Reference Guide

Real Time Integration

By CMiC

CMiC
Computer Methods
international Corp.

Proprietary Notice

The contents of the CMiC software product, including both this manual and the program components, are proprietary to Computer Methods International Corp. (CMiC), are copyright protected and are considered Confidential Information by CMiC. All rights are reserved by CMiC. A copy of the manual and the program has been provided to the original licensee under license with CMiC strictly for their own use under the terms of the license. All copies of this manual whether in print or electronic format must contain a copy of this Proprietary Notice. Any selling, licensing or other distribution of the contents of either this manual or the program components, whether for profit or not, is unlawful and may subject the violator and the original licensee to termination of license, criminal charges, civil action, or any combination of these.

Copyright © 2020

Computer Methods International Corp.
4850 Keele Street
Toronto, Ontario M3J 3K1
Canada

Risk of Use Notice

The CMiC software product, including both this manual and the program components, is licensed on an "AS IS" basis. The entire risk as to the results of its use is with the licensee. Except in those jurisdictions which impose certain warranties by statute which may not be waived by one or more of the parties, and only to that extent, Computer Methods International Corp. (CMiC) makes no warranties whatsoever, either expressed or implied, with respect to the quality, performance, merchantability or fitness for any particular purpose of any or all components of this software product, except as provided in the licensee's license agreement. The licensee (and not CMiC or its agents) will be solely responsible for the costs of all service, or of any defect in this software product and any incidental or consequential damages caused or alleged to be caused either directly or indirectly by the software product to the licensee or any other person, including, but not limited to, any interruption of service, or loss of business or anticipatory profits, even if CMiC has been advised of the possibility of such damages.

"Computer Methods International Corp" and "CMiC" are registered trademarks of Computer Methods International Corp. Oracle, Oracle9i™, Oracle Application Server11g™, Oracle Database 11g™, Oracle® Discoverer™ are trademarks or registered trademarks of Oracle Corporation.

Contents

Real Time Integration

Overview – Real Time Integration

CMiC RTI is a tool that allows for the bi-directional updating of databases via XML standards. Called CMiC Real Time Integration (RTI), the tool flows data entered into the general contractor's system to the Owner's system by Internet and XML protocols, allowing for different systems to seamlessly communicate.

CMiC RTI doesn't just create records in both databases, but it updates them both as well, which means that when an Owner is answering an RFI in their system, the answer is appearing, in real-time, in the general contractor's system.

XML provides a basic syntax that can be used to share information between different kinds of computers, different applications and different organizations without needing to pass through many layers of conversion. CMiC RTI will save countless hours by eliminating the need to re-enter information into different systems when collaborating on a project. The elimination of data re-entry will increase data accuracy, improve productivity and enhance collaboration. CMiC RTI is currently bi-directly communicating with two CMiC Project Management databases, but in the future CMiC RTI will also support multiple platforms through standard XML formats such as AGCxml.

This document describes the CMiC Real Time Integration (RTI) which is in essence a web service architecture for inter-system messaging. This architecture is designed to facilitate an efficient, robust, and secure computer-to-computer messaging system. RTI implies a sender and a receiver (sometimes called a producer and a consumer).

Our messaging system is built in 3 layers:

- The Adapter Layer
- The Workflow Engine Layer
- The Database Layer

Technical Overview

In order to maximize throughput, avoid bottlenecks, and minimize use of system resources, the CMiC RTI (Real Time Integration) Adapters Layer is implemented as a series of asynchronous processes. Each process operates independently. The processes are inter-connected via an Oracle database technology known as Oracle Streams Advanced Queuing.

Oracle Streams Advanced Queuing provides message queues and the ability to enqueue (i.e. insert into a queue) and dequeue (i.e. remove from a queue) messages.

The CMiC RTI system produces messages that are in XML format and enqueues and dequeues them using 3 RTI-specific queues:

- RTI_OUTBOUND_Q (for messages to be transmitted)
- RTI_INBOUND_Q (for messages that were received)
- RTI_ERROR_Q (for errors that occur during transmission)

There are 6 discrete processes in the RTI Adapters Layer. They are:

1. Dataflow Initiation
2. Message Dispatching
3. Message Transmission
4. Receive Adapter on Remote System Processes a Valid Message
5. Valid Message Received in Remote Database
6. Error Handling

Each process is diagrammed on the pages that follow.

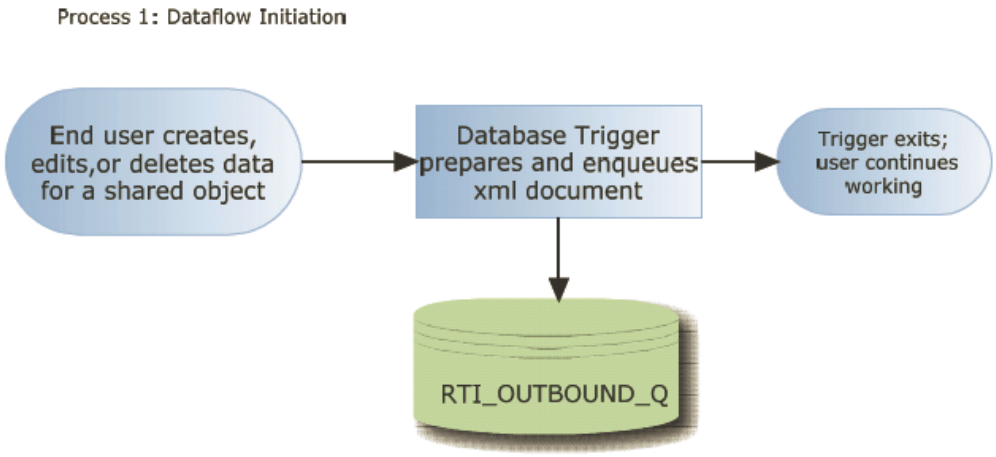
The RTI system is a transport layer that any CMiC Application can use to transmit data from one site to another. At this time we are programming a transfer of Project Management data between two sites; therefore the examples in this document refer to Project Management data transfers. In the more general case the data being transferred could be from any CMiC application module.

Also note that the business rules for what data is transferred and how it is formatted are defined in the application itself, not in the RTI Layer. This document does not describe those business rules. This document does describe Actionflow processes. It is during those Actionflow processes that the business rule logic is applied at both sites.

Besides these 6 processes, there are also file maintenance screens that allow the System Administrator on each side to define the RTI-specific information (i.e. the communication links and specific communication information) that allows the messages to be transmitted.

Technical note about queue processing: In the diagrams that follow you will see that the CMiC programs that are reading messages from the Oracle Queues are called RTI Message Dispatch processes. Although the diagram makes it appear as if there is only 1 instance of any given RTI Message Dispatch process, this is a configurable parameter in the database, and thus multiple RTI Message Dispatch processes can be running concurrently to expedite messaging throughput.

