

Kofax ReadSoft Collector

Customization Guide

Version: All

Date: 2018-03-15



© 2018 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table of Contents

- Table of Contents** 3
- Introduction** 5
- Updating customizations from COLLECTOR 6-2 to ReadSoft Collector 6.5** 5
- Customizations overview** 5
- Customization basics** 6
 - Settings on the job level 7
- Developing ReadSoft Collector plugins** 8
- Reference** 9
 - ICollectorPlugin 9
 - ICollectorPluginEx 10
 - Flow chart 11
 - CollectorPluginConfiguration 12
 - JobConfiguration and IPluginConfiguration 13
 - TreatEachAttachmentAsASeparateInvoice method 14
 - SampleCollectorPlugin – the sample project 14
 - General idea 14
 - Exception handling 17
 - Exceptions in ReadSoft Invoices 17
 - Exceptions in ReadSoft Documents 18

Introduction

This document introduces the customization of ReadSoft Collector. Plugins can be developed to be valid for both ReadSoft Invoices and for ReadSoft Documents at the same time. The user interface in the two applications are very similar.

This document is for people who:

- Develop plugins for ReadSoft Collector.
- Install and use plugins for ReadSoft Collector.

Note: ReadSoft Collector 6.5 does not work with ReadSoft Documents. However, the process of writing and connecting plugins is the same as in the previous version.

Updating customizations from COLLECTOR 6-2 to ReadSoft Collector 6.5

For plugins that were used with COLLECTOR 6-2 to work with ReadSoft Collector 6.5, references must be updated and the plugins must be recompiled.

Customizations overview

ReadSoft Collector is a plugin to ReadSoft Invoices and an add-on to ReadSoft Documents. The customizations discussed in this document apply to ReadSoft Collector only. However, it is possible to access the hosting application's API as well.

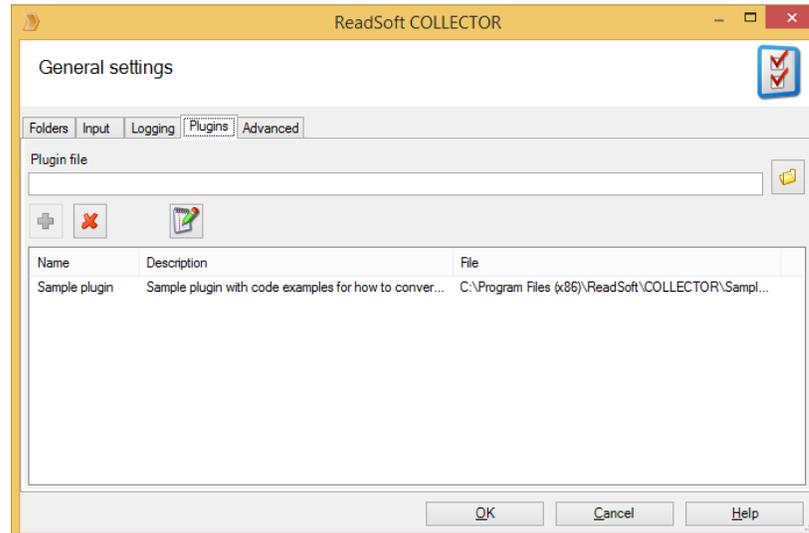
These scenarios have been identified as suitable for writing customizations for ReadSoft Collector:

- Converting email-attachment file formats
- Checking email attachment files
- Extracting data from email attachments and email, and applying it to invoices (UDVs, Queues, Buyers), etc.
- Replace the image for XML invoices (for example, use the PDF image that came in the same email.
- Create the XML invoice image based on the XML/SXML.
- Use the email body RTF or HTML.

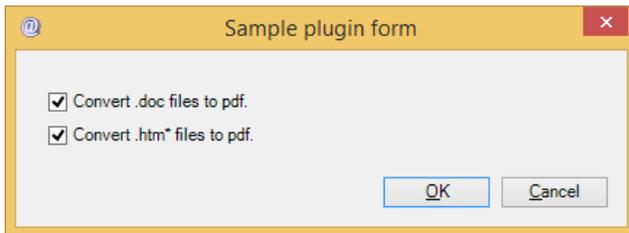
Customization basics

The process to enable customizations in ReadSoft Collector is very straight forward. The entire configuration can be done in the user interface. To enable a customization in ReadSoft Collector, do the following.

