ BottomBorder/LeftBorder  *BorderColor / *Weight	There are four sets of attributes for each border side (*top, right, bottom and left) and each attribute has a corresponding color and weight attribute. Use these attributes to draw border lines around a cell; use the weight will specify the thickness of the line in points (pt).

Element	Attribute	Description
	colspan	<p>Specifies the number of columns a given cell spans. The spanning column entries are not authored. For example, assume the table has a total of 3 columns and the first column spans both columns 1 &amp; 2. The first entry with the colspan attribute represents two columns; therefore, only one additional entry is authored for the third column.</p> <div data-bbox="1073 701 1341 1108" style="border: 1px solid black; padding: 5px;"> <p><b>Note:</b> If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring the colspan element.</p> </div>
	FontSize	<p>This attribute allows the user to specify the font size explicitly (measured in points). This may be needed when a regulatory rule requires a specific font size be used (i.e. 16pt).</p>
	padding	<p>Allows user to override default cell padding margins. The default is 1pt for top, right, bottom and left. The attribute value as string of four spaced points in the following format: 1pt, 2pt, 3pt, 4pt (where 1 represents top, 2 is right, 3 is bottom, and 4 if left). If the default margin only needs to change for one side, all four values must still be passed using the 1pt for the others that do not need to change (0pt, 1pt, 1pt, 1pt).</p>

Element	Attribute	Description
	rowhead	<div data-bbox="1076 352 1341 758" style="border: 1px solid black; padding: 5px;"> <p><b>Note:</b> If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring the <b>rowhead</b> element.</p> </div>
	rowspan	<p>Specifies the number of rows a given cell spans. The spanning row entries are not authored. For example, assume the table has a total of 3 columns and 2 rows. The first row's column entry spans rows 1 &amp; 2. The first row's column entry with the rowspan attribute represents two rows; therefore, the second row would only contain 2 entries for the 2nd and 3rd column cells of that row as the value for the first column cell is rowspan contents.</p> <div data-bbox="1076 1234 1341 1640" style="border: 1px solid black; padding: 5px;"> <p><b>Note:</b> If you have purchased the Tagged PDF Outputs add-on license, refer to <a href="#">Accessibility in Tagged PDF Outputs Implementation Guide</a> for details on authoring the <b>rowspan</b> element.</p> </div>
	SortByWrapper/SummedRow/ResultEntry	See <a href="#">Defining the Number of Rows (SortByWrapper Element)</a> for further information.
	valign	This attribute controls the vertical alignment of the cells contents. Possible values: top, bottom, middle. The default is top.

## Customizing the Header, Rows, and Borders within a Table

Enter topic summary (Optional) - remove all text if not used

Users can customize the header, rows, and borders within a table through the use of a custom stylesheet. After configuring the custom stylesheet, users can now specify table shading through the **TableStyle** option.

**Note:**

The font-size specified for the table styles below is what will actually be used. Normally adjustments are performed on certain fonts to make them larger than specified if the font does not meet legal size requirements but that adjustment will not be done for custom tables.

1. Within the

Custom\_WKFS\_Styleguide.xsl file, navigate to the Tables section. Within each style <x> table section modify the following:

- style<x>-header-attrs - font, color, background color, font weight, font style, text decoration, and font size variables for the table header.
- style<x>-body-attrs - font size, color, and font family variables for the table rows.
- style<x>-border-attrs - border size, style, and width variables for the table borders.

```
<!-- Begin options for style1 table -->
<xsl:attribute-set name="style1-header-attrs">
  <xsl:attribute name="font-family">Times New Roman</xsl:attribute>
  <xsl:attribute name="color">blue</xsl:attribute>
  <xsl:attribute name="background-color">pink</xsl:attribute>
  <xsl:attribute name="font-weight">normal</xsl:attribute>
  <xsl:attribute name="font-style">italic</xsl:attribute>
  <xsl:attribute name="text-decoration">underline</xsl:attribute>
  <xsl:attribute name="font-size">16pt</xsl:attribute>
</xsl:attribute-set>

<xsl:attribute-set name="style1-body-attrs">
  <xsl:attribute name="font-size">13pt</xsl:attribute>
  <xsl:attribute name="color">purple</xsl:attribute>
  <xsl:attribute name="font-family">Comic</xsl:attribute>
</xsl:attribute-set>

<xsl:template name="style1-border-attrs">
  <xsl:attribute name="border-color">red</xsl:attribute>
  <xsl:attribute name="border-style">dotted</xsl:attribute>
  <xsl:attribute name="border-width">1.25pt</xsl:attribute>
</xsl:template>

<xsl:variable name="style1-TableBodyEvenRowBackgroundColor" select="'green'" />
<xsl:variable name="style1-TableBodyOddRowBackgroundColor" select="'yellow'" />
<!-- End options for style1 table -->
```

2. Save the changes.

3. Within the Requirements Editor, create a table. On the **Table** element, right-click and select **Add Attribute/TableStyle**.

4. Within the **TableStyle** attribute, double-click **No Value** and select the desired **Style** (as created in the `Custom_WKFS_Styleguide.xml` file). Return to the `.xml` file to make any further updates to a particular **TableStyle** style.

**Note:**

The particular **TableStyle** selection will override other attributes selected elsewhere in the **table** element: font, weight, color, etc.

## Changing Border Colors in a Table

You have a requirement to apply non-default (black) color to the border of one or more table cells in a table. The target REQ file is open in the Requirements Editor.

Some documents, specifically those related to TILA/RESPA, require the table border (and table cell borders) to be gray color and also require select cell borders to be light gray (hex: #CDCDCD) while others to maintain the black color (the default color is black). The border color of an individual table cell (top, left, bottom, right) is defined by setting the hex value using one of the following attributes: *TopBorderColor*, *LeftBorderColor*, *BottomBorderColor*, and *RightBorderColor*. This are enabled when the respective Border attribute is set to **Yes**. For example, you must set **RightBorder** to Yes and then define **RightBorderColor**. The suggested HEX value for light gray is #CDCDCD.

1. In a table, select a table cell.
2. With the table cell (*entry* element) active selection, right-click and select **AddAttribute**.
3. In the **Attributes** list, select either the *RightBorder*, *LeftBorder*, *TopBorder*, or *BottomBorder* attributes and ensure the value is **Yes**.
4. In the **Attributes** list, select the corresponding color attribute (*RightBorderColor*, *LeftBorderColor*, *TopBorderColor*, or *BottomBorderColor*).
5. In the **-Color** attribute, double click the attribute and change the attribute value (*No Value*) to the desired HEX value (light gray = #CDCDCD).

Color changes using these attributes must be entered as HEX values; there is no option to use the color palette or non-HEX entity values.

## Defining the Number of Rows (SortByWrapper Element)

The *SortByWrapper* element is used to set the number of disbursement lines appearing on a generated page. The element iterates over the number of payments and then totals them together; attributes on the element then define how the totals are presented on the generated output.

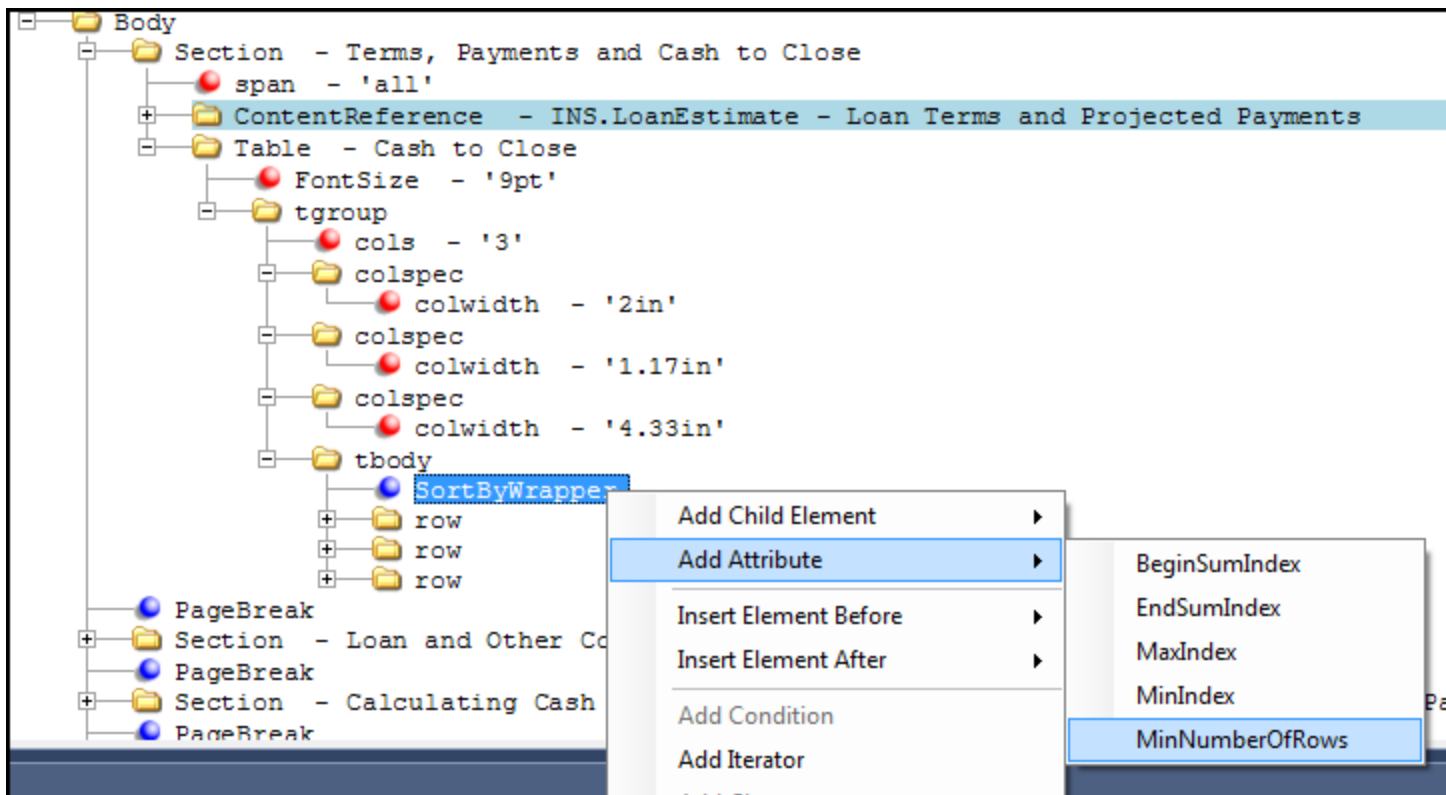
A use case for this behavior is a requirement for a total of fifteen rows of Payoffs and Payments to be included on page 3 of *Closing Disclosures*, even if the transaction has less than fifteen disbursements, to comply with TILA/RESPA regulations.

**Note:**

If the `/Txn/Notes/Note[1]/DisbursementToOthersAmount` is passed in the transaction data, the payments lines are set to show 15 lines total by default. The `SortByWrapper` element mimics that behavior for situations where `DisbursementToOthersAmount` is not passed in the transaction data.

The `SortByWrapper` element is available as a child element on a table body (`tbody`) element using the **Add Child Element** option or can be added before or after a row element (`row`) using the **Insert Element After/Before** options. The element can also be added as a child to an `Iterator` element.

The `SortByWrapper` element comprises five attributes; three attributes that can be used to determine how to present the payment lines and two attributes defining how to sum entries in the table.



**Note:**

The `BeginSumIndex`, `MinIndex`, and `MaxIndex` are based off the iterator index, not the overall index.