Process 1: Dataflow Initiation



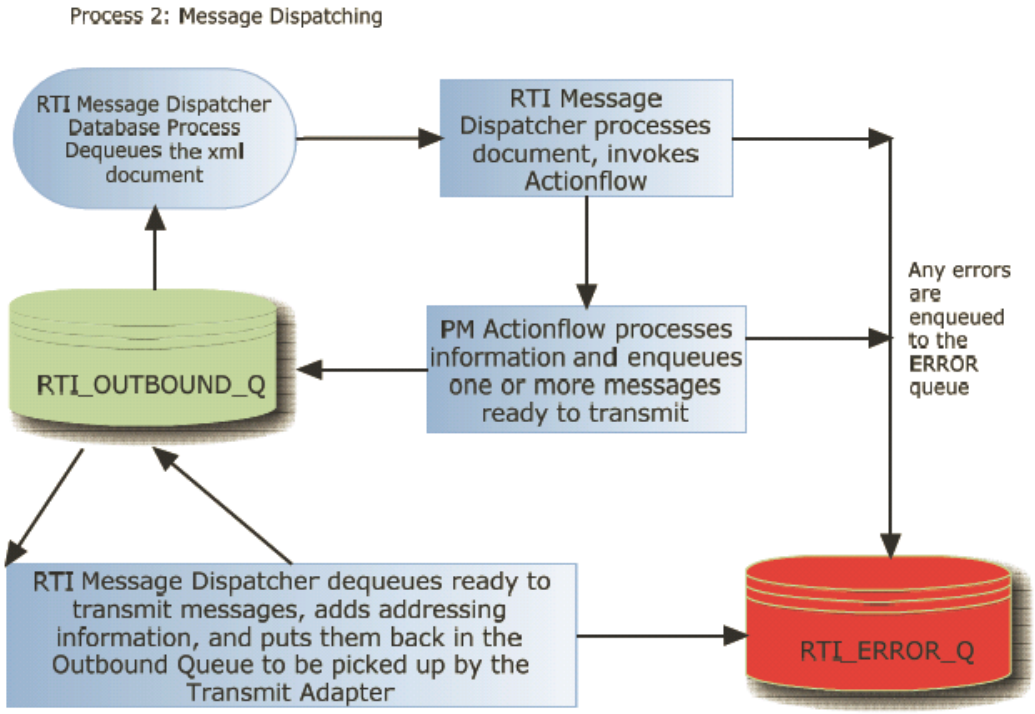
The process starts when end users make changes to data, and the application (in this case CMiC Project Management) detects that those changes were made to one or more shared objects.

A CMiC PM Database trigger fires to start the transfer process. It very quickly enqueues an XML document into the RTI_OUTBOUND_Q describing the data that changed and then returns control to the end user.

The end user is not aware that the transfer mechanism has been invoked, and is not interrupted by the RTI transfer mechanism that was just invoked on his/her behalf.

The outbound XML document will sit in the RTI_OUTBOUND_Q until a separate process dequeues it and acts upon it.

Process 2: Message Dispatching



This part is complex, because three different processes are accessing the RTI_OUTBOUND_Q. Two are RTI Message Dispatcher transmittal processes, and the third is a PM Actionflow process. The two RTI Message Dispatcher transmittal processes are each running in its own completely separate database session. Both are listening on the RTI_OUTBOUND_Q.

The first one (shown on the top row of the diagram) listens for any message produced by the database trigger in Process 1. The second one (the blue rectangular box on the bottom row) listens for messages that are enqueued by the PM Actionflow process on the middle row.

The two RTI Message Dispatcher transmittal processes run independently from each other, and collaborate to process messages and dispatch them to the Transmit Adapter for transmission.

The first RTI Message Dispatcher transmittal process listens for messages from the database trigger in Process 1, and passes them to the PM Actionflow for processing. The second RTI Message Dispatcher transmittal process listens for messages from the PM Actionflow process, adds addressing information to them, and puts them back in the queue for the Transmit Adapter to process in Process 3.

Details of Process 2:

When the database trigger in Process 1 enqueues its XML document into that queue, Oracle Advanced Queuing automatically notifies that RTI Message Dispatcher transmittal process of the new message.

If for any reason the RTI Message Dispatcher transmittal process is not running when the database trigger enqueues the message, the message will remain in the RTI_OUTBOUND_Q until it starts running, at which time it will be notified that the message is available.

At that point the first RTI Message Dispatcher transmittal process dequeues and processes that message. If it finds any errors, it enqueues an error message into the RTI_ERROR_Q. Any error at this point would be due to improper system setup, such as the information for the remote site is not setup properly.

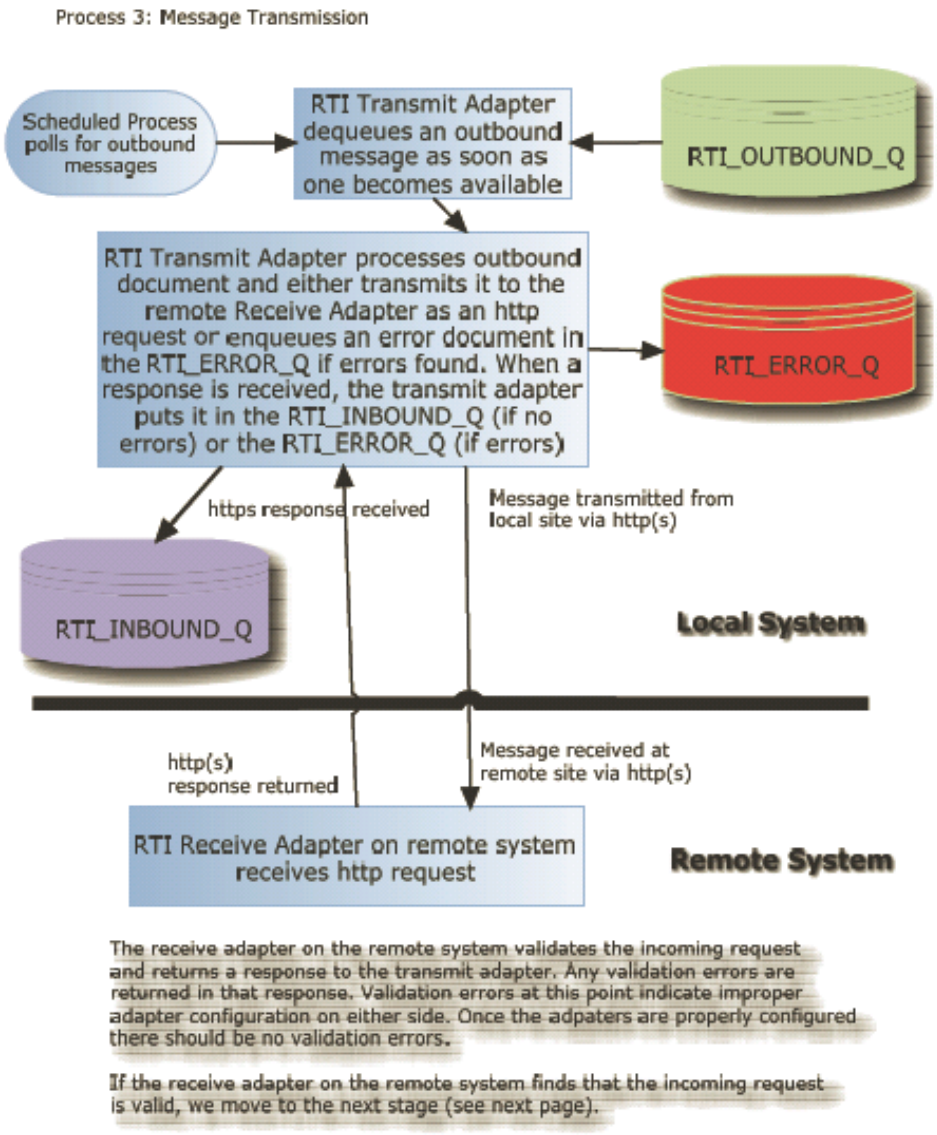
If it finds that the message is valid, it invokes an application-specific Actionflow (in this case a PM Actionflow). An Actionflow is a CMiC-proprietary programming construct that defines a series of processing steps that occur in the database to apply application specific business rules as a series of discrete events.