1. Copy the plugin for ReadSoft Collector to the ReadSoft Collector installation folder (on every machine where you run ReadSoft Collector).
2. Open ReadSoft Invoices Manager or ReadSoft Documents Administration modules.
 - In ReadSoft Invoices - In the main menu, select **Plugins > ReadSoft Collector**. Select the **Plugins** tab.
 - In ReadSoft Documents – On the Configuration tab, expand **System specifications > Add-ons**. In the right pane, double-click the ReadSoft Collector add-on. In the dialog that is displayed, select **Configuration > Plugins**.



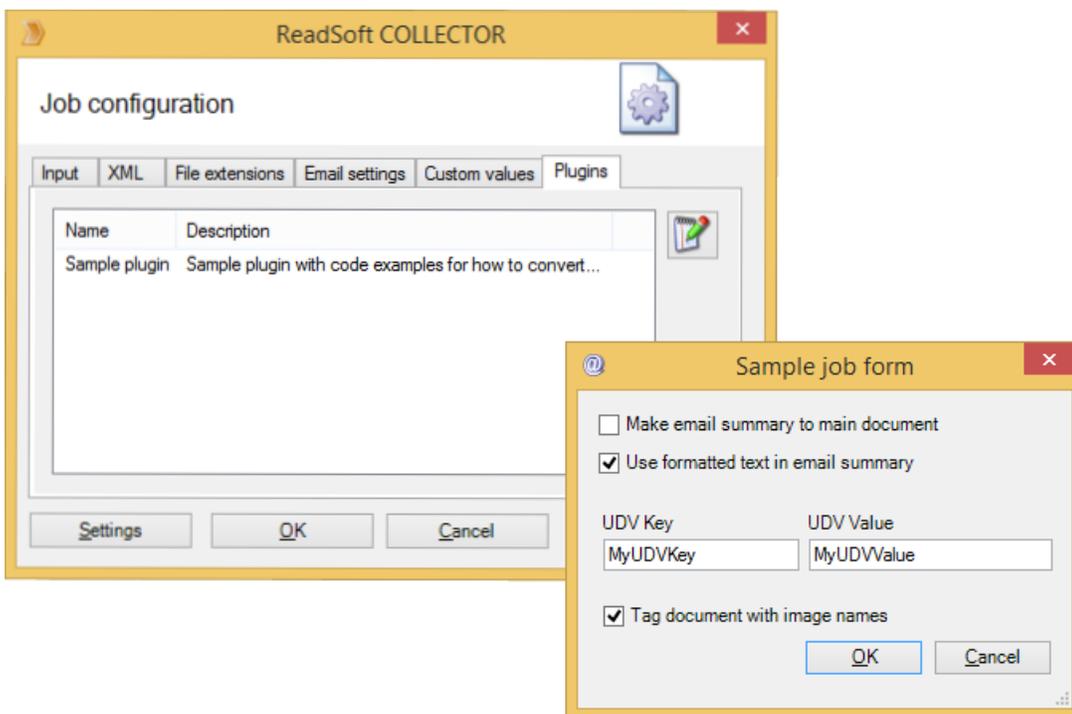
3. Press  to the right of the text box called *Plugin file*, and select the plugin you just copied.
4. Press . This actually loads the plugin and adds it to the list.
5. Select the Plugin that you added in the list.
6. Press the  to set plugin specific settings.



This is the "Sample plugin" general dialog. It applies to all jobs/tasks.

Settings on the job level

1. Open the desired ReadSoft Invoices job, select the **Interpret** > **Source** dialog page and click **Configure**.
2. In the **Plugins** tab you see all plugins that you added above.
3. Select a plugin and click the Edit button .
4. Now you can configure settings in the plugin that are unique for this particular job.