- **BeginSumIndex:** When a value is supplied to this attribute, it defines how to sum additional items within a table. This value should be one (1) greater than the `MaxIndex` value. See [Summing tables](#)
- **EndSumIndex:** When a value is supplied to this attribute, it defines the how to sum the values that appear in the table, even if additional values have been passed. For example, if 15 items have been passed but only 10 rows appear in the table, set the value to 10. The `EndSumIndex` value should be greater than `BeginSumIndex` value and less than or equal to the `MaxIndex` value. See [Summing Values within a Table](#)

- **MaxIndex:** When a value is supplied to this attribute, it defines the highest index number (maximum number of rows) to render on the table. Any rows with an index higher than the *MaxIndex* value are truncated and not rendered in the generated document.
- **MinIndex:** When a value is supplied to this attribute, it defines the lowest possible index number (number of rows) to render on the table. Any row with an index less than the *MinIndex* value is truncated and not rendered in the generated document.
- **MinNumberOfRows:** When a value is supplied to this attribute, it defines the minimum number of rows that will be generated. If the attribute is not used, the *MaxIndex* number is used. If the attribute is supplied with a value *greater than* the number of rows naturally generated by the iterator then extra blank rows are added in the output. If the attribute is supplied with a value *less than or equal* to the number of rows naturally generated by the iterator, then no rows are appended.

**Note:**

This attribute is used when at least one item is passed; if no items are passed then you must create a condition to print all blank lines.

## Defining the Minimum Number of Rows

This section explains how to use the **MinNumberOfRows** attribute to define a minimum number of rows generated by the **SortByWrapper** element.

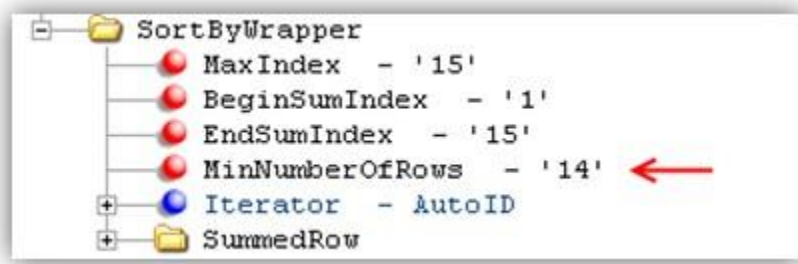
**Note:**

When the **MinNumberOfRows** attribute is not used, there is no change to the behavior of the **SortByWrapper** element. For example, if the **SortByWrapper**'s iterator generates 13 rows, the absence of the **MinNumberOfRows** will not change the result. The table will contain as many or as few rows as the iterator itself generates.

The following example

1. From the **Table/tgroup/tbody/SortByWrapper** element, right click and select **Add Attribute/MinNumberOfRows**.
2. From the **MinNumberOfRows** attribute, double-click on the **NO VALUE** text field. Enter a value; for example: 14.

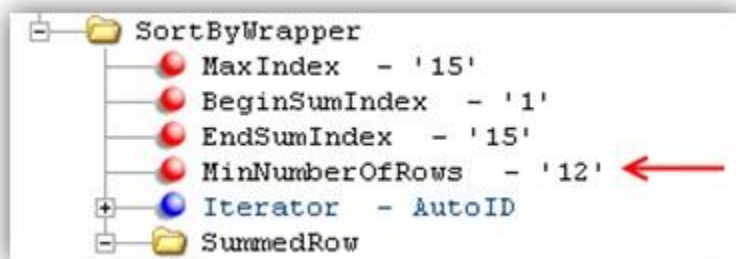
When the **MinNumberOfRows** is supplied with a value greater than the number of rows naturally generated by the iterator, then extra blank rows are appended. For example when **MinNumberOfRows** is 14:



then the result looks like this:

12 S Desc D2	\$20.20
13 W Desc D12	\$120.20
14	Extra row added here
K. TOTAL PAYOFFS AND PAYMENTS	

When the **MinNumberOfRows** is supplied with a value less than or equal to the number of rows naturally generate by the iterator then no rows are appended. The resulting table is identical to the table generated as if the attribute was not supplied at all. For example when **MinNumberOfRows** is 12:



then the result looks like this:

12 S Desc D2	The iterator generated these 13 rows.	\$20.20
13 W Desc D12		\$120.20
K. TOTAL PAYOFFS AND PAYMENTS		\$915.00

### Summing Values within a Table

This section explains how to only sum the values that appear within a table, even if additional values have been passed. For example, if 15 items have been passed but only 10 rows appear in the table, set the **EndSumIndex** attribute to 10. This procedure also assumes that you have created a table.

1. From the **Table/tgroup/tbody/SortByWrapper** element, right click and select **Add Attribute/EndSumIndex**.
2. From the **EndSumIndex** attribute, double-click on the **NO VALUE** text field. Enter a value; for example: 10.

**Note:**

The **EndSumIndex** value should be greater than **BeginSumIndex** value and less than or equal to the **MaxIndex** value.

## Summing tables

This section explains how to use the **BeginSumIndex** attributes to correctly sum additional items within a table.

- Typically, the summed column is formatted as a dollar amount; if necessary, verify that the column has a dollar output format.
- If the output format for the DTA that prints the amounts is set to the standard \$1,234.50, the summed row's amount column will appear with a \$ and two decimals. If the output format for that same DTA is set to plain numbers (1234), the summed row's amount will also print in plain numbers.
- Note that **BeginSumIndex**, **MinIndex**, and **MaxIndex** are based off the iterator index, not the overall index. For example if the **IndexedPara/AddToIndex** attribute is used to start at two (2) instead of one (1) the **BeginSumIndex**, **MinIndex** and **MaxIndex** do not take the **AddToIndex** value into consideration and must be one less than normal.

1. Create a table. On the **tbody** element, right-click and select **Add Child Element/SortBy**.
2. On the **SortBy** element, right-click and select **Add Attribute/BeginSumIndex**. This value should be one (1) greater than the **MaxIndex** value.

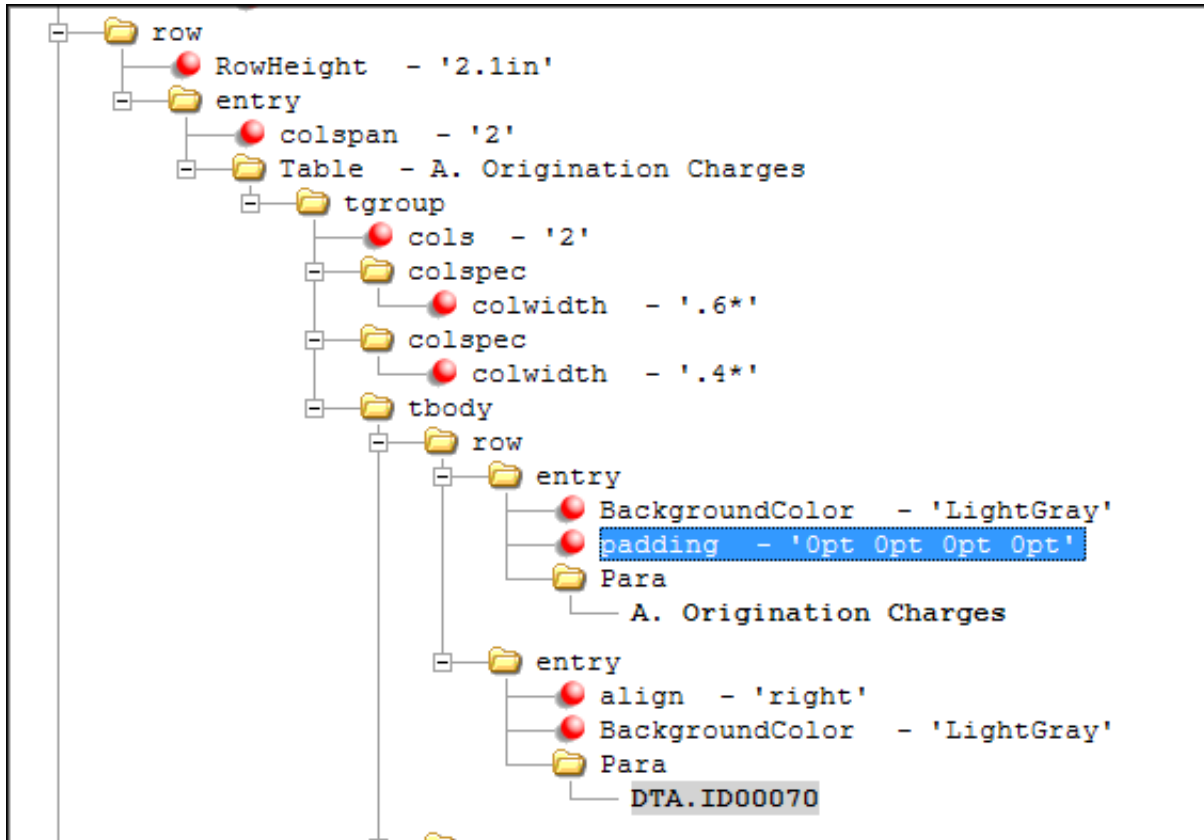
## Removing Padding between Table Headings

This topic explains how to remove padding (the gap) between a table heading and sub-table heading.

1. Within the child **table** element, right-click on **Row/Entry** and select **Add Attribute/padding**.
2. Within the **padding** attribute, enter the following values:
  - **Opt Opt Opt Opt**

These values represent the top, right, bottom, and left margins. Designating each margin as **Opt** removes the gap between the table heading and sub-table heading.

An example table might look like:



## Signature Areas

In many documents a signature is required at various places within the document. Most often a signature is required at the end of a document, sometimes signatures are required within the document on various pages. The Requirements Editor provides a defined set of elements to support the inclusion of signature lines (or signature areas) within a document.

There are two distinct types of signatures that may be added to a document depending on the use case.

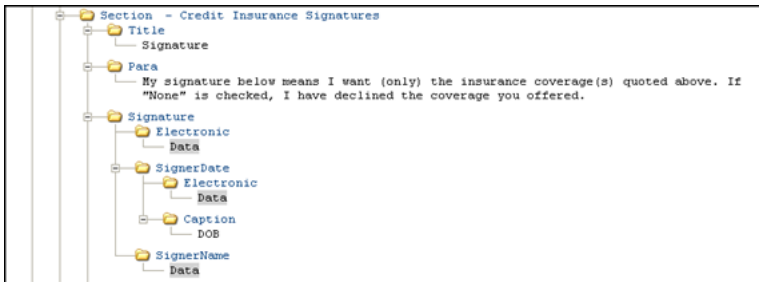
- To define signatures within the *body of the document*, use the **Signature** element. This type is often used for credit insurance signatures and notice signatures.
- To define signatures for the *entire document or agreement*, use the **SignatureSection** element. This type is typically placed at the end of the document.

In a signature area, whether within the body of the document or in a signature section, there are several options for defining the look and feel of the signature area including:

- the number of signature lines
- the text to include in the signature lines
- filling a signature area (highlighting)
- adding a box around the signature area

- For signatures within the body of the document the **Signature** element (one **Signature** element for each signature line needed) is used along with the necessary children. The following output would be produced based on the REQ structure shown below.

[Space for Signature]	[Date of Birth]
<b>[Name of Signer]</b>	<b>DOB</b>



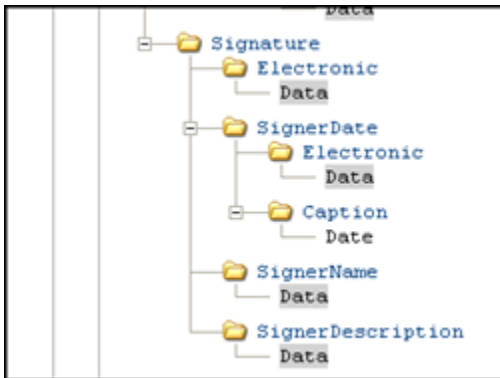
- For agreement signatures: use the **SignatureSection** element; depending on the type of document the children of **Signature** will vary. The standard for static documents is to have two entity signature blocks and two individual signature blocks. Standard structure for an entity signature block (each entity block should be in its own **SignatureGroup**):



The output would look resemble the following:

<b>Borrower</b>	
ABC Industries, Inc. <i>a Minnesota corporation</i>	
<i>John Doe</i>	<i>01/01/2009</i>
<b>By: John Doe</b> <i>President</i>	<b>Date</b>

- The **By** element is not used as a general rule, but is available for commercial documents if required. Standard structure for an individual signature block (not required to be in separate SignatureGroups) resembles the following:
- The **Lender** signature is another **SignatureGroup** within the **SignatureSection**. The standard structure resembles the following:



- The structure for the entire **SignatureSection** at a high level would resemble the following:



- Witness and notaries will use the same signature structure.