The PM Actionflow reads the message and validates it. If it finds any errors, it enqueues the message into the RTI_ERROR_Q. Once again, any error that it detects at this point would be due to improper setup, such as missing data mappings for mapping the PM data from one site to another. Otherwise it produces one or more messages for transmission and enqueues them back into the RTI_OUTBOUND_Q.

The first RTI Message Dispatcher transmittal process now goes back to listening on the RTI_OUTBOUND_Q for any messages produced in Process 1. During the time that it was processing the previous new messages may have been enqueued into the RTI_OUTBOUND_Q by Process 1. As soon as they arrive, this RTI Message Dispatcher transmittal process re-starts the process shown as Process 2 on the diagram to process them one by one.

Meanwhile Oracle Streams Advanced Queuing now notifies the second RTI Message Dispatcher transmittal process running of the new message that was produced by the PM Actionflow. That RTI Message Dispatcher transmittal process now adds complete addressing information to the message contents and enqueues it one more time in the RTI_OUTBOUND_Q, this time addressed to the RTI Transmit Adapter.

Process 3: Message Transmission



In this process the RTI Transmit Adapter dequeues outbound messages that are addressed to it.

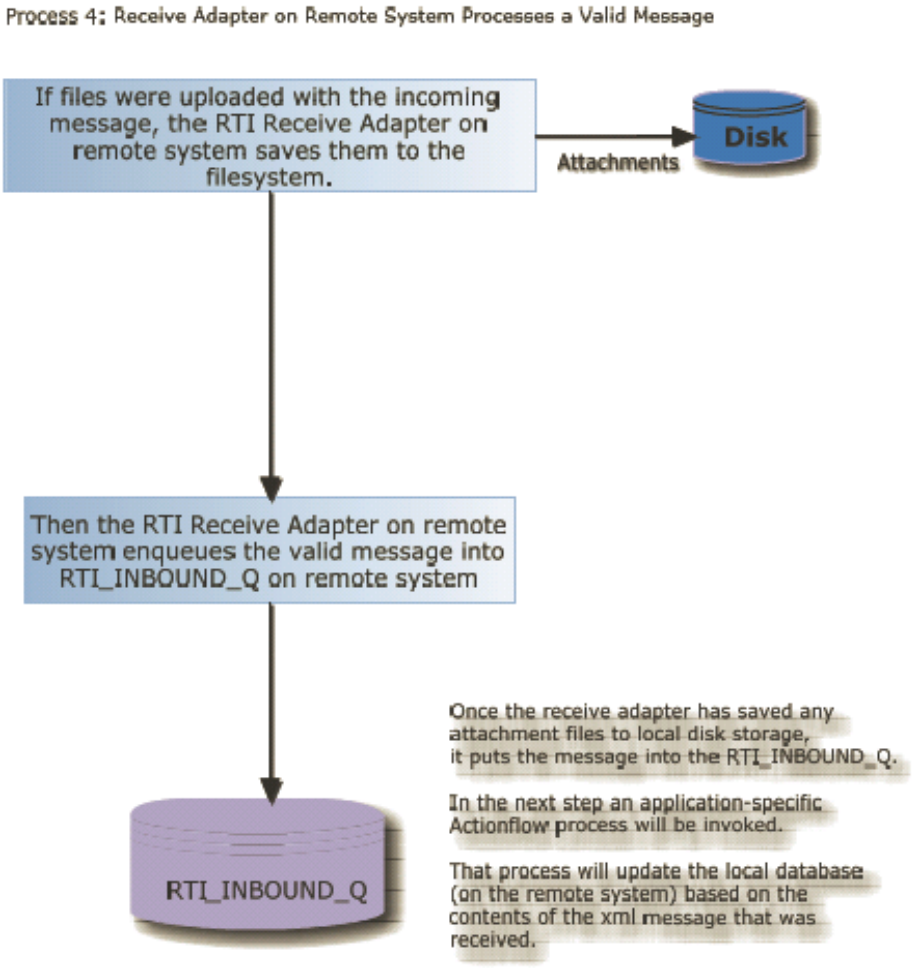
If it finds validation errors, it enqueues them in the RTI_ERROR_Q. Otherwise it attempts to transmit the message to the remote Receive Adapter.

If the message is received successfully by the remote side it gets a response from the Receive Adapter. If that response has no error messages, the Transmit Adapter enqueues the response in the RTI_INBOUND_Q as an audit trail to show that the message was successfully transmitted. If the response contains one or more error messages, the Transmit Adapter enqueues the response in the RTI_ERROR_Q.

If a transmission error occurs, the Transmit Adapter also enqueues the response into the RTI_ERROR_Q.