Developing ReadSoft Collector plugins

To develop plugins for ReadSoft Collector, it is more or less required that you have Microsoft Visual Studio 2008 or later installed. This chapter explains how to make a simple plugin.

1. Create a Visual Studio .NET class library.
2. Add references to
 - ReadSoft.Collector.Plugin.dll,
 - ReadSoft.Collector.Core.dll, and
 - ReadSoft.Collector.Logging.dll
 - System.Windows.Forms

All of these are located in the ReadSoft Collector folder.

- If you want to interact with ReadSoft Invoices, it might be just as good to reference `EHICOM.dll` (COM library).
 - If you want to interact with ReadSoft Documents it might be just as good to reference `RSAdvancedAPI` and `RSCCommonAPI` (.NET Libraries)
3. In one of your classes, you need to implement either the `ReadSoft.Collector.ICollectorPlugin` or `ReadSoft.Collector.ICollectorPluginEx` interfaces. The latter makes it possible for you to customize the image for XML invoices.
Tip: By right-clicking the interface and selecting to implement it, Visual Studio will create stubs for you, and it will save you some work. Note, however, that Visual Studio cannot create stubs for all methods. We will work on that when time allows.
 4. Replace the `NotImplementedException`-code with your own code. Look at the sample plugin for some ideas.
 5. Compile the plugin, and put it in your ReadSoft Collector folder. Then you can start to use it.
 6. **Tip!** – Use ReadSoft Collector’s logging functionality.
 - Add Reference to `ReadSoft.Collector.logging`

Use `CodeLogger` class (see the examples in the Sample plugin code). Then write your plugin name in each log entry so that your logs are easy to find.

Reference

ICollectorPlugin

The following tables list the members exposed by `ICollectorPlugin`.

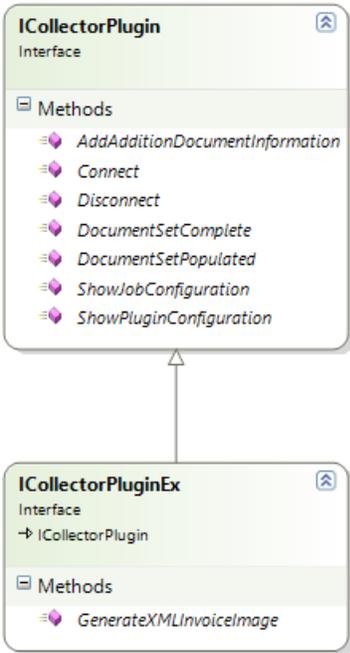
Public Methods

Name	Description
<code>AddAdditionDocumentInformation</code>	This method is called once per invoice/document after <code>DocumentSetPopulated</code> . This is a good place to put functionality such as setting ReadSoft Invoices or ReadSoft Documents properties, UDVs, Buyer, Queues, etc. In ReadSoft Invoices the <code>AppObject</code> will be the application object. In ReadSoft Documents the <code>AppObject</code> is the Document. This method is not called for XML invoices.
<code>Connect</code>	This method is called when the plugin is first created.
<code>Disconnect</code>	This method is called when the hosting application (ReadSoft Invoices or ReadSoft Documents) is shutting down. Connections, etc., should be closed here.
<code>DocumentSetComplete</code>	This method is called when the hosting application ReadSoft Invoices has interpreted the document/page. Or when ReadSoft Documents has fully imported the document. You can retrieve info but not update the invoice/document object (like <code>InvoiceID</code>).
<code>DocumentSetPopulated</code>	This method is called before the hosting application receives the document/page, but after the document set has been populated (once per email).
<code>ShowJobConfiguration</code>	This method is called when the user presses the edit-button in the ReadSoft Collector job configuration dialog.
<code>ShowPluginConfiguration</code>	This method is called when the user presses the edit-button in the ReadSoft Collector General Settings dialog.