Signature areas are defined through a series of elements; the presentation of the signature areas on the generated document are defined through a stylesheet (*signatures.xsl*). **Signatures** and **SignatureSections** contain the following elements:

## Element Reference Information

Element	Description
SignatureSection/Title	The title of the SignatureSection. Formatted the same way a standard section title would be.
Para	standard para.
Notice	standard notice
SignatureGroup	Used to group multiple signatures together or group <b>Entity Name</b> and <b>Description</b> with a signature. There can be multiple levels of <b>SignatureGroups</b> .
SignatureGroup/Title	The name given to the group of signatures (ex. Borrower, Seller, Lender).
EntityName	The entity name.
EntityDescription	The entity description (ex. a Minnesota corporation).
HighlightSignaturesInd	This element is used as a flag (Boolean type) where the default value is False (0). A false value disables the highlighted signatures feature. A true value (True = 1) enables the feature. If submitted as true, the target signature block in the transaction data is highlighted in the defined manner in the output if both the EnableHighlight element is present in the document and the SignatureHighlightColor attribute defined
EnableHighlight:	This element is used to enable highlighting within an REQ when present at the document level. The element enables the SignatureHighlightColor attribute. If this element is not present in a document, highlighting is disabled, even if the SignatureHighlightColor attribute is defined.
Signature/Electronic	Used when the document will support electronic signatures.
SignerDate	Included if the signature line should be dated (signature date or date of birth). Notary and Witness signatures do not require dates.
Signature/SignerDate/Electronic	Used when the document will support electronic signatures. This area is for the signer date.
Signature/SignerDate/Caption	Used to indicate what caption should be used below the <b>SignerDate</b> area. Typically Date or DOB (Date of Birth).

Element	Description
SignerName	Used to indicate the signer name should print below the signature line.
SignerDescription	Can include the signer title or information for Power of Attorney. For some consumer products the signer description may not be included to save space.
Seal	Including the Seal element will trigger the stylesheet to include ( <i>seal</i> ) below the signature line.
By	Including the By element will trigger the stylesheet to include “By:” before the signer name below the signature line.

## SignatureSection Element

Base content normally uses a ContentReference to pull common signature sections into the document. Implementing special formatting in the SignatureSection will require changes and/or updates to the SignatureSection. Consideration should be given to creating new common signature documents that include the desired formatting to be referenced into the documents, or implementing the SignatureSection directly in the REQ itself. For purposes of the visual examples in this documentation, we will implement the updates directly in the REQ itself.

**Note:**

Set the structure to the following when a separate line is required for the signer to print their name:

```
SignatureGroup
Signature
Signature
SignerTitle
[Printed Name]
```

## Editing a ContentReferenced Signature Section

To perform this task you must have an REQ open in the Outline Editor and have the corresponding common signature document also open in Requirements Editor.

The steps below were written to replace a SignatureSection from a ContentReference to being directly in the document.

1. In the main document, locate the SignatureSection that needs special formatting.
2. Open the corresponding common REQ and navigate to the appropriate section called in the ContentReference.



- Copy the section from the common REQ and paste it into the main document. Now the SignatureSection can be edited for features. The original ContentReference can be moved into InactiveContent or deleted. The image below shows the original ContentReference in an InactiveContent element (container).



The SignatureSection(s) can now be edited and updated in the REQ.

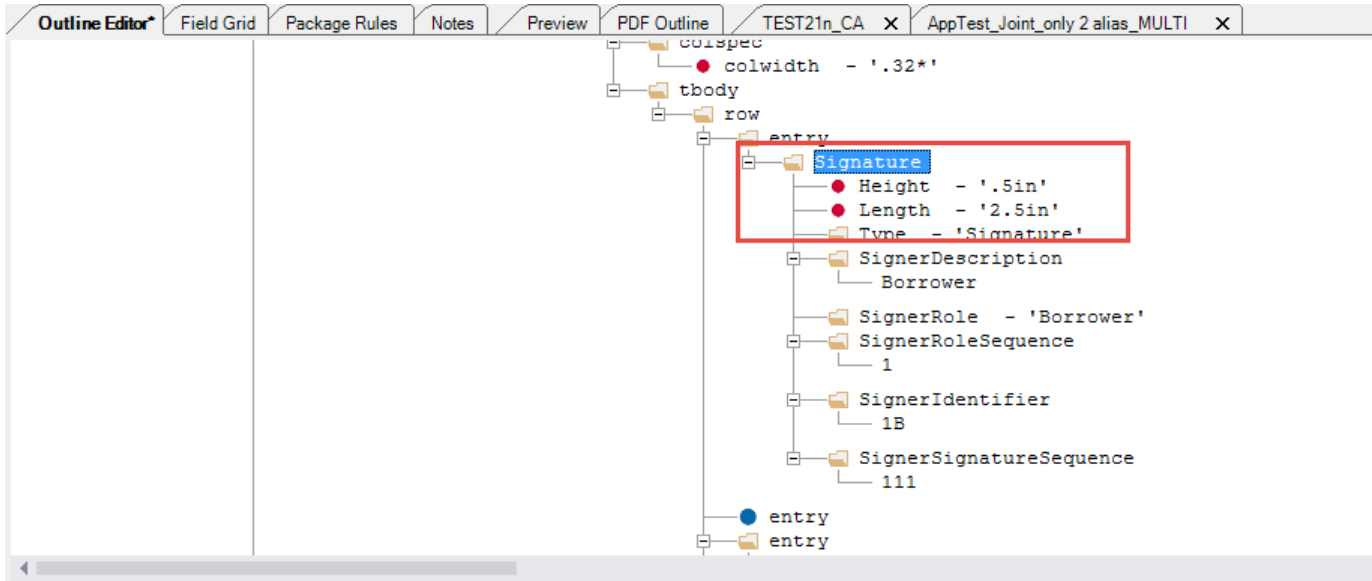
## Editing Signature Height and Width

You can modify the height and length of any **Signature** field within a document. Default height is *.5in* and default length is *2.5in*.

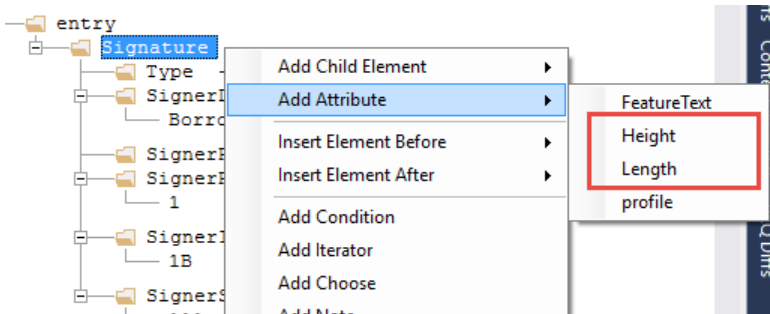
**Note:**

For this example, we will increase the length of the Signature field.

- Open the REQ file within which you wish to modify the Signature dimensions.
- Locate the **Signature** field that you wish to edit.

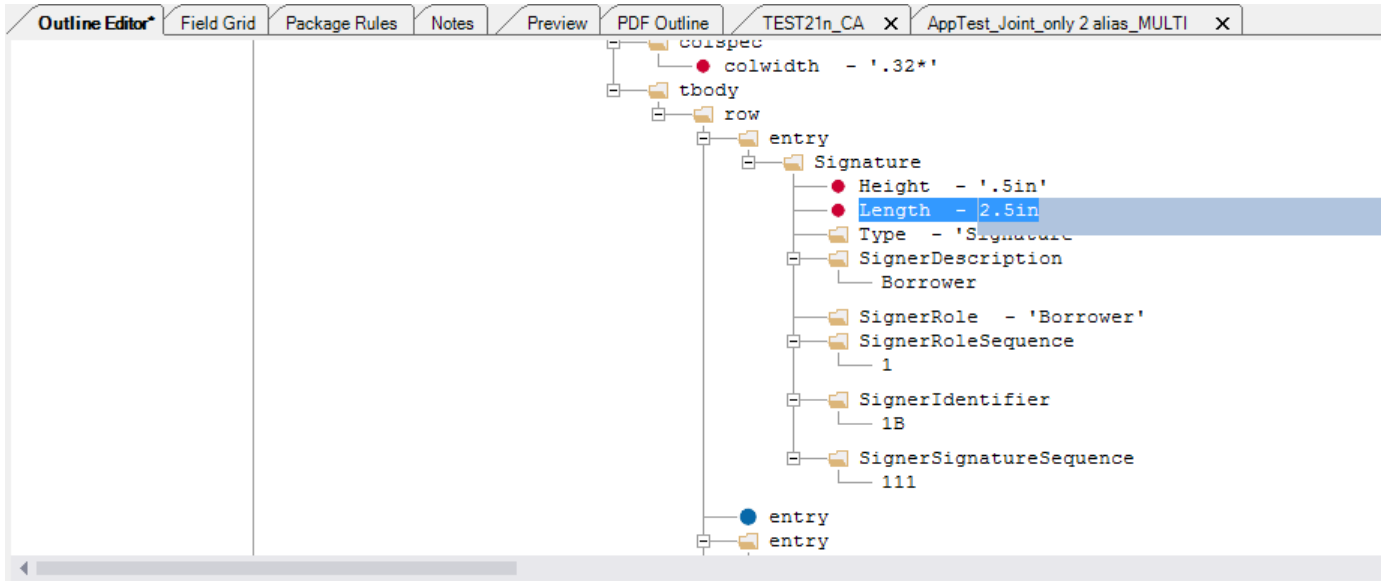


3. The attributes of Height and Length are not automatically added when a **Signature** element is added to a document. To add a Height or Length, you will need to add them as an attribute.

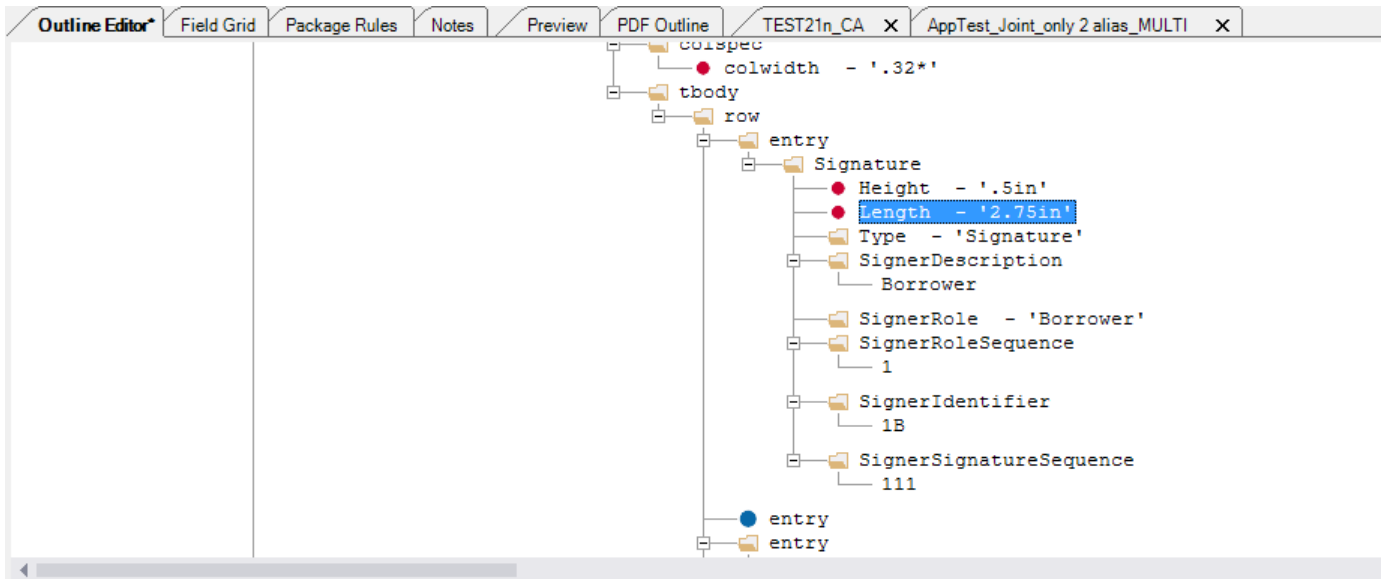


After you add the Height and Length, you will need to enter in a value. Entering the value in inches is best practice. If you do not need to override the default .5in height and/or the 2.5in length, then there is no need to add the additional attributes.