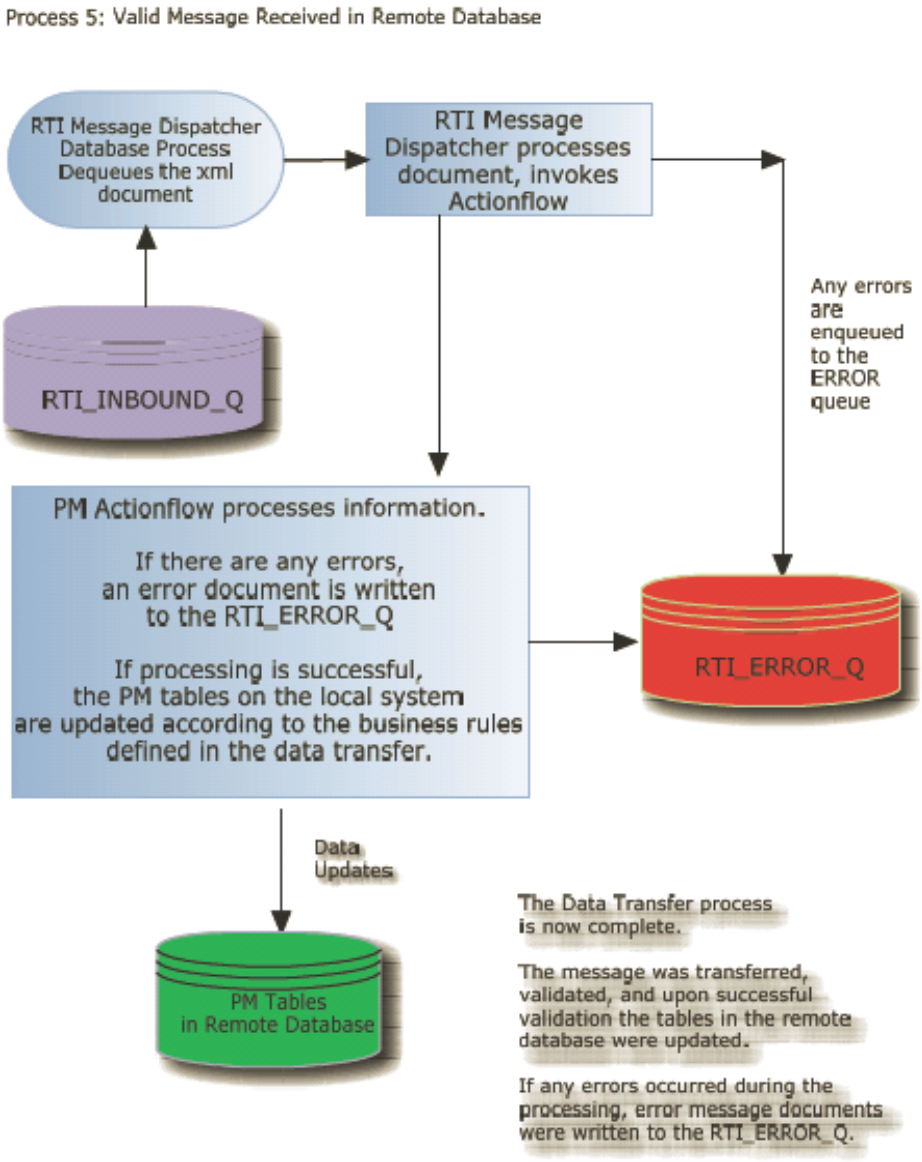
Note that if the outbound message included files to be uploaded, the transmit adapter uploads those files as well as transmitted the message to the Remote Receive Adapter.

Process 4: Receive Adapter on Remote System Processes a Valid Message



If files were transferred, the Receive Adapter on the remote side first saves them, then it enqueues the message into the RTI_INBOUND_Q on the remote side.

Process 5: Valid Message Received in Remote Database



An RTI Message Dispatcher receiver process is always listening on the RTI_INBOUND_Q. Oracle Streams Advanced Queuing automatically notifies this process when a new message arrives. Just like on the outbound side, the RTI Message Dispatcher receiver process first validates the message, enqueuing it into the RTI_ERROR_Q if any validation errors occur.

Otherwise it invokes a PM Actionflow for the inbound message. The PM Actionflow validates the message according to the PM business rules, and again enqueues it in the RTI_ERROR_Q if it finds any validation errors. Otherwise it uses the message contents to update the PM tables in its database based on the business rules for that update. Control then returns to the RTI Inbound Message Dispatcher receiver process which waits for the next message to arrive, or, if one has already arrived, immediately begins processing it.

Process 6: Error Handling

Process 6: Error Handling

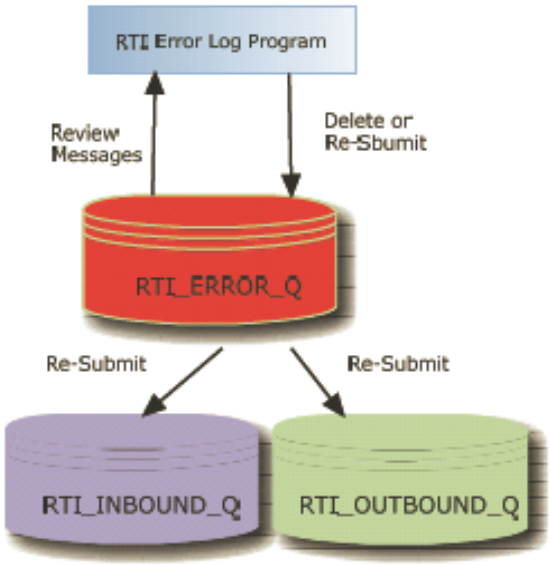
If any validation errors occur during processing, an error message document is written to the RTI_ERROR_Q.

The system administrator at either side has access to the RTI Error Log program. This program displays the error documents that were created, and allows the administrator to review the original xml documents that caused those errors.

The system administrator has 3 ways to deal with each error:
- edit data on the local system (for instance, to add missing data) and then re-submit the message.
- edit the data in the message directly and then re-submit the message
- delete the message (i.e. decide that this message is no longer important)

The records remain in the RTI_ERROR_Q until the system administrator re-submits or deletes them.

When the records are re-submitted, they go into either the RTI_INBOUND_Q or the RTI_OUTBOUND_Q and the processing cycle starts over at process 2 (OUTOUND) or process 5 (INBOUND).



If the original message that caused the error was a message that was being transmitted and the administrator re-submits it, the RTI Error Log program enqueues the message into the RTI_OUBOUND_Q and processing continues back at process 2.

If the original message that caused the error was a message that was being received and the administrator re-submits it, the RTI Error Log program enqueues the message into the RTI_INBOUND_Q and processing continues back at process 5.

Part 1 – The Adapter Layer Message Adapters

CMiC RTI is built on the concept of Message Adapters which naturally fall into two types: Receive Adapters and Transmit Adapters. Adapters are optional components that can be plugged in to any system.

Optional means that they are not required in order to operate CMiC Software, and therefore can be separately licensed. Each adapter supports a specific communications protocol, a specific messaging style, and is targeted to a specific type of system (i.e. “speaks a specific dialect” or “conforms to a particular API”).

In summary, adapters facilitate the building of a messaging interface between two disconnected systems.

Receive Adapters

A receive adapter in CMiC Software receives messages from a separate external system. The external system could be but is not necessarily a separate installation of CMiC Software.

A minimum of one receive adapter is necessary in order to perform messaging. With a single receive adapter CMiC Software can perform inter-system messaging functions with any external system software that encodes and decodes messages according to the receive adapter API. In other words, receive adapters allow external systems to talk to a CMiC Software installation.

