Functional Design

Exchequer 2016 R2 - eRCT Plugin

Copyright © Advanced Computer Software Group Ltd 2016

This document contains confidential and / or proprietary information. The content must not be disclosed to third parties without the prior written approval of Advanced Computer Software Group Limited or one of its subsidiaries as appropriate (each referred to as “Advanced”). External recipients may only use the information contained in this document for the purposes of evaluation of the information and entering into discussions with Advanced and for no other purpose.

Whilst Advanced endeavours to ensure that the information in this document is correct and has been prepared in good faith, the information is subject to change and no representation or warranty is given as to the accuracy or completeness of the information. Advanced does not accept any responsibility or liability for errors or omissions or any liability arising out of its use by external recipients or other third parties.

No information set out or referred to in this document shall form the basis of any contract with an external recipient. Any external recipient requiring the provision of software and/or services shall be required to enter into an agreement with Advanced detailing the terms applicable to the supply of such software and/or services and acknowledging that it has not relied on or been induced to enter into such an agreement by any representation or warranty, save as expressly set out in such agreement.

The software (if any) described in this document is supplied under licence and may be used or copied only in accordance with the terms of such a licence. Issue of this document does not entitle an external recipient to access or use the software described or to be granted such a licence.

The development of Advanced software is continuous and the published information may not reflect the current status. Any particular release of the software may not contain all of the facilities described in this document and / or may contain facilities not described in this document.

Advanced Computer Software Group Limited is a company registered in England and Wales with registration number 05965280 whose registered office is at Ditton Park, Riding Court Road, Datchet, Berkshire. SL3 9LL.

A full list of its trading subsidiaries is available at www.oneadvanced.com/legal-privacy

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Issued By | Changes |
| 07/07/2016 | 1.0 | Chris Sandow | Initial release |

Version Approval

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Name | Title / Organisation | Approval Record |
|  |  |  |  |

Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Name | Title |
|  |  |  |  |

Document Control

|  |  |  |
| --- | --- | --- |
| Filename | Location | Minimum Retention |
|  |  |  |

Contents

[eRCT Plugin 1](#_Toc455647424)

[Overview 1](#_Toc455647425)

[Technical Details 1](#_Toc455647426)

[COM Interfaces 1](#_Toc455647427)

[Process Flow 2](#_Toc455647428)

[interface IRCTGateway 6](#_Toc455647429)

[IRCTGateway Properties 6](#_Toc455647430)

[IRCTGateway Methods 7](#_Toc455647431)

[RCT Status Codes 7](#_Toc455647432)

[IRCTGateway Property Errors 7](#_Toc455647433)

[TagTransaction() Errors 7](#_Toc455647434)

[interface IRCTPlugin 7](#_Toc455647435)

[MenuCaption: string 8](#_Toc455647436)

[Start(Gateway: IRCTGateway): bool 8](#_Toc455647437)

[OnHeaderError(IRCTGateway Gateway, TRCTStatusCode StatusCode, string Description): bool 8](#_Toc455647438)

[Process(Gateway: IRCTGateway): bool 8](#_Toc455647439)

[Finish(Gateway: IRCTGateway) 8](#_Toc455647440)

[Appendix I: Creating a C# eRCT Plugin 8](#_Toc455647441)

# eRCT Plug-in

## Overview

The eRCT Plug-in system in Exchequer allows an external plugin to create and control a Batch Payments run, tagging the transactions that it wants to generate Purchase Payments from.

There are two COM objects involved, one being the Exchequer COM Customisation object which provides the RCT Gateway object that handles the Exchequer side of the system. The other is a plugin, RCT.Plugin, which is created by a third-party and which controls the external side.

When an RCT Plugin is registered with Windows, Exchequer will detect this and will add a menu item to the Procedures menu of the main Exchequer window, using a caption supplied by the plug-in.

When a user selects this menu item, Exchequer communicates with the RCT Plugin, first to get the header details (the equivalent of the details that a user would enter in the Batch Payments setup dialog), and then to get the transactions which should be tagged.