4. If you wish to change a default value, double-click the value you wish to modify.  
The field is highlighted and placed in edit mode.



5. Enter the desired value and press **Enter**. Best practice is to use values in inches (ex: 2.75in)  
The new value is accepted.



6. Save the REQ file to save your changes.

## Add the HighlightSignatureIndicator to the Transaction

Adding the HighlightSignature indicator to a transaction is slightly different depending on the schema associated with the document.



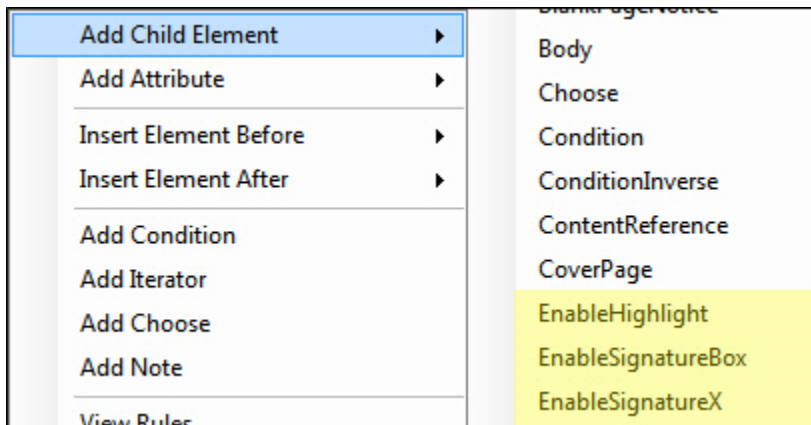
transaction for the *Highlight Signature Indicator*, add the desired feature (for example, the highlight color) to the signature area, and updating the flag in the transaction being processed.

- Add a Condition to Check for Highlighting in the Transaction
- Adding fill (highlighting)
- Adding a border box (boxing)
- Adding an X marker
- Add the HighlightSignatureIndicator to the Transaction

To access the available features, in the condition enabling the features right-click and add the the child element for the feature(s) that you want to apply in the document. The Condition will test if the transaction has the HighlightSignatureIndicator set to True/Yes and will include (turn on) each of the features within that condition.

**Note:**  
The features can be used individually, or in any combination.

The following shows the SignatureSection features.



### Add a Condition to Check for Highlighting in the Transaction

The best practice to implement a features in a signature area is to use a condition and coresponding flag in the transaction being processes. You can add the feature without the condition and this will permanently enable (or turn on) the signature feature(s) selected. The condition could be written to test any transaction field.

Before you define the fill feature with a color attribute, you must add a condition to the document and include a new element, EnableHighlight, in the condition. The EnableHighlight element, when present at the document level in an REQ, activates the SignatureHighlightColor attribute. If this element is not present in a document, highlighting is disabled (even if the SignatureHighlightColor attribute is defined).

**Note:**

Adding the EnableHighlight element at the document level outside a condition results in highlighting being permanently enabled in the document, not in the transaction. This is not the suggested best practice.

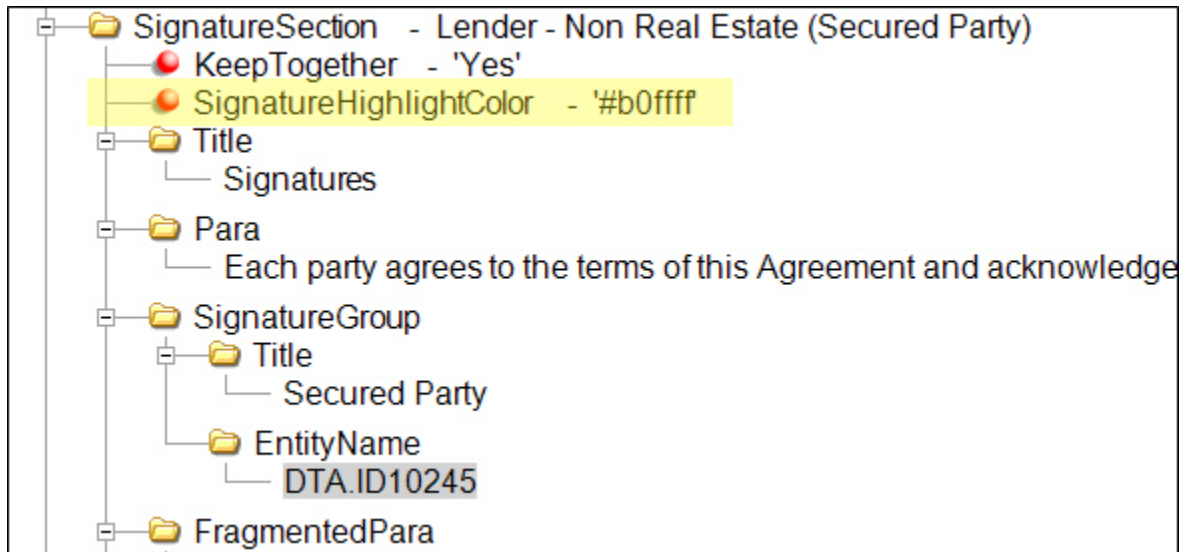
To add a condition for this purpose, using the Outline Editor perspective, select the target *Document* element in the requirements file and right-click to add a *Child Element*. In the right-click menu, click *Condition*. In the condition, add the PTR for the condition to check for the *HighlightSignatureIndicator* to be enabled (on/true/yes). The condition holding the EnableHighlight element evaluates the transaction to always include highlighting when the condition (true) is met. The PTR look like:

IF /TxnHighlightSignaturesInd = '1' THEN INCLUDE ENDIF. When the Condition test is true, the INCLUDE command enables, or turns on, the elements in the condition.

Business Logic:	Test for Enabling Highlighting
PTR:	<pre> iterator Scope: /Txn/CollateralItems/Collateral IF /Txn/HighlightSignaturesInd = '1' THEN INCLUDE ENDIF </pre>

### Adding Fill (Highlighting)

You can fill a signature area with a shade or color of your choosing to enhance the visibility of signature lines in a documents. Filling a signature area, or highlighting, can be applied to both dynamic and static documents. For dynamic documents, this is done by enabling highlighting in a condition and then defining the color of the highlighted area. In a document, the EnableHighlight element is an indicator flag that must be present at the document level of an REQ. The best practice is to set this element within a condition so that it enables at the transaction level. Once set, highlighting is authored in the **SignatureSection** by adding an attribute, *SignatureHighlightColor*, to define the color of the highlighted area.



The `SignatureHighlightColor` attribute defines the color of the highlighted area and allows you to enter a color (for example, yellow), select a color through the standard color palette through the shortcut menu (right-click and select Choose Color), or enter the HEX values (for example, #FFFF00). If the `SignatureHighlightColor` attribute is not supplied, no highlighting is applied. If the `SignatureHighlightColor` attribute is supplied, but no color defined, the default color is white and the area appears without highlighting.

**Note:**

A paler shade of each color is also provided, and might be preferable for visual documents. Based on our testing with the colors below, Yellow is the only color that does not appear if the document is photocopied. Remember that colors are dependent on the printer being used and will only appear as color if the printer supports color printing.

## Adding a Border (Box)

Adding a Border (Box) Around the Signature Area

A new element, **EnableSignatureBox**, added to the Requirements Editor introduces the ability to add a box to each signature area in a document. The **EnableSignatureBox** element adds a 1 pt. black box around **all** signature areas in the document. It's not possible to add the box to some signature areas and not others. This element can be applied to dynamic documents only at this time.

The **EnableHighlight** element adds a highlighted box to selected signature areas in the document. The highlighting is added inside the box when used in conjunction with **EnableSignatureBox**.

## Adding an X Marker

## Inserting a Signature Image

## Working with eSignatures

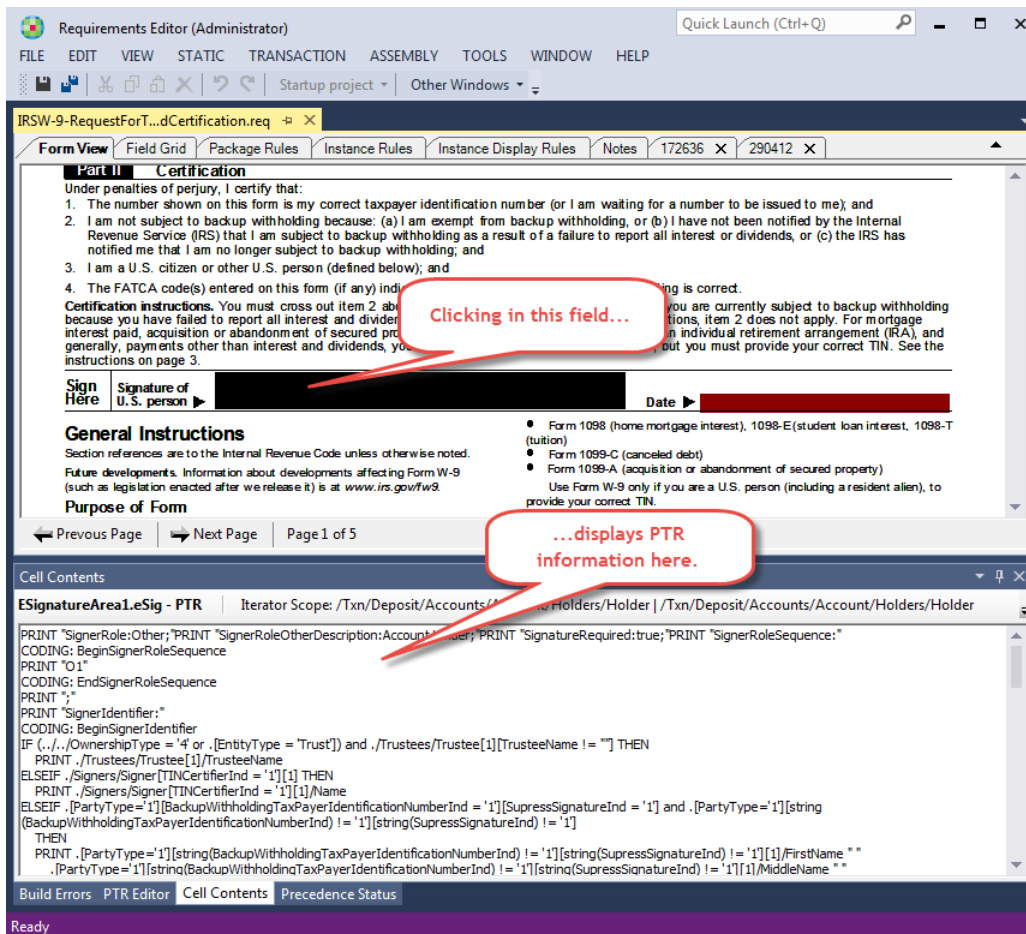
- Editing PTR Information Within Static Documents

### Editing PTR Information Within Static Documents

**Note:**

If you have purchased the Tagged PDF Outputs add-on license, refer to the Accessibility in Tagged PDF Outputs Implementation Guide for authoring the AlternateText attribute.