In most cases transmission adapters will also be required. The only time a transmission adapter would not be required is when the interface only sends data in one direction, i.e. data only flows into CMiC Software. In any bi-directional data interface, a receiver and a transmit adapter are required.

Example 1:

The receipt of purchase order records into CMiC from an external ERP or legacy system. In this case, only a receive adapter is required.

Example 2:

The requirement to send and receive data from an external system operating third party software. In this case, transmit and receive adapters are required.

Example 3:

The requirement to send and receive data between two separate CMiC Software installations. In this case, transmit and receive adapters are required.

Transmission adapters, although built generically, must be written to conform to the needs (i.e. the API) of the external system. Transmit adapters allow a CMiC System to talk to a specific external system, using its dialect.

Http-Receive Adapter for CMiC Software

Each adapter supports a specific protocol, uses a particular messaging style, and is targeted to a specific system.

The Protocol

In the initial release the Receive adapter for CMiC software will support the http protocol with additional protocols to be added as necessary.

Messaging Style

There are two styles of web service messaging in popular use today. One is the WSDL/SOAP/XML-RPC style; the other is the simpler REST style. CMiC is providing an Http-Receive Adapter for CMiC software that is REST-oriented.

The System

By definition, every receive adapter we create will be receiving messages sent to a CMiC System.

Transmit Adapters

As mentioned above, in most cases we need one or more transmit adapters, each one tuned to the needs of a particular external system.

Http-Transmit Adapter for CMiC Software

The protocol and messaging style implemented by a transmission adapter are determined by the external system that it transmits messages to.

The Protocol and Messaging Style

Like the Http-Receive Adapter for CMiC Software, it will also use the http protocol and will follow a REST-oriented messaging style. The requirements on the receiver side determine how the transmit adapter is built.

Adapters are just Java servlets

Each CMiC adapter is implemented as a java servlet running in a standard J2EE container.

So who needs Adapters anyway?

The rationale for implementing a single receive adapter is easy to explain.

Messages from external systems are by definition triggered by events that occur outside of CMiC. A method to efficiently, reliably, and securely receive those messages is required. While in theory, many different hooks could be developed to receive messages, allowing each program to deal directly with the specific messages that it handles, it is much more efficient to develop one receive adapter for the entire system. All the algorithmic complexity of receiving and decoding messages, and returning responses to those messages, is isolated in one receiver program. The incoming data is always passed into CMiC in a standard way, and is available to whatever programs need to access it.

Part 2 – The Workflow Engine Layer

OK, so how are these messages actually processed inside CMiC Software?

How the Receive Adapter Uses this Same Technology

The receive adapter responds to events.

When it receives a message from an external source, it dumps that data into a generic table, and calls a database package. Because the database package is completely data-driven, any process flow at all inside CMiC Software can be invoked from the received message. The specific updated application decides exactly what flow should be invoked, codes the necessary processes as pl/sql procedures and functions, and supplies the process flow itself as data in the process flow tables.

This completely de-couples the receiving and processing of messages on the middle tier (accomplished by the receive adapter) from the process flow that they invoke in the CMiC database (programmed by the application specialists).

Because the process flow is completely flexible the application specialists can create whatever processes are necessary, using whatever application processing and validation logic is required.

The receive adapter follows its generic processing rules to translate data into the response that it returns to the message originator in the external system.

Internal Plumbing – Transmit Adapter

The transmit adapter works differently because the messages it transmits originate within CMiC Software and based upon events that occur in that system.

The transmit adapter listens on an Advanced Queuing (AQ) pipe that is set up in the CMiC database. Any program (or database trigger, etc.) that wants to transmit messages to an external system must put a message in the AQ pipe.

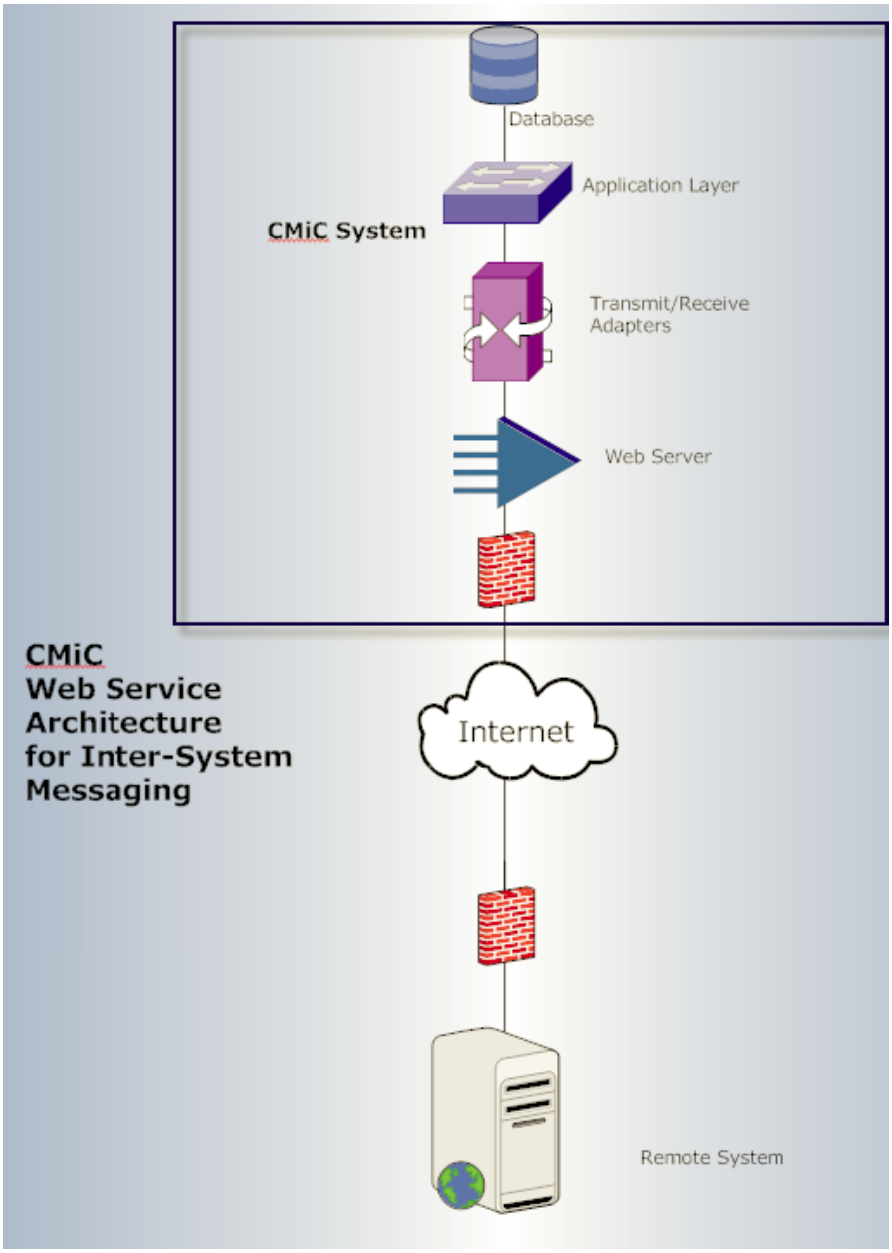
The message payload in the pipe is represented as an XML Document. The transmit adapter receives the XML Document through the AQ pipe, transforms it into the format required for the external system, and transmits it according to the protocol and messaging style required by the external system.