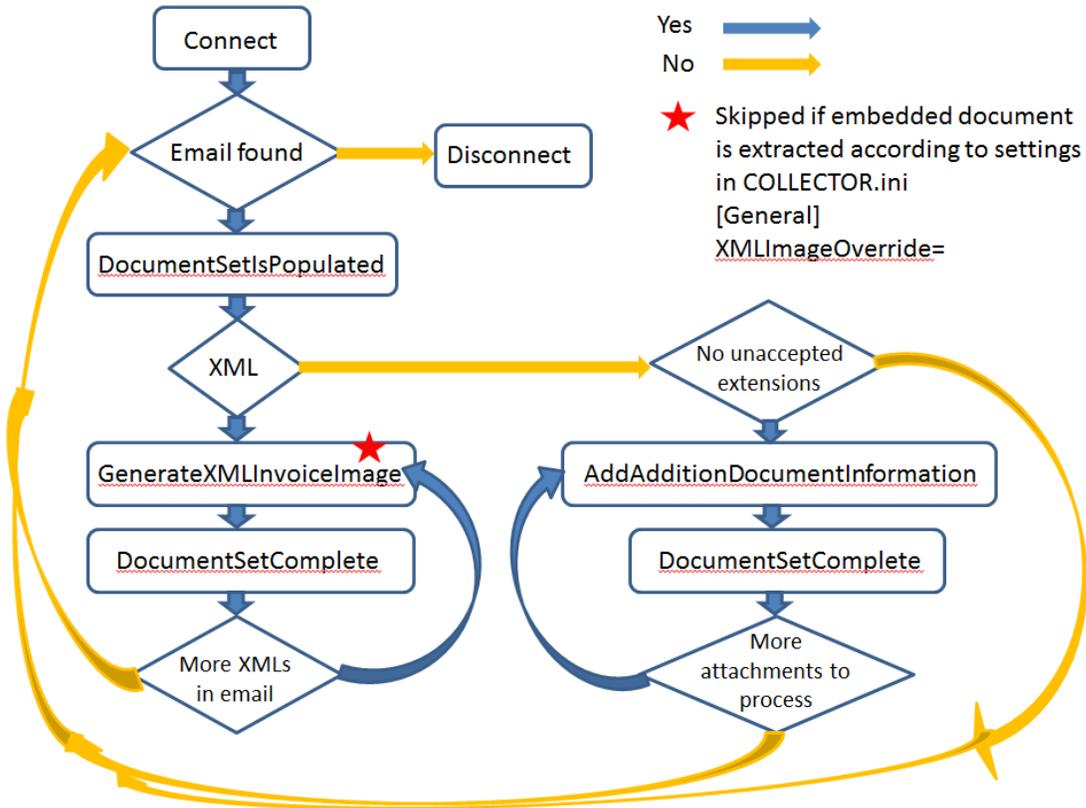
ICollectorPluginEx

Public Method

Name	Description
GenerateXMLInvoiceImage	<p>This method is called directly before the image is to be generated by Collector for the XML invoice. The plugin can override the internal image generation in two ways:</p> <ol style="list-style-type: none">Create its own using the XML or SXML.Point to an existing image file (like a PDF that is included in the same mail). <p>Notes:</p> <ul style="list-style-type: none">The image supplied by the plugin must be in a format that ReadSoft Invoices supports.This is not supported in ReadSoft Documents.If the plugin fails to create a file (that is, Collector cannot find the image), Collector's standard image is used so that the XML can be processed anyway.ReadSoft Invoices/COLLECTER may not retain the reference to the original image file throughout the entire life of the invoice processing cycle. If the invoice is reinterpreted, for example, the plugin may no longer know where all email attachments are stored and may not be able to extract the image correctly as a result. If the plugin expects to use an existing image file (like a PDF that is included in the same mail), it must make sure that the reference to this file is preserved by the plugin code (by using a UDV variable, for example).