Clicking within the **eSignature** field of a static document displays PTR information within the **Cell Contents** View window, providing quick access to editing the information.



The screenshot shows the Requirements Editor interface. The main window displays a form titled "Part II Certification" with a signature field. A red callout bubble points to the signature field with the text "Clicking in this field...". Below the signature field, there are "General Instructions" and "Purpose of Form" sections. The bottom window, "Cell Contents", shows the PTR information for the signature field, including fields like "SignerRole", "SignerRoleOtherDescription", "SignerRoleSequence", "SignerIdentifier", and "SupressSignatureInd". A red callout bubble points to the PTR information with the text "...displays PTR information here.".

## Signature Guidelines for Static Documents

To add a signature area (box) to a static document you must first update the XTemplate using I32 Design 6, then add PTRs in Requirements Editor. Authoring static documents requires adding a new field to the XTemplate using I-32 Design and then updating the PTR for the new field when the XTemplate is brought into the Requirements Editor.

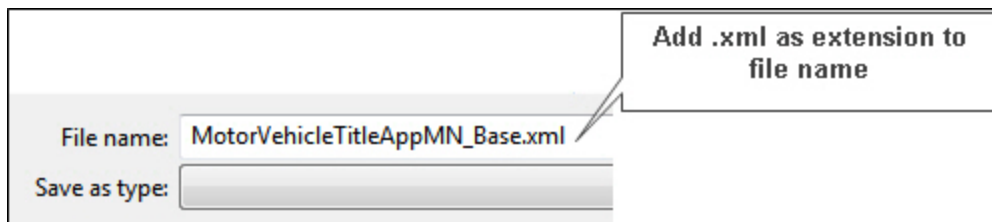
The PTR is updated in the Requirements Editor to include the attribute, HIGHLIGHT\_SIGNATURE, in the PRINT statement. When authoring XTemplate, select the Signature Tool (using the Signature Tool option in the toolbar or menu) and navigate to the target signature area. Then select the area above the signature line and drag to apply the SignatureTarget to the area. Select the object on the properties tab and define the color to apply to the signature area (properties > color) and save the XTemplate. The result is a SignatureTarget element in the XTemplate.

Highlighting is applied to the signature line defined through the SignatureTarget as the area above a signature and date line on a document. When the XTemplate is opened in the Requirements Editor, the PTR should be edited with a PRINT statement:

```
IF /LOAN/_APPLICATION/LOAN_PRODUCT_DATA/LOAN_FEATURES/@LienPriorityType = 'FirstLien'  
THEN PRINT "HIGHLIGHT_SIGNATURE" ENDIF
```

### Obtain an xTemplate Image from an existing REQ

1. Open the static document in Requirements Editor (first tab is titled Form View rather than Outline Editor).
2. From the menu, select Static > Export xTemplate. You will be prompted to save the xTemplate to a folder. Navigate to the desired folder. When naming the xTemplate, be sure to add the extension of xml to the file name.



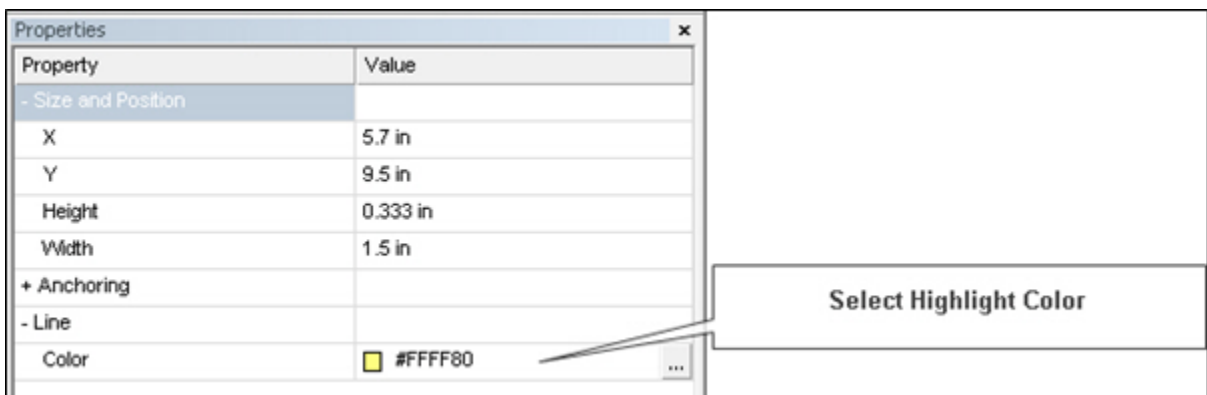
### Modify the xTemplate in I32 Design 6

1. Open the xTemplate in I32 Forms Design 6. You will need to navigate to the folder where the xTemplate was stored to and change the Files of Type to xTemplate Document (.xml) rather than the default of .sfx.
2. From the tool buttons, select the Signature Tool and draw a rectangle in the area of the signature to be Highlighted.



This will be a separate field “on top” of the existing area for signature. This additional field is needed for boxing around a signature, highlighting a signature, and for eSignatures.

3. Once the rectangle is drawn, in the Properties window for that box, select the line color and enter the hex for the desired highlight color.



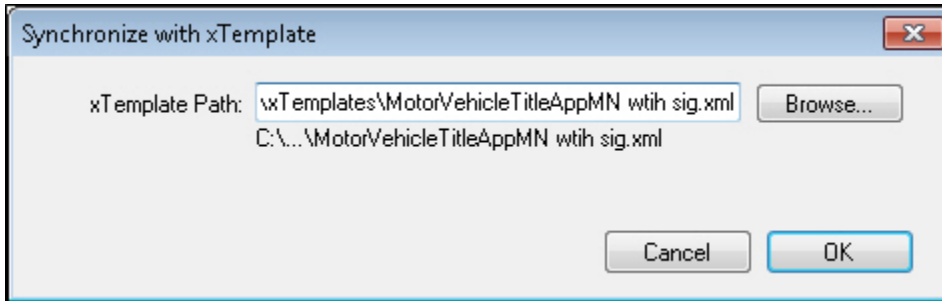
While the Properties > Object window refers to Line Color, the entire area within the rectangle will be filled with the indicated color.

4. When all signature areas have been defined, save the updated template as an xml.

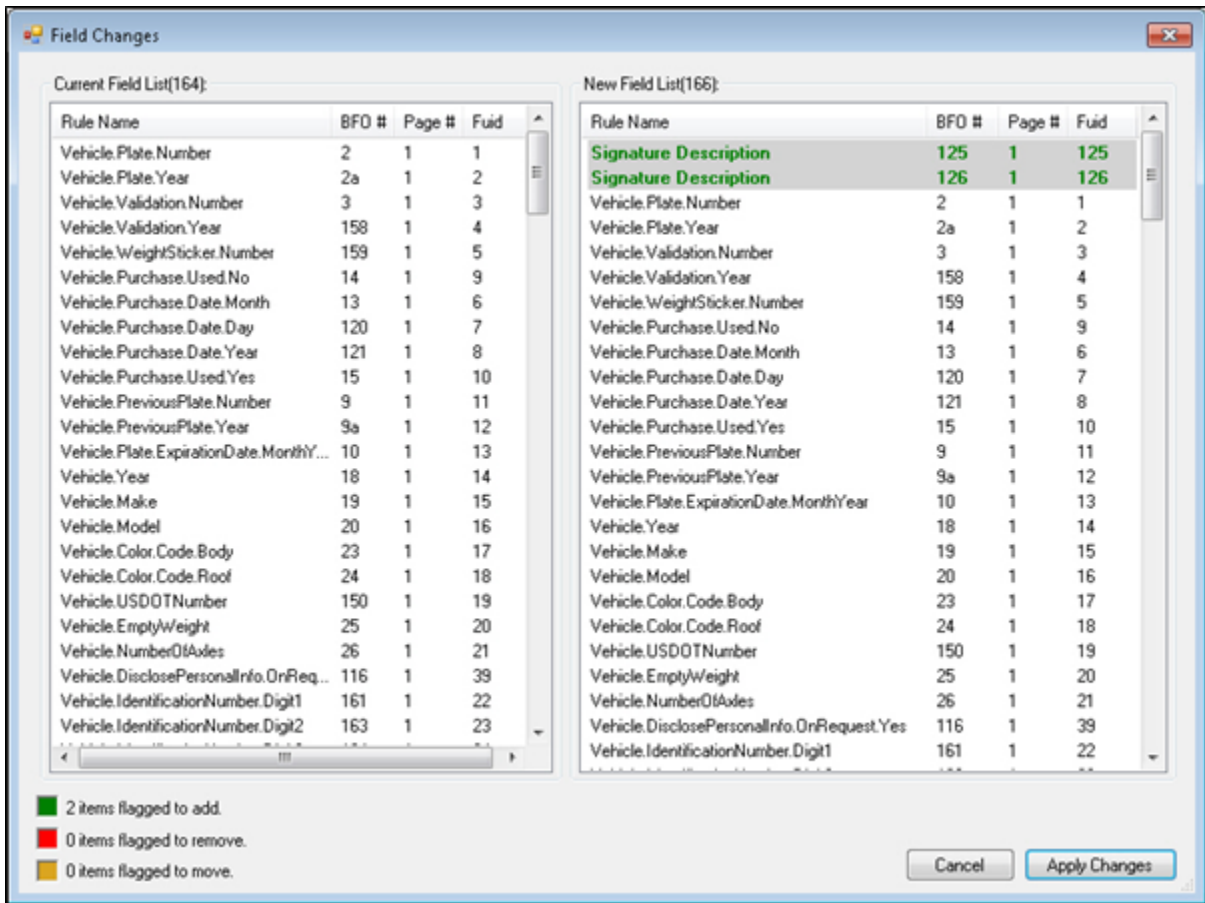
The xTemplate image has been updated with field(s) for highlighting.

### Import the updated xTemplate into Requirements Editor

1. Open the Requirements Editor tool.
2. Open the existing static document REQ file.
3. From the Requirements Editor menu, select Static > Import xTemplate.
4. Navigate to the path and xTemplate name of the updated xTemplate and click OK.



5. A new window will appear showing the field changes between the current xTemplate and the updated xTemplate. Review the changes (new signature and/or image fields) and click Apply Changes. This process retains all of the existing PTRs.



You are now ready to go to the next step, Modify the PTR for the Signature Areas, where you will edit the new fields in the PTR or Field Grid.

### Modify the PTR for the Signature Areas

1. In the Requirements Editor tool, using either the PTR window or the Field Grid, enter the appropriate PTR for each signature field. In the example, the first signature field is always highlighted and the second field is only highlighted if there is a second name (Borrower[2]/LastName is not empty).

Rule Name	BFO Field Name	Form Page	Business Logic	PTR	Field
Signature Description	125	1	Highlight Second Signature if needed	IF string(Borrower/Borrower[2]/LastName != "" THEN PRINT "HIGHLIGHT_SIGNATURE" ENDIF	125
Signature Description	125	1	Highlight First Signature	PRINT "HIGHLIGHT_SIGNATURE"	125

Watch the order in which the fields appear in the field grid. In the case of our example, the second signature field appears in the Field Grid BEFORE the first signature.