Internally, the CMiC application that needs to transmit a message uses the same data-driven process flow functionality that it uses when receiving messages, but this time in an outbound manner. It calls a standard database package which processes the data-driven process flow.

The process flow that it invokes may be very simple (“just send this piece of data to an external system”) or it may be complex (“do some validations”, “if they pass get some approvals”, “when they are received send one or more messages to one or more external systems”).

The point of this is that the process flow itself is defined in data by the specific application. At any point in the process flow the application can invoke the transmit adapter to send a message to an external system by inserting the message contents in XML form to the queue where the adapter is listening.

Part 3 – The Database Layer



The content and purpose of each message is irrelevant to the higher layers (Adapter and Workflow Engine). Once an adapter has been built, and connected to the Workflow Engine, application programmers can use it as a channel to transmit any appropriate application messages from system to system.

The CMiC Software sits inside the black rectangle.

The Application Layer receives message events from a receive adapter and processes the message according to the rules of the application, possibly handing off a new message to the transmit adapter as a result of the processing.

The Application Layer can also hand off new messages to the transmit adapter as a result of application-specific event triggers (such as the creation of a new object in the database).

The transmit adapters encode message payloads into the appropriate formats depending on the communication protocol and style that is expected by the remote system.

The receive adapters decode message payloads from the CMiC published message formats based on the communication protocol and style of the receive adapter.

The Basics of Data Transmission: SELECT, INSERT, UPDATE, & DELETE

The Http-Receive Adapter for CMiC Software supports 4 basic operations: select, insert, update, and delete.

Data objects that are exposed by the adapter for transmission are called “resources”. Two types of resources can be accessed: collections and individual objects.

Collections are accessed via “<object_plural_name>_list”.

Individual objects are accessed via “<object_name>”.

Requests are sent to http://<server>:<port>/<cmic_environment>/wsrti/v1. The path /wsrti is the root entry into the adapters.

The version number is provided to allow us to support external systems that transmit messages to us using different versions of our adapter interface.

For example, two CMiC customers could be using RTI adapters to communicate messages between their systems. Initially they both use our v1 adapters. Then one of them upgrades to a new version of our software which includes v1 and v2 adapters. The adapters in the new release must be able to understand messages from the external system which still only knows how to transmit v1 messages.

For example, if we provided two data objects called “departments” and “employees”, the following illustrates how any external system would perform select, insert, update, and delete operations through the http receive adapter running on a local system.

URI’s and URL’s

This section of the document uses the term URL and URI. Here we will provide a brief description of these two terms.

A URL is the request string that you type into a browser address bar. For example a URL might be http://www.oracle.com.

A URL is a Universal Resource Locator.

The format for a URL is: <protocol>://<server>:<port>/<URI>

The beginning part of the URL identifies what protocol is being used, and where the resource that is being requested is located.

Everything after the <protocol>://<server>:<port>/ is the URI portion.

A URI is a Universal Resource Indicator. It is the portion of the URL that uniquely identifies a specific resource at the location <server>:<port>, using protocol <protocol>.

The CMiC RTI <protocol> is always http.

Most of the work done by the receive adapter is to parse out and understand the URI portion of each URL in a request.

Similarly, the work done by the transmit adapter is to form a valid URI for the URL request that it will transmit.

Key Values

CMiC will provide a published API for each resource identifying the key value or values that must be provided in order to uniquely identify each specific resource.

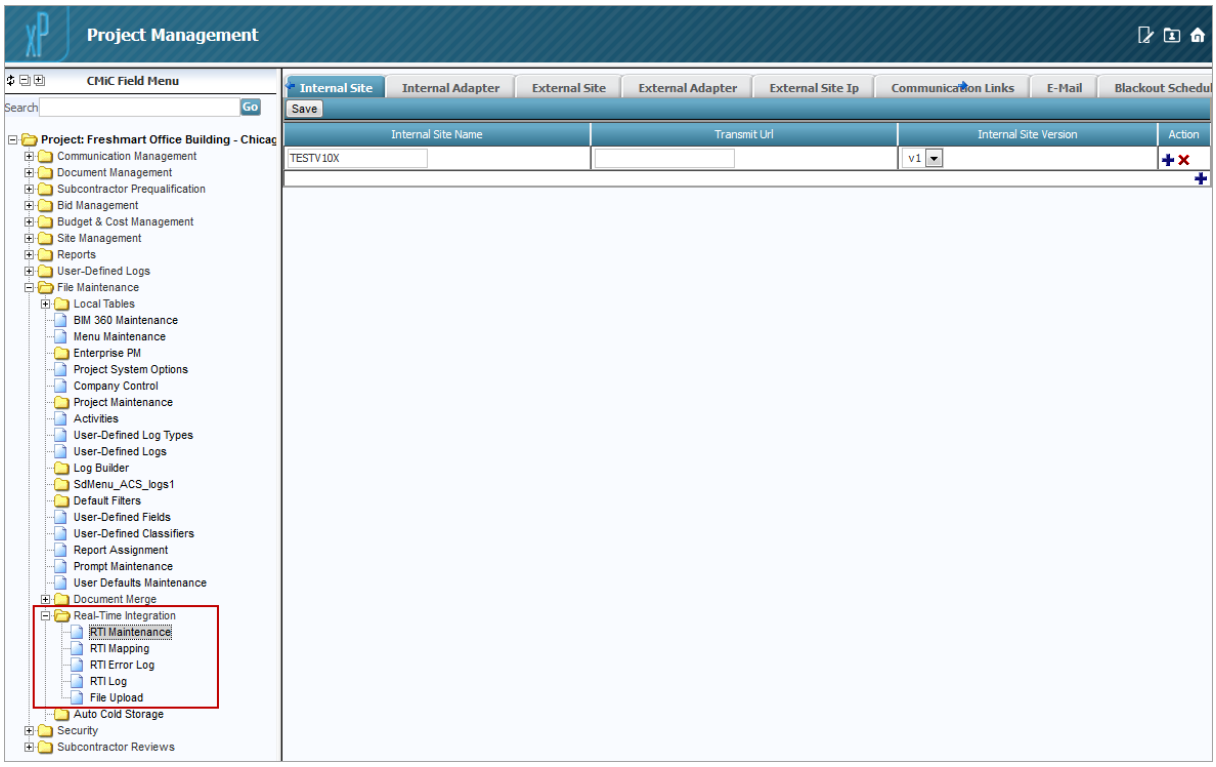
There is one set of key fields that are provided for every resource (i.e. one path to directly address every specific resource in the collection). In some cases the key consists of an individual field. In other cases the key is formed from a combination of fields.