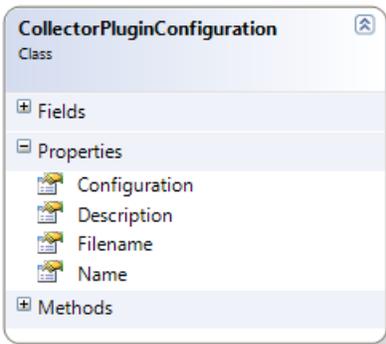


Flow chart



CollectorPluginConfiguration

This is the class that stores the plugin configuration. Typically, the **Filename** is set as adding the plugin in the UI (by ReadSoft Collector), the **Name** and **Description** properties are set in `Connect` (See `ICollectorPlugin`), and what is specified for them is displayed in the Plugins tabs for both the *General settings* and *Job configuration* dialogs. **Configuration** is where the plugin can store its own configuration. Please note that this is the only place where configuration information regarding the plugin is stored, so even information regarding job configurations will be stored in this class. See example in the “SamplePlugin”.



IJobConfiguration and IPluginConfiguration

These interfaces are used internally and are provided as parameters to some methods in `ICollectorPlugin`. What you see in the user interface reflects what these contain. However, please note that some of the settings might not be used.

The image displays two side-by-side windows from a software application, each showing a list of configuration properties for a specific interface.

IJobConfiguration Interface

- AcceptedExtensions
- AcceptEmailsWithNoAttachments
- BatchLimit
- BatchLimitInfo
- BodyAsImage
- CustomizedImage
- DefaultAcceptedExtensions
- DefaultIgnoredExtensions
- FilePrefix
- HandleAttachedEmails
- HandleAttachmentsToXML
- IgnoreAll
- IgnoredExtensions
- InputFilter
- InputSourceConfiguration
- InputSourceConfigurationName
- JobId
- KeepOriginalName
- LookForTwoSidedFileNames
- MailFilterName
- OverrideAttachmentLocation
- TreatEachAttachmentAsOneDoc
- UDVMappings
- UseShortFileFormat
- UseXSL
- XMLAttachmentLocation
- XmJobSettings
- XSLFile

Methods

- IsIgnoredImageFileExtension
- IsSupportedImageFileExtension

IPluginConfiguration Interface

- ArchiveDirectory
- DefaultIMAPConfiguration
- DefaultIMAPLLConfiguration
- DefaultLoggingConfiguration
- DefaultMAPIConfiguration
- DefaultOldEmailConfiguration
- DefaultSMTPSettings
- DefaultXMLFileConfiguration
- Filters
- HandleError
- InputSourceConfigurations
- LoggingConfigurations
- PluginConfigurations
- ReprocessFailingEmails
- TempDirectory
- UseConcurrencyLocking

TreatEachAttachmentAsASeparateInvoice method

This method has changed and results in the following:

Value in ReadSoft Invoices	Results in:
True	Treat each email attachment as a separate invoice
False	Treat entire email as a single document (including all attachments)

Value in ReadSoft Documents	Results in:
True	Create a new document for each email attachment
False	Use ReadSoft Documents sorting function to separate document pages

SampleCollectorPlugin – the sample project

General idea

The `SampleCollectorPlugin` can do several things:

- A. Converts Microsoft Word document files to PDF files for all jobs/tasks.
- B. Converts *.htm* files to PDF files for all jobs/tasks.

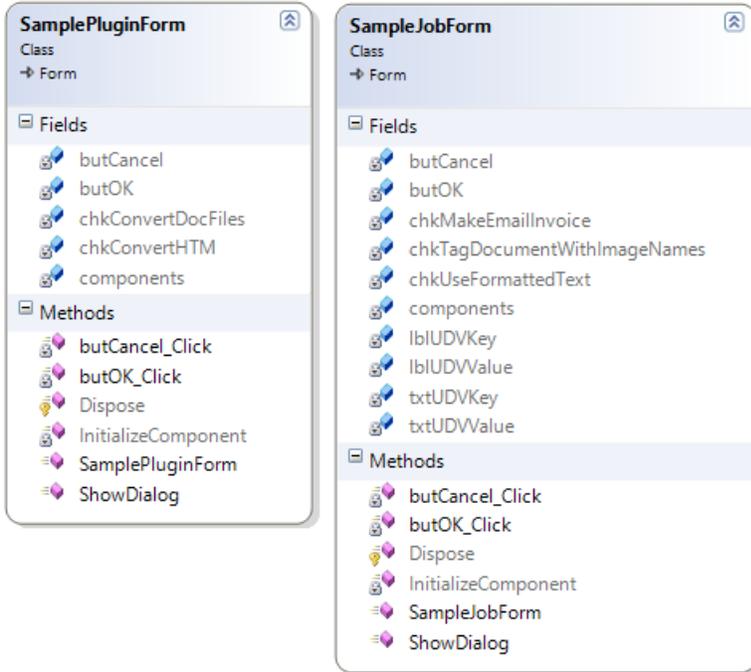
On the job level, the sample plugin can be configured to do these:

- C. Uses the email body text as the main document before the email attachment.
- D. Retrieves the RTF or HTML formatted email body and replace it with formatted text.
- E. Adds a job-specific UDV. This is for inspiration only as to what a plugin can do. The UDV's functionality within ReadSoft Collector is more flexible than the sample plugin.
- F. Tags each document that has a UDV with the value = image filename.

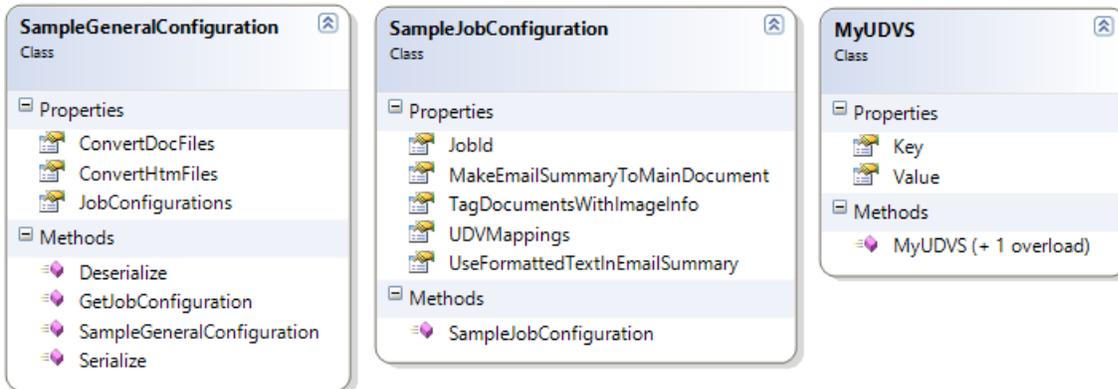
In addition, it replaces the XML invoice image with the first document that is embedded in the XML (if there is one). If the XML does not contain an embedded document, it then looks for the first PDF file in the email (if there is one) and uses that as the invoice image. This is done automatically, and there is no GUI to turn this feature on/off.

Note: Microsoft Word 12 (2007) or higher must be installed for the conversions A, B, and D, mentioned above, to work.

To show the configuration on the general level and on the job level, these two classes will do the implementation.

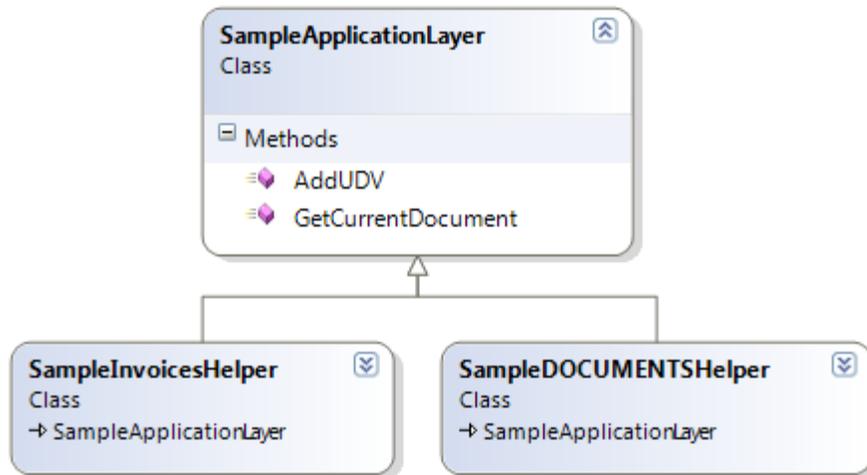


To store all the configuration information, the following classes are used.