2. Save the updated REQ file.

## Working with Images

### Adding an Image File

#### InlineGraphic

Procedure 1. In an open .REQ within the Outline Editor, navigate to the desired element to place the InlineGraphic element (for example: Section) 2. Right-click and select Add Child Element/InlineGraphic. 3. Right-click on InlineGraphic and select Choose Picture. 4. From the Open dialog, navigate to the desired image. Select the image and click Open. 5. To modify the image, right-click on InlineGraphic and select Add Attribute; select any of the following: • Height • Width 72 6. Click on the No Value portion of the desired attribute; enter the value. Note: Enter the necessary unit of measurement; for example, inches, centimeters, etc.

#### inline graphic

A .tif (preprint output only) file can be inserted into a document. Examples include required state or IRS logos or customer logos. The .tif image is received from graphics and must be sized to the size the image will appear on the document. After inserting the InlineGraphic element, right click on the element name and click on the Choose Picture option from the menu. A browse window will open and allow you to navigate to the .tif image to be inserted into the file. #Standard for preprint logos is 1200 DPI but may be 300 DPI up to 2400 DPI depending on source of logo. #Standard for Eforms TIF file is 300 DPI. If the image used is not 300 DPI the image will not appear on the final output (board).

Selecting the image places it in the .REQ file. Do NOT edit any of the attributes available on the InlineGraphic element. This causes problems with the creation of the output.

### Adding an Image Reference to File System

#### External Graphic

## Adding an Image Reference to Transaction Data

TransactionGraphic

## Finding Rules Across a Collection

In the Field Grid, you may want to search the active collection (or repository) for a given rule or data point. For example, you may want to find all the documents within a package where a rule in a specific REQ is repeated. You can search documents for data points in the active REQ to determine if the data points exists in other documents and, more importantly, if the data points match across all documents.

This feature searches all documents in the repository that is pointed to as the authoring web service (Tools\Options). In the Expere options, point to a repository by assigning the URL for the // *SchemaRequestWebService/Service.asmx* of the target environment where the repository is stored.

To search the repository for all occurrences of a given rule, use the **Find Forms Using This Rule** option (Tools\Find Forms Using This Rule). In an REQ file, navigate to the Field Grid view. In the Field Grid, select the rule you want to search for and, in the Tools menu, click **Find Forms Using This Rule**.

Another example of how this feature might be used is to search for data points defining a car model/ make/year or boat hull width to identify all occurrences of a rule defining the car within all documents in an entire package.

The **Find Rule for a Public Name** option allows users to search documents for data that may pertain to the current .REQ. Users can compare data to other .REQ's to determine if data fields match. For example, this data could consist of a car model/make/year or boat hull width.

This option allows users to establish a baseline by consulting other documents to create fields for the current static .REQ.

## Exporting Field Grid Information

The **Export Field Grid** option allows users to export the field grid data as an XML file.

To export Field Grid data, perform the following:

1. Open an REQ file in the Editor and select the **PDF Outline** tab. Wait for the PDF to generate.
2. Select **Tools > Export Field Grid Data**.  
The Save As window is displayed.
3. Enter a file name, such as the document name, and select the **Save** button.
4. Open the file in an XML editor or use the instructions below to open the file in Excel.

## Output Formats

Output formats are used to define how certain entries are presented in the system. The goals behind output formats is to provide consistency in how certian values are presented and to ensure compliance with regulations regarding presentation.

In the Requirements Editor, output formats are defined in the *PRT Editor* in hte **Output Format** drop-down list. You can select a format for the active field or element in the *Outline Editor* perspective using the *PTR Editor*.

## Output Format Reference

The following output formats are available in the Requirements Editor.

### General Types - Checkbox, State, Text

Output Rule	Input Format	Example	Notes
Checkbox	Boolean (TRUE/FALSE)	[X]	Check the box if the condition is true, leave empty if false.
Texas	AK	Alaska	
TX	AK	AK	
String	a fee of \$15.00	a fee of \$15.00	Print exactly what is passed

The RAW format output is used in cases where a more appropriate format is not available. The RAW format outputs the input value as it is entered with unique processing instructions.

### RAW Output Formats

Output Rule	Input Format	Output Format	Notes
Raw	12345	12345	If this output is selected and no data is passed, nothing will print. (not even a line)
Raw 7	0.123456789	0.1234567	Print number as given truncating after the seventh decimal position. No rounding; no dropping leading or trailing zeros; no filling with leading or trailing zeros.

Output Rule	Input Format	Output Format	Notes
Raw 7%	0.123456789	0.1234567%	Print number as given, truncating after the seventh decimal position with a % sign following it. No rounding; no dropping leading or trailing zeros; no filling with leading or trailing zeros.
RAW_IntDisc	-1234.56	-\$1,234.56	Used to apply a minus sign [ - ] to negative numbers as opposed to containing it in parenthesis. This applies to Integrated Disclosure documents using the Integrated Disclosure style sheets. Other documents will render a negative amount within parenthesis [ ( ) ]. Also, if used but no data is included no line will print on the output.
	12.345-0%	12.345-0%	Print number as given, truncating after the third decimal position. Trailing zeros are dropped. This applies to general numbers and percents. Percents with four (4) or more places after the decimal are rounded at the third decimal place.

### Date

Output Rule	Input Format	Output Format	Notes
d mmmm yyyy	20140105	5 January 2014	
m/d/yy	20140105	1/5/14	
m/yy	20140105	1/14	

Output Rule	Input Format	Output Format	Notes
mm/dd/yy	20140105	01/05/14	
mm/dd/yyyy	20140105	01/05/2014	
mm/yy	20140105	01/14	
mmm d	20140105	January 5	
mmm d, yyyy	20140105	January 5, 2014	
mmm, yyyy	20140105	January, 2014	
d	20140105	5	no leading zeros
dd Date	20140105	05	include leading zero for 1 thru 9
dth	20140105	5th	ordinal
m	20140105	1	no leading zeros
mm	20140105	01	include leading zero for 1 thru 9
mmm	20140105	Jan	3-letter abbreviation of month
mmm	20140105	January	Full text of month
yyyy	20140105	2014	4-digit year
yy	20140105	14	2 digit year

### Currency (Dollar)

Output Rule	Input Format	Output Format	Notes
\$1,234	0	\$0	Rounds to whole dollar
	1234.4678	\$1,234	
	1234.5678	\$1,235	
\$1,234.50	0	\$0.00	Print at least 2 decimals (trailing zeros, if any) up to 3 decimal places truncates at last decimal
	0.00	\$0.00	
	1234.5	\$1,234.50	
	1234.5678	\$1,234.567	
	12345.51863	\$12,345.518	

## Number

Output Rule	Input Format	Output Format	Notes
.05	1234.03	03	Print decimal part only without decimal point; two places, with leading, trailing zeros if needed; rounds after 2 decimals
	1234.5678	57	
	1234.5	50	
01	1.2345		Leading zero for 1-9 Rounds to whole dollar
	1231.5	01	
		1232	
1234	0	0	Rounds to whole dollar
	0.00	0	
	1234.35	1234	
	1234.56789	1235	
1,234	0	0	Rounds to whole dollar
	0.00	0	
	1234.25	1,234	
	1234.56789	1235	
1,234.	1234.56789	1,234.56789	Print up to six decimal places. Delete the trailing zero(s). Rounds sixth decimal place
	1234.5678916	1,234.567892	
1,234.56	1234.56789	1,234.57	Round at 2 decimals; print trailing zero(s) if needed
	1234.5	1,234.50	
1,234.567	1234.56789	1,234.568	Round at 3 decimals; print trailing zero(s) if needed
	1234.5	1,234.500	

Output Rule	Input Format	Output Format	Notes
0.12	.5678	0.57	Round at 2 decimals; print trailing zero(s) if needed. Print 0 in 1's place if no 1's digit passed.
	123.45678	123.46	
	12.3	12.30	
#th	5	5th	
	123	123rd	
##th	5	fifth	
	123	one hundred twenty-third	
Alpha	5	five	Sentence case rounding at two decimal places; watch capital letters in sentences
	123	one hundred twenty-three	
	6.578	six and fifty eight	
Alpha Check Amount	1,234.56	One thousand two hundred thirty four and 56/100	String value rounded after 2 decimals; print trailing zero(s) if needed
Alpha Percent	0.0625	zero and one-sixteenth four and three-eighths	Lower case percent
	4.375		

## Percent

Output Rule	Input Format	Output Format	Notes
12%	12.345678	12%	Round decimals
	4.567	5%	
12.%	12.345678	12.345678%	Print as up to six decimal places. Delete the trailing zero(s). If the seventh decimal is 5 or more then it rounds up, otherwise it rounds down.
	12.300000	12.3%	

Output Rule	Input Format	Output Format	Notes
12.34%	12.345678	12.35%	Round at 2 decimals; print trailing zero(s) if needed
	12.3	12.30%	
12.345%	12.345678	12.346%	Round at 3 decimals; print trailing zero(s) if needed
	12.3	12.300%	

### Address

Output Rule	Input Format	Output Format	Notes
Block	(name and) address	Lynne Rossetto Kasper, Jr. Suite 456 123 Elm Street Minneapolis, MN 55410	Print (name and) address in block format. Name, if included, on first line. Comma before suffix, if any. Each street address on separate line. City, state, ZIP on last line, with comma after city. No other punctuation Might be Static Doc Only
In line	(name and) address	Lynne Rossetto Kasper, Jr., Suite 456, 123 Elm Street, Minneapolis, MN 55410	Print (name and) address in line. Name, if included, with comma before suffix, if any. Then a comma. Then each street address, each followed by a comma. Then city, state, ZIP, with comma after city

### Date/Time

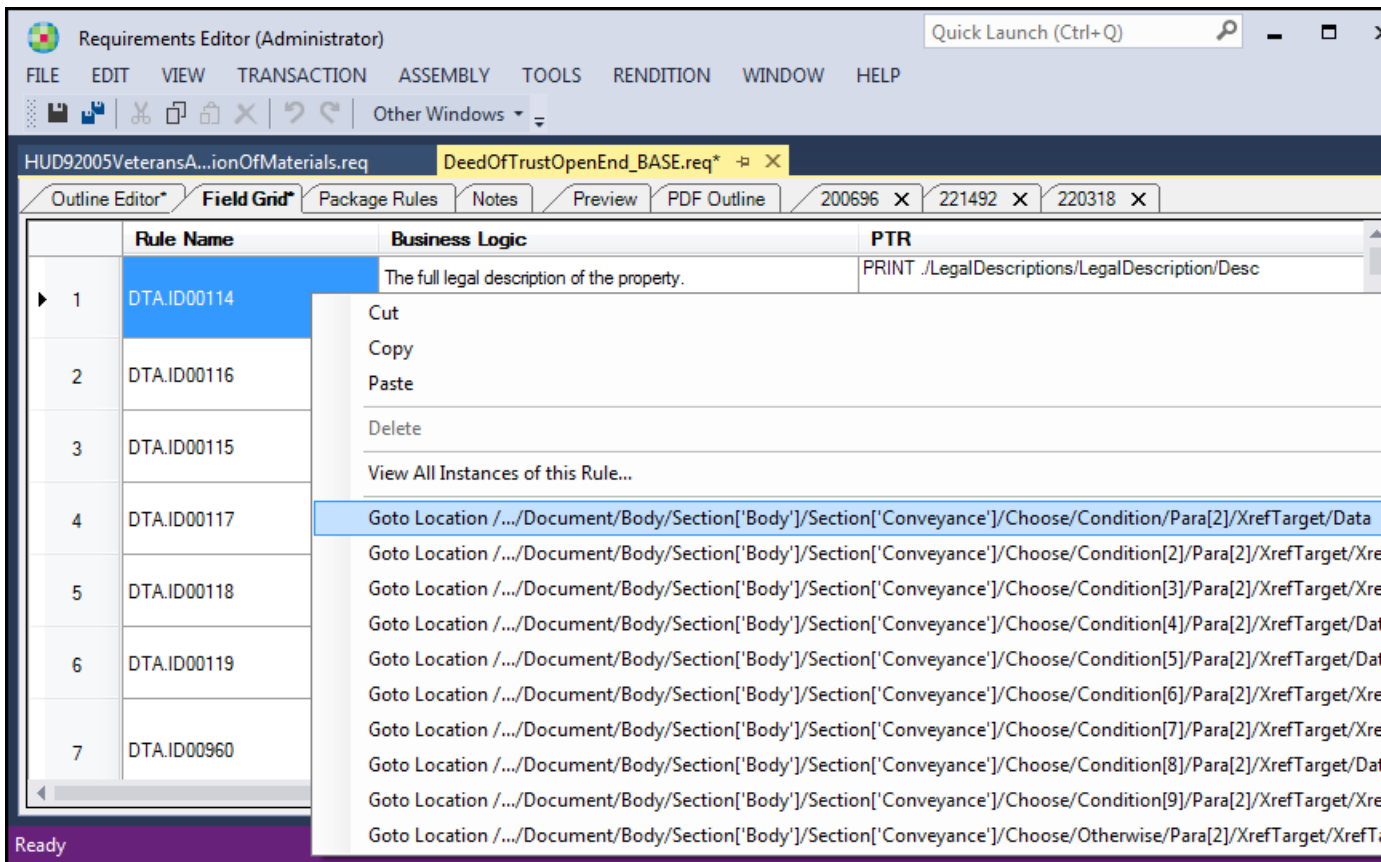
Output Rule	Input Format	Output Format	Notes
hh:mm:ss[24hr]	20070101T24:00:00	13:30:33	
hh:mm A.M. /P.M.	20070101T24:00:00	12:00 a.m.	
hh:mm:ss A.M. /P.M.	20070101T24:00:00	12:00:33 a.m.	