Each request that requires key values must specify a value for every key field. Partial key requests are not permitted.

Key values are required for every UPDATE and DELETE request. UPDATE and DELETE requests always operate on specific resources. SELECT requests can select a specific resource or a collection resource.

Configuration of CMiC RTI

Overview – Configuring RTI for CMiC to CMiC



This section details the configuration steps involved in setting up your CMiC System to utilize RTI Transmissions to and from another CMiC System installation.

Requirements for CMiC RTI Functionality

In order to utilize CMiC RTI Functionality, the appropriate RTI and Collaboration User Licenses must be installed on the system. The RTI Functionality is designed for maintenance and identification within the PM JSP screens only.

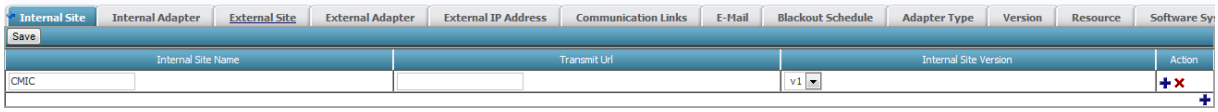
In order to setup RTI for functional use with another CMiC Customer, both users must have same Version installed (Adapter Versions – not Patch Versions – as users can have multiple versions of adapters over time)

Additionally, the two users of the system must co-ordinate the setup maintenance to ensure the correct settings are made in each environment.

RTI Maintenance Setup is general (not specific to Project) and must be completed prior to configuration of the Project to Project mapping that each party must complete within the 'linked' Project through the RTI Mapping screens.

RTI Maintenance – CMiC to CMiC Adapters

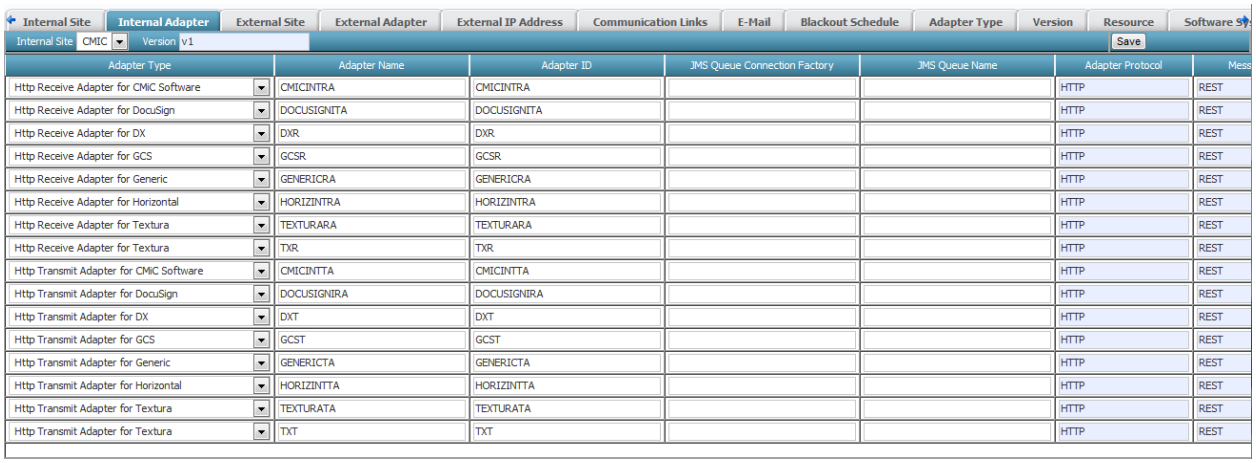
Internal Site



The Internal Site configuration is the setup of a name for use for your environment. This will be shared with the other RTI users, which they will be entering as their External Site information.

Note, the **Transmit Url** field is not currently used.

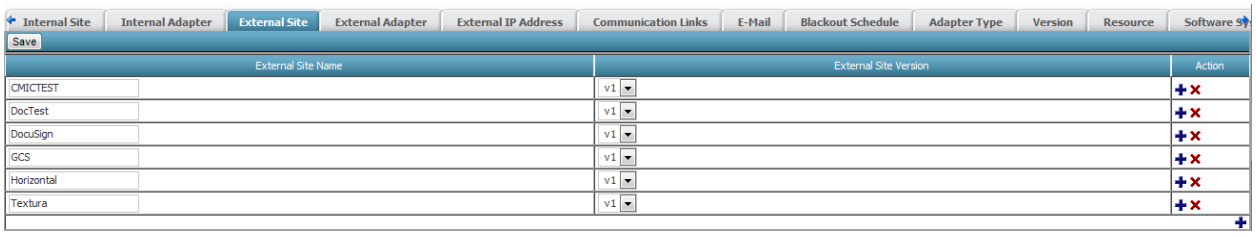
Internal Adapter



The Internal Adapter configuration is the setup of names and ID references for your adapters within your environment. This will be shared with the other RTI Users which they will be entering as their External Adapter information.

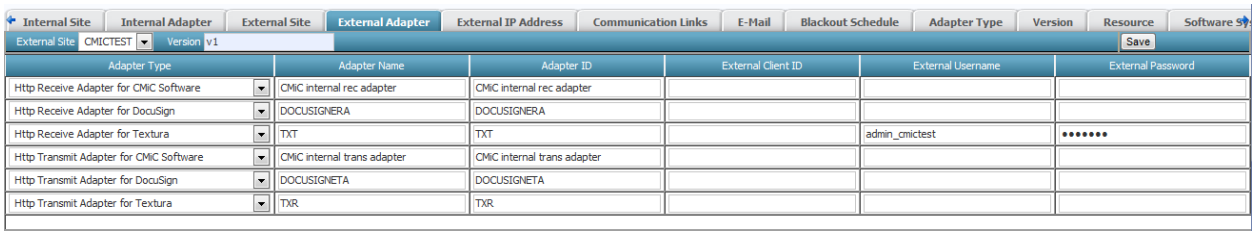
This setup is also used to show the System/Type and Adapter protocols for the Adapters you have selected to be used with your site.

External Site



The External Site is the reference to the other user system with which you will be utilizing the RTI Transmissions. There will be a separate External Site for each 'other' Environment you may be connecting with.