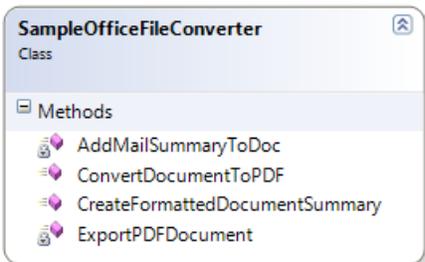


[SampleConfiguration.cs](#)

A hosting application layer is introduced so that the code apart from the layer can be identical for both ReadSoft Invoices and ReadSoft Documents.



The SampleOfficeFile converter is used to convert a Microsoft Office Word document to PDF. A trick is to record a macro within Microsoft Word when performing the operation, and then convert it to C#.



Exception handling

The ReadSoft Collector plug-in framework will catch any .NET exceptions thrown by a plug-in and take appropriate actions. Below follows a list of the measures taken when the plug-in throws an exception in the different events for each hosting product.

Exceptions in ReadSoft Invoices

Event	Action
<code>AddAdditionDocumentInformation</code>	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this and the job stops, unless the ReadSoft Invoices <code>continuejobonerror</code> ini-setting has been set to 1. Note that if there were attachments in the mail already processed by ReadSoft Invoices from this email, the email will be half-processed.
<code>Connect</code>	When there is an exception in this method ReadSoft Invoices reports that ReadSoft Collector could not be initialized and the error is logged. ReadSoft Collector will not be available from the Interpret module but it can be configured from the Manager module.
<code>Disconnect</code>	When there is an exception when calling this method of the plug-in, the error is logged, but the disconnecting operation proceeds as normal.
<code>DocumentSetComplete</code>	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this and the job stops, unless the ReadSoft Invoices <code>ContinueJobOnError</code> ini-setting has been set to 1. Note that if there were attachments in the mail already processed by ReadSoft Invoices, those invoices will remain processed by ReadSoft Invoices just as the one for which the error occurred.
<code>DocumentSetPopulated</code>	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this and the job stops unless the ReadSoft Invoices <code>continuejobonerror</code> ini-setting has been set to 1.
<code>GenerateXMLInvoiceImage</code>	When an image is not generated by the plugin, an error log is generated, and Collector continues to create the standard image.
<code>ShowJobConfiguration</code>	When there is an exception when configuring the

	job and plug-in using this method ReadSoft Invoices reports an error in a dialog and logs details about the error.
ShowPluginConfiguration	When there is an exception when configuring the plug-in using this method, there is a dialog saying the plug-in could not be configured and the error is logged.

Exceptions in ReadSoft Documents

Event	Action
AddAdditionDocumentInformation	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this as well as details about the exception, and the task stops. The document that this occurs for is discarded.
Connect	<p>If this exception is thrown in the administration module when configuring the plug-in, ReadSoft Collector will report that the plug-in could not be initialized and the details about the error are logged.</p> <p>If this exception is thrown in the administration module when starting the module it will report that ReadSoft Collector could not be initialized correctly and ReadSoft Collector is shown as having a failed Load Status in the Add-on management dialog. Also details about the error are logged.</p> <p>If this exception is thrown in the production module when starting the module it will report that ReadSoft Collector could not be initialized correctly and ReadSoft Collector is shown as having a failed Load Status in the Add-on management dialog. Also details about the error are logged.</p>
DocumentSetComplete	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this as well as details about the exception, and the task stops. The document that this occurs for is discarded.
DocumentSetPopulated	When there is an exception thrown in this method the corresponding email is moved to the error folder, there is an error log reporting this, as well as details about the exception, and the task stops. The document set, that is, the entire email, that this occurs for is discarded.

<code>ShowJobConfiguration</code>	When there is an exception when configuring the task and plug-in using this method the administration module reports an error in a dialog and logs details about the error.
<code>ShowPluginConfiguration</code>	When there is an exception when configuring the plug-in using this method there is a message saying the plug-in could not be configured and the error is logged.