## Using the Field Grid to Navigate to a Data Point Location

A dynamic or static requirements file is opened in the Requirements Editor.

Authors can navigate to the location of a particular data point (DTA) as it appears in the XML by using the **Field Grid** perspective. . If the same DTA is used in multiple places a list will appear displaying all locations to select from.

1. With a requirements file open, click the **Field Grid** tab.
2. In the *Field Grid*, select the DTA you want to locate and right-click. You can select any field in the row for the target data point whether Rule Name, Business Logic, PTR, et al).
3. In the shortcut menu, a list of all locations in the XML where the data point exists is displayed.



4. In the list of *Goto Locations*, click the target location.

Once selected, the Outline Editor displays the data point as it occurs in the file at the selected Goto Location.

The Outline Editor displays the selected data point at the location it occurs in the requirements file.

## Defining Measurement Values

When authoring content you can apply measurement values to various attributes defining the dimensions (length, width) at the element level. The best practice is to define measurement values in lowercase; entering the values in UPPERCASE may cause unexpected results in the generated output.

The standard units of measurement supported by the Requirements Editor include all valid CSS measurement units: %, in, cm, mm, em, ex, pt, pc, and px. You can use both absolute units (cm, mm, in, pt, and pc), font-centric units (em, ex), the pixel (px) and percentage (%) units. If a measurement unit is entered in UPPERCASE (for example, 1PT), the Requirements Editor will accept the value but will not assemble the document in a manner that produces the desired output.

**Note:**

The Requirements Editor will not present a validation error or any indicator that a measurement value was entered as UPPERCASE and, as a result, the output is at risk of not producing the desired results.

## Using Text Formatting Tags within DataPoints

You can use text formatting tags in transaction data contained in a data point structure (DTA) assembled for output as a dynamic document. When authoring a dynamic document, data within the string comprising a DTA can be authored using standard HTML text formatting tags (bold, italics, underline) as well as line breaks. The supported text formatting tags (bold = <b>, italics = <i>, underline = <u>, and a line break = <BR>) can be applied to the entire string or partial text within a string as part of the data point field.

**Note:**

The use of the emphasis (<emph>) formatting tag is not supported.

For example, a dynamic document authored with tags such as:

`<b>Fees</b>: <i>nominal fee for</i> <u>registration</u>.`; will appear in the output as:  
**Fees: *nominal fee for*registration.**

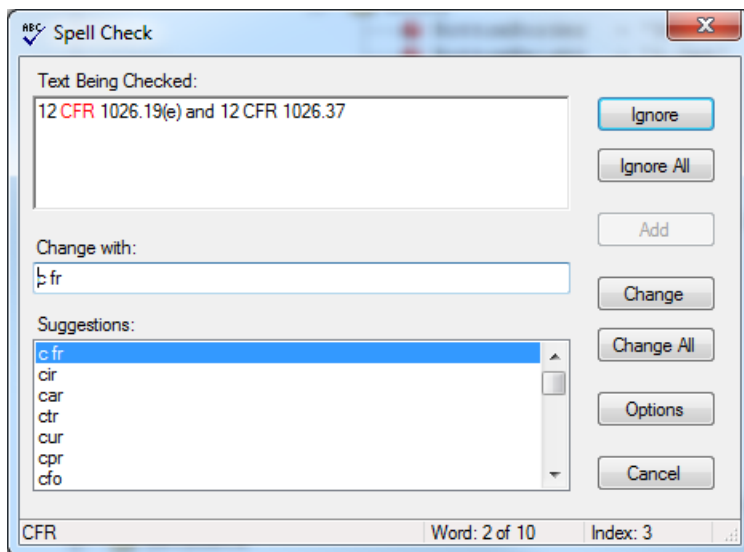
This same functionality works differently for static documents. In a static document text formatting tags are removed from transaction data prior to assembly for output as a static document. So that when a static document is assembled for generation and includes the HTML text formatting tags (bold, italics, underline), the tags are stripped out of the transaction data to prevent the tags from rendering in the output. The standard text formatting tags (bold = <b>, italics = <i>, and underline = <u>) are stripped out of the transaction data prior to processing. This applies to any of these four tags contained within an entire string or partial text within a string as part of the data point field.

For example, a static document authored with tags such as:

`<b>Fees</b>: <i>nominal fee for</i> <u>registration</u>.`; will appear in the output as:  
Fees: nominal fee for registration.

## Using Spell Check within Dynamic Documents

The Spell Check feature of Requirements Editor allows you to check the spelling of all text in any open dynamic document. Selecting **Tools > Spell Check** displays the familiar Spell Check dialog.



Note the following:

- Regardless of your location within the document, Spell Check begins at the top of the information within the **Outline Editor** tab.
- Only the information within the **Outline Editor** tab is checked. All other tabs are ignored.
- Code for graphics are not checked.

## Author Fillable Field or Fillable Checkbox

Fillable Fields or Fillable Checkboxes may be authored to reserve space on an assembled PDF to capture data input. These element details are output to the assembly-log.txt file saved to the application data folder each time a transaction is assembled (`c:\Users\user.name\AppData\Local\Temp\`) as well as included in the Expere API services.

When a Fillable Checkbox element is encountered in content, logic is used during processing to reserve the appropriate space for formatting and a post-processing operation adds an AcroForm checkbox object at the position indicated. The checkbox is an Adobe object sized as .1in by .1in plus a following space. The `FieldName` element value authored is used as the AcroForm object name.

A Fillable Field is placed so that text entered in the field reads consistently with preceding and following text, such that the baseline of text entered in the field must be the same as the baseline of other text on the same line. This placement is independent of the height property associated with the field. AcroForm field values are vertically aligned in the middle of the field rectangle and this alignment cannot be modified programmatically; therefore, the placement of the field rectangle is computed from the desired baseline, taking into account the height of the field, the height of the font for field data.

Fillable Checkbox and Fillable Fields are tagged for accessibility when producing a TaggedPDF. See [Accessibility in Tagged PDF Implementation Guide](#) for further details.

The `FillableField` and `FillableCheckbox` elements may be authored as a child of the following elements; however, the most common use is `FragmentedPara`.

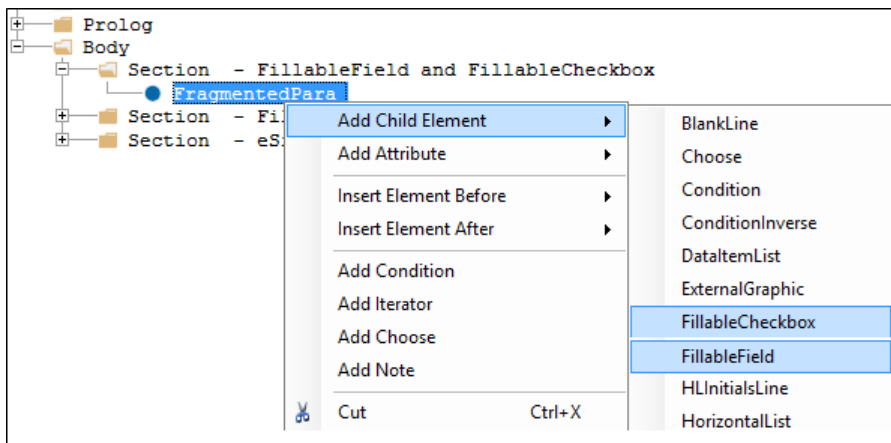
- Tagline

- HorizontalListItem
- FragmentedPhrase
- FragmentedPara
- LinkText

**Note:**

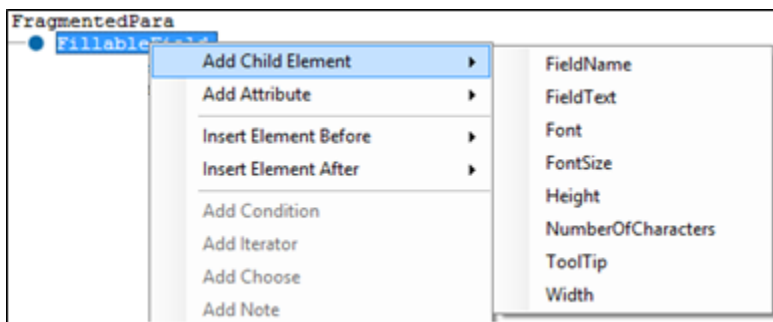
FillableField/FillableCheckbox may be authored in a HeaderNotice. However, the functionality to create the Acroform text field/object accurately on the PDF is not supported as this time.

Right click the **FragmentedPara** element and select **Add Child Element > FillableField** or **FillableCheckbox** from the shortcut menu.

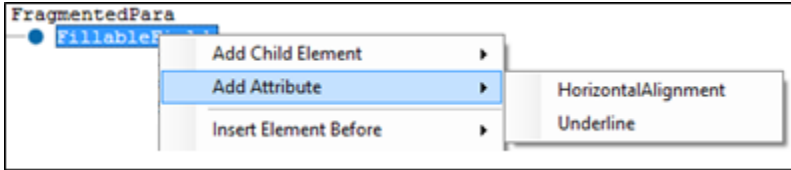


To add elements or attributes, right-click the **FillableField** or **FillableCheckbox** element and select **Add Child Element** or **Add Attribute** from the shortcut menu.

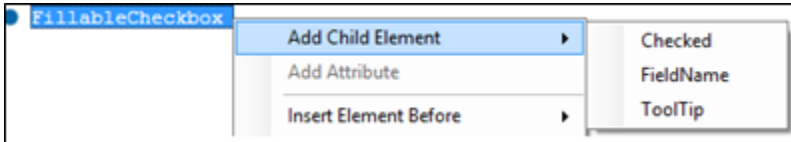
FillableField Child Elements:



FillableField Attributes:



FillableCheckbox Child Elements:



The table below outlines the attributes and elements available and authoring rules for the **FillableField** and **FillableCheckbox** elements.