Finally the Batch Payments process is run to generate the Purchase Payment transactions.

## Technical Details

### COM Interfaces

The Exchequer COM Customisation object defines two interfaces to support this system, the IRCTGateway interface and the IRCTPlugin interface.

The IRCTGateway interface is implemented by Exchequer, and an instance of the object which supports this interface is passed to the plugin methods.

The IRCTPlugin interface is implemented by a third-party and is responsible for setting up the Batch Payment Header details (via properties on the IRCTGateway instance), and then supplying the OurRefs of the transactions which are to be tagged. It finally calls a ProcessPayment method on the IRCTGateway instance, which then performs the actual Batch Payments processing to generate the required Purchase Payments.

Note that the COM object which implements this interface **must** be given the name RCT.Plugin (in C# this is generally done by using a namespace of "RCT" and a class name of "Plugin").

### Process Flow

The plugin (if present) is loaded when Exchequer starts, and the MenuCaption method is called to get the caption to use for the RCT Plugin menu option. If no plug-in is registered the menu item is hidden.

When the user selects this menu item, Exchequer checks that a Batch Payment run is not already active. If it finds one, it displays a message to the user and ends the process (the plug-in is never called).

Assuming that there is no Batch Payment process running, the Start() method of the plug-in is called, passing the RCT Gateway object to it. The plug-in must then set up any required details for the Gateway, and then return True to indicate that the process can continue.

If the plug-in returns False from the Start() method, the process is stopped, and the Finish() method is called to allow the plug-in to do any tidy up.

Exchequer then validates the header details. If it finds any errors, it calls the OnHeaderError() method of the plugin, passing IRCTGateway object and the error details to it.

The plugin can correct the error and return True, in which case Exchequer will perform the verification again (and if it still finds errors will call OnHeaderError() again, until all the errors are corrected, or OnHeaderError returns False.

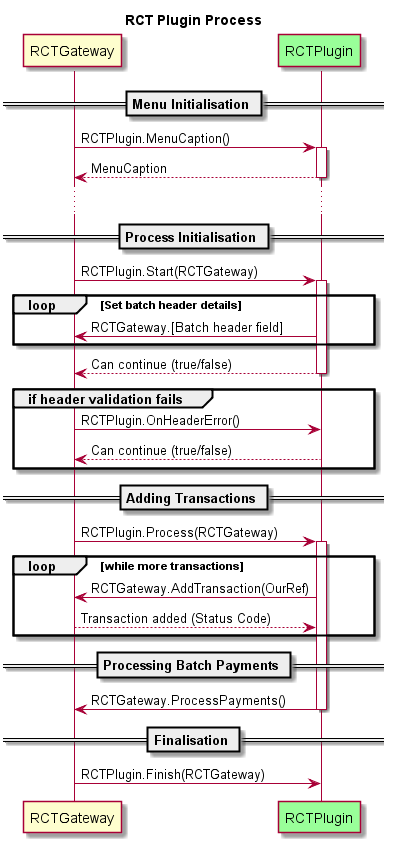
If OnHeaderError() returns False, the process is stopped, and the Finish() method is called to allow the plug-in to do any tidy up.

Assuming the process can continue, Exchequer then calls the Process() method on the plug-in, passing the IRCTGateway instance to it. The plug-in should now call TagTransaction() on the Gateway for each transaction that it wants to tag. This will return 0 if the tag was successful, otherwise it will return an error code (currently the only error is rsTransactionNotFound).

The Process() method can return False if it wants to cancel the process at this stage. The Finish() method will be called to allow the plug-in to do any tidy up.

Otherwise the Process() method should call ProcessPayments(). This will launch the Exchequer Process Batch Payments routine, which will generate Purchase Payments based on the transactions that the plug-in has tagged.

Once this process is completed, the Finish() method will be called to allow the plug-in to do any tidy up.



## interface IRCTGateway

This is the interface provided by Exchequer. An object supporting this interface will be passed to the plugin as a parameter to all the methods (except the MenuCaption method). The plugin can then call the methods on this interface to set up the Batch Payments run, tag transactions in Batch Payments, and to trigger the processing of the Batch Payments run.

### IRCTGateway Properties

These match with the fields on the Batch Payments settings dialog. They will be assigned valid default values, but the plugin should set the values that it needs (in the Start() method), and on return from the IRCTPlugin.Start() method they will be validated to make sure that any changed values are still valid.

* PaymentType: string
  + "C" - Cheque
  + "B" - BACS
  + "2" - Cheque, Alt 1
  + "3" - Cheque, Alt 2
* BankGLCode: int
* GLControlAccount: int
* CostCentre: string
* Department: string
* UseAccountCCDept: bool
* InvoiceCurrency: int
* PaymentCurrency: int
* AgeBalancesBy: int
  + abDays
  + abWeeks
  + abMonths
* AgeInterval: int
* UseChequePrinting: bool
* ChequeNumber: string
* YourRef: string
* IncludePPD : bool
* PPDPaymentDate : string
* PPDExpiryTolerance : int
* IncludeSettlementDiscount: bool
* DaysOverSettlementDiscount: int

#### TBatchPaymentAgeType

This enumeration is for the values for the AgeBalancesBy property.

* abDays
* abWeeks
* abMonths

### IRCTGateway Methods

#### TagTransaction(OurRef: string)

This tags the transaction identified by the supplied OurRef. If no matching transaction can found, the method will return an rctTransactionNotFound status code.

#### ProcessPayments()

This launches the final Batch Payments Process routine to generate the Purchase Payments from the transactions that have been tagged.

## RCT Status Codes

These codes are returned to the plug-in in the event of errors.

### IRCTGateway Property Errors

The RCT Gateway properties will be validated on return from the Start() method of the plugin, which is expected to have set the relevant property values on the IRCTGateway object.

The following errors will be passed to the plug-in's OnHeaderError method if any invalid values are found:

* rsInvalidGLCode
* rsInvalidCostCentre
* rsInvalidDepartment
* rsnvalidBankGLCurrency
* rsInvalidPPDPaymentDate

### TagTransaction() Errors

This error code will be returned by the TagTransaction() method of the IRCTGateway object if the specified transaction could not be found.

* rsTransactionNotFound

## interface IRCTPlugin

This is the interface that the plugin is required to support

### MenuCaption: string

This method is called by Exchequer at start-up to get the caption which should be displayed for the RCT Plugin menu option.

### Start(Gateway: IRCTGateway): bool

This method is called to start the process. It is passed the IRCTGateway instance, and should set up the correct header values via the properties on this instance (see IRCTPlugin Properties above).

### OnHeaderError(IRCTGateway Gateway, TRCTStatusCode StatusCode, string Description): bool

This is called if an error is found in the Batch Payments Header values that were set up in the Start() method. It should return True if the header value has been corrected and the process can continue. If it returns False the process will be cancelled.

### Process(Gateway: IRCTGateway): bool

This is called when the Batch Payments system is ready for transactions to be tagged. It should call AddTransaction() on the IRCTGateway instance for every transaction that should be tagged, and finally should call the ProcessPayments() method on the IRCTGateway instance to carry out the final processing.

Returning False will abandon the process, unless ProcessPayments() has already been called.

### Finish(Gateway: IRCTGateway)

This is called once all the processing has completed. It also called if the process is abandoned at any point.

## Appendix I: Creating a C# eRCT Plugin

* Create new C# project, of type Class Library
* Change the namespace to 'RCT'
* Change the class name to 'Plugin'
* Go to Project Properties
* On the Application tab, click Assembly Information
* Tick the 'Make assembly COM-visible' checkbox
* On the Build tab, tick the 'Register for COM interop' checkbox
* Add a reference to the Enterprise COM Customisation interop assembly
* Add a reference to Exchequer COM Toolkit (DLL)
* In the properties for this reference, set Embed Interop Types to False
* Implement the IRCTPlugin interface in the Plugin class

Note that the plug-in does not need to be registered with Exchequer in any way. As long as it is a registered COM component (which it will be if the above steps are followed) Exchequer will detect and load it.