### Element/Attribute Reference Information

Type	Element/Attribute	Required	Description
FillableField	FieldName	Required	<p>The name of the fillable field. The contents may be a string, a Data element (DTA) with a PTR authored, or a mix of text and Data elements. The FieldName must conform to the rules for an AcroForm field.</p> <p><i>Note that Adobe Acrobat has special rules for handling fields when the field name contains a period or space. Do not use spaces or periods when authoring FieldName. Special characters may be used; however, use of the ampersand (&amp;) will result in assembly error.</i></p>
	ToolTip	Optional	<p>The text that displays when the user hovers over the fillable field on the PDF. The contents may be a string, a Data element (DTA) with a PTR authored, or a mix of text and Data elements. Special characters may be used; however, use of the ampersand (&amp;) will result in assembly error. The ToolTip property becomes the hover text associated with the Acroform text field on the PDF.</p> <p>If the ToolTip element is not authored, the FieldName value will be used as a default for the ToolTip as accessibility requirements for Tagged PDFs requires a ToolTip for AcroForm fields.</p>

Type	Element/ Attribute	Required	Description
	Width	Required	<p>Identifies the Width of field. May be authored in points (144pt) or inches (2in). If authored as inches, the value is converted and output as points in the response file. Entering text in the fillable field is limited to the lesser of the Width or Number of Characters element value.</p> <p>Note: If FieldText is authored to prepopulate data for the fillable field and the value exceeds the characters that can visually fit in the Width defined, all text continues to populate the field but only those characters that fit in the Width are visible; the non-visible characters may be removed from clicking the field text and removing visible characters. To truncate characters, author the NumberOfCharacters elements to control the maximum characters allowed. Entering data in Fillable Field AcroForm text fields on the PDF without the NumberOfCharacters element authored is limited to characters that fit in the Width dimension authored. Refer to NumberOfCharacters element for additional details on data truncation.</p>
	Height	Required	<p>The name of the font. A large number of fonts may be authored and honored on the PDF; however, the following fonts are those that apply true height adjustments. For more information, refer to the following table, <i>FontName Element Values</i>.</p>
	FontName	Required	<p>The name of the font. See <a href="#">FontName Element Values</a> for the FontName acceptable values to author.</p> <p>Note: Font Scaling modifications occur on Fillable Field FieldText similar to existing Expere Content to adjust to true font height for applicable fonts (CourierNew and TimesNewRoman). See the Expere Developer's Guide 'Font Scaling Adjustments Stylesheet Details' section to review font scaling modification details.</p>

Type	Element/ Attribute	Required	Description
	FontSize	Required	<p>Numeric value. Represents the size of the font authored in points (i.e. enter 10 to represent a 10pt font).</p> <p>Note: Font Scaling modifications occur on Fillable Field FieldText similar to existing Expere Content to adjust to true font height. The FontSize value is modified from what is authored to apply font scaling adjustments for applicable fonts (CourierNew and TimesNewRoman). See the Expere Developer's Guide 'Font Scaling Adjustments Stylesheet Details' section to review font scaling modification details.</p>
	NumberOfCharacter	Optional	<p>The maximum number of characters allowed. Note: A value of 0 (zero) is ignored when the transactions is assembled; if there is no limitation on the number of characters, do not author this element.</p> <p>When the NumberOfCharacters element is authored, when entering data in the fillable field on the PDF, the user will be limited to the number of the characters value authored.</p> <p>Note: If FieldText was authored to default or populate data for the fillable field that exceeds the number of characters, the data that exceeds the maximums characters will be truncated. If the Width element is authored such that the dimension is less than the Number of Characters authored, the prepopulated text is not truncated but only those characters that fit in the Width are visible. When modifying prepopulated text in this case, user may delete characters to see additional text that is not visible but will be limited to entering only the number of characters that fit the Width.</p>
	FieldText	Optional	<p>The prepopulated text for the field. The contents may be a string, a Data element (DTA) with a PTR authored, or a mix of text and Data elements. Special characters may be used; however, use of the ampersand (&amp;) will result in assembly error.</p>
	HorizontalAlignmen (Attribute)	Optional	<p>Indicates the horizontal alignment so of the text in the fillable field. Values are: left, right, center. If the horizontal alignment attribute is not authored or value not specified, the default is left alignment.</p>

Type	Element/Attribute	Required	Description
	Underline (Attribute)	Optional	Indicates if the fillable field should be underlined. Values are: Yes, No. A value of Yes is automatically defaulted when the attribute is added. The Underline element draws a bottom border the width of the Fillable Field.
FillableCheckbo	FieldName	Required	<p>The name of the fillable checkbox. The contents may be a string, a Data element (DTA) with a PTR authored, or a mix of text and Data elements. The FieldName must conform to the rules for an AcroForm field.</p> <p>Note that Adobe Acrobat has special rules for handling fields when the field name contains a period or space. Do not use spaces or periods when authoring FieldName. Special characters may be used; however, use of the ampersand (&amp;) will result in assembly error.</p>
	ToolTip	Optional	<p>The text for the tooltip that is displayed when the user hovers over the field on the PDF. The contents may be a string, a Data element (DTA) with a PTR authored, or a mix of text and Data elements. Special characters may be used; however, use of the ampersand (&amp;) will result in assembly error. The ToolTip property becomes the hover text associated with the Acroform checkbox object on the PDF.</p> <p>If the ToolTip element is not authored, the FieldName value will be used as a default for the ToolTip as accessibility requirements for Tagged PDFs requires a ToolTip for AcroForm fields.</p>
	Checked	Optional	<p>Indicates if the fillable checkbox should resolve as selected or deselected. If the element is not authored, the checkbox will default as deselected. The reserved value 'CHECKBOX' can be used when authoring the PTR to set the checkbox to be selected (functions the same as Output Format 'Checkbox'). PTR Examples that resolves as a selected checkbox:</p> <ol style="list-style-type: none"> <li>PTR with if/then logic: <pre>IF [expression] THEN   CHECKBOX ENDIF</pre> </li> <li>PTR that resolves as checkbox being selected: <pre>CHECKBOX</pre> </li> </ol>

Type	Element/ Attribute	Required	Description
<i>*Note: If required elements are not authored, the document will fail to assemble.</i>			

### FontName Element Values

RE Tool - Font Value Authored	Assembly Log & Expere API Response File - FontName Output Value
arial (arial)	ArialMT
arialbd (arial bold)	Arial-BoldMT
arialbi (arial bold italic)	Arial-BoldItalicMT
ariali (arial Italic)	Arial-ItalicMT
cour (courier new)	CourierNewPSMT
courbd (courier new bold)	CourierNewPS-BoldMT
courbi (courier new bold italic)	CourierNewPS-BoldItalicMT
couri (courier new italic)	CourierNewPS-ItalicMT
times (times new roman)	TimesNewRomanPSMT
timesbd (times new roman bold)	TimesNewRomanPS-BoldMT
timesbi (times new roman bold italic)	TimesNewRomanPS-BoldItalicMT
timesi (times new roman italic)	TimesNewRomanPS-ItalicMT
sansmd (BSI Sans)	BSISans
sansbd (BSI Sans Bold)	BSISansBold
sansbdit (BSI Sans Bold Italic)	BSISansBoldItalic
sansmdit (BSI Sans Italic)	BSISansItalic
serfmd (BSI Serif)	BSISerif
serfbd (BSI Serif Bold)	BSISerifBold
serfbdit (BSI Serif Bold Italic)	BSISerifBoldItalic
serfmdit (BSI Serif Italic)	BSISerifItalic

### Assembled PDF Examples

The following screencaps show an assembled PDF with a Fillable Field and Fillable Checkbox.

In this example, the FieldText element for the FillableField was authored with string text to populate a default value. Likewise, the Checked element for the FillableCheckbox was authored with a DTA and PTR to default to selected.

## ***FillableField and FillableCheckbox***

This is a fillablefield:

This is a fillablecheckbox:

In this example, the FieldText/Checked elements are not authored resulting in no pre-populated/default data.

## ***FillableField and FillableCheckbox***

This is a fillablefield:

This is a fillablecheckbox:

Hovering the mouse over the FillableField or FillableCheckbox displays the ToolTip text.

## ***FillableField and FillableCheckbox***

This is a fillablefield:

This is a fillablecheckbox:

Enter text here.















Click the box to deselect/select.







Clicking or tabbing to the Fillable Field AcroForm text field or Fillable Checkbox object gives it focus and the value may be modified/entered.

## Appendix

### Icons

This section describes each icon displayed within Requirements Editor.

This icon	Indicates
<b>Merge Session Icons</b>	
	<b>Added:</b> a document that has been added to the incoming repository
	<b>Deleted:</b> a document that has been deleted in the incoming repository
	<b>Impacted:</b> a root org modification was made that impacted this inherited document
	<b>Modified:</b> the incoming document has been modified from the existing repository
	<b>Orphaned:</b> the incoming document is missing a parent document within the existing repository
	<b>Spared:</b> this child level document was not impacted by a change to a root org modification
	<b>Unmodified:</b> the incoming document has not changed compared to the existing document
	<b>Invalidated</b>
<b>Circles</b>	Circle icons represent the progress of the reconciliation of an REQ file.
	<b>No Action Needed:</b> A document has been modified, but no resolution is required.
	<b>Pending:</b> the incoming document requires reconciliation
	<b>In Progress:</b> indicates reconciliation has begun on the REQ file and there are other changes requiring attention.
	<b>Resolved:</b> the document has been successfully reconciled. This icon appears only after all ContentReferenceBlock and ChangeBlocks have been resolved.
<b>Flags</b>	Indicates a document that has been affected. Flags only appear if the Section it resides in has been changed. <div style="border: 1px solid green; padding: 5px; margin: 10px 0;"><b>Note:</b> No flag indicates the document was not affected by any content reference changes.</div>
	<b>Grey Flag:</b> The document was affected by a content reference change at the root level and no action is required.
	<b>Red Flag:</b> The document was affected by a content reference change and pending a decision.

This icon	Indicates
	<b>Green Flag:</b> The document was affected by a content reference change and has been resolved.
	<b>Note:</b> Indicates the associated node is a Note
	<p><b>ChangeBlock:</b> Indicates the associated node is a ChangeBlock.</p> <p>When a ChangeBlock has been resolved, the icon is changed to a lighter color. In the following example, the top ChangeBlock is Pending and the bottom ChangeBlock has been Resolved.</p> <pre> ChangeBlock - 2 (Both) ChangeBlock - 3 (Existing) </pre>
	<b>Not Recordable</b>
	<b>ContentReferenceBlock</b>
	<b>Row</b>

**About Wolters Kluwer Financial Services** - Whether complying with regulatory requirements or managing financial transactions, addressing a single key risk, or working toward a holistic enterprise risk management strategy, Wolters Kluwer Financial Services works with more than 15,000 customers worldwide to help them successfully navigate regulatory complexity, optimize risk and financial performance, and manage data to support critical decisions. Wolters Kluwer Financial Services provides risk management, compliance, finance and audit solutions that help financial organizations improve efficiency and effectiveness across their enterprise. With more than 30 offices in 20 countries, the company's prominent brands include: FRSGlobal, FinArch, ARC Logics®, TeamMate®, Bankers Systems, VMP® Mortgage Solutions, AppOne®, GainsKeeper®, Capital Changes, NILS®, AuthenticWeb™ and Uniform Forms™. Wolters Kluwer Financial Services is part of Wolters Kluwer, a leading global information services and solutions provider with annual revenues of (2012) €3.6 billion (\$4.6 billion) and approximately 19,000 employees worldwide. Please visit our website for more information.

**Wolters Kluwer Financial Services**  
6815 Saukview Drive  
St Cloud, MN, 56303  
Toll-free: 800.274.2711

To learn more visit [WoltersKluwerFS.com](http://WoltersKluwerFS.com).

© 2014 Wolters Kluwer Financial Services, Inc. All Rights Reserved.