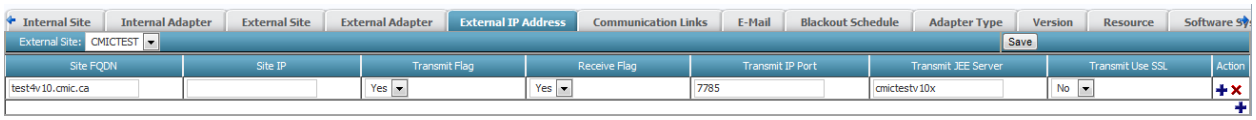
External Adapter



The External Adapters is similar to the Internal Adapters; however the values entered here are based on the other user site Internal Adapter configuration. (What you create as Internal is their External and vice-versa)

A drop list selection of External Sites is provided. Ensure the correct Site is selected prior to entering the Adapter information you have received from them.

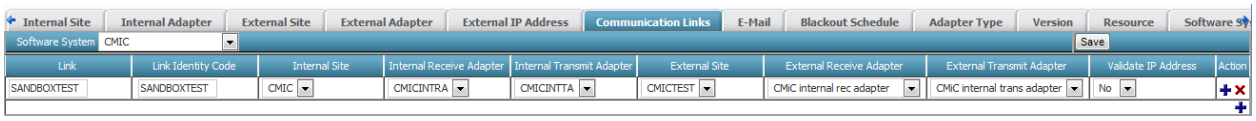
External IP Address



External Site IP Configuration is the mapping for the internet travel of RTI data to the other site. Use of either the FQDN Name or the Site IP (recommended) is used, and then specification of which sites Transmit/Receive or both, as well as the Port to use, J2EE Server name and whether the communication is Secure (HTTPS) or non-secured (HTTP).

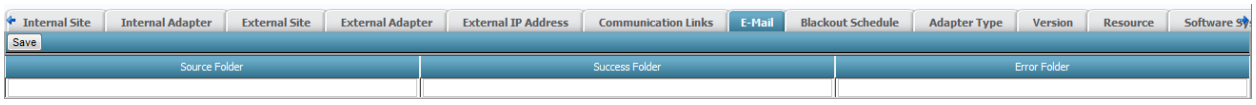
NOTE: When specifying a server's URL, do not include the "**http://**" or "**https://**" portion of the URL, as that is done in the backend and will cause errors if specified.

Communication Links



The Communication Link is the final stage in the setup where you will be setting up the Link Code and Names, and specifying which Internal Site/Adapters will be communicating with the selected External Site/Adapters.

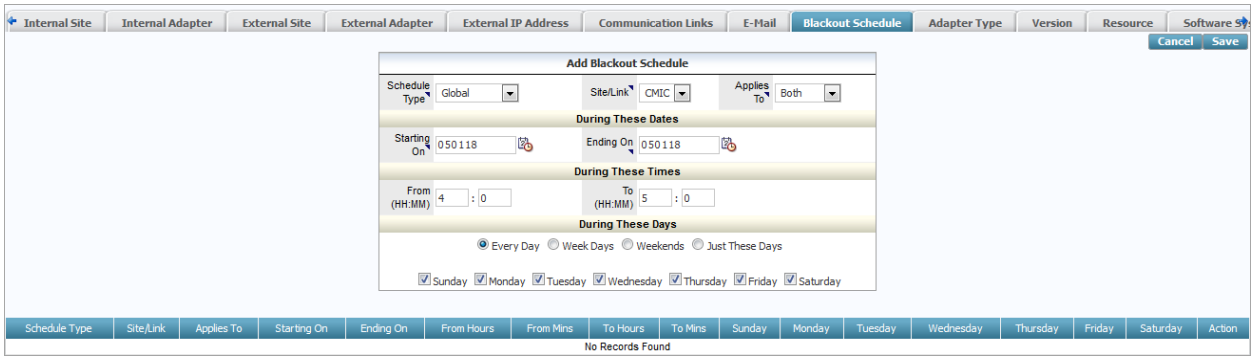
E-Mail



The section is for setup of Email Folders as required for the relevant RTI Adaptor.

Note, this tab is not used for CMiC to CMiC RTI Transmissions.

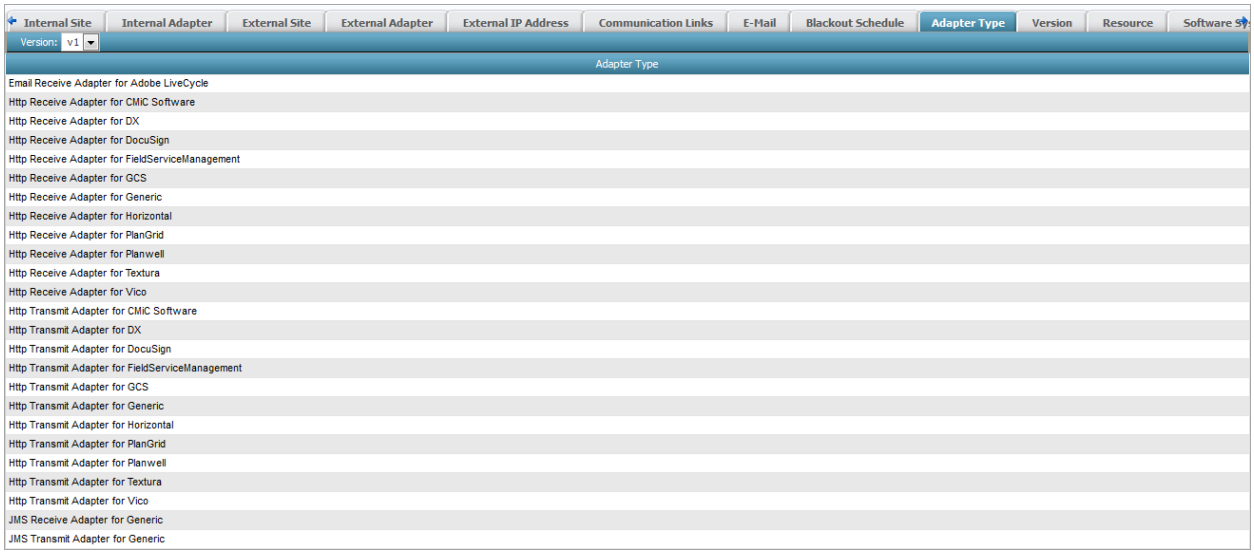
Blackout Schedule



The Blackout Schedule allows configuration of settings to prevent communication attempts when the environments will not be able to receive. For example, schedule of Blackout should be entered for both Sending and Receiving sites based on scheduled maintenance of Servers, Patch installation windows, or any other event that may prohibit success in transmission due to environments or databases being unavailable.

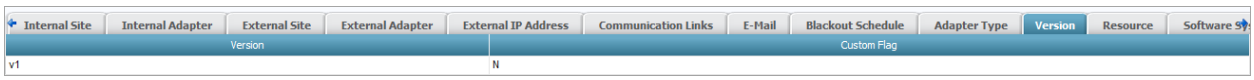
Multiple schedules can be entered as required.

Adapter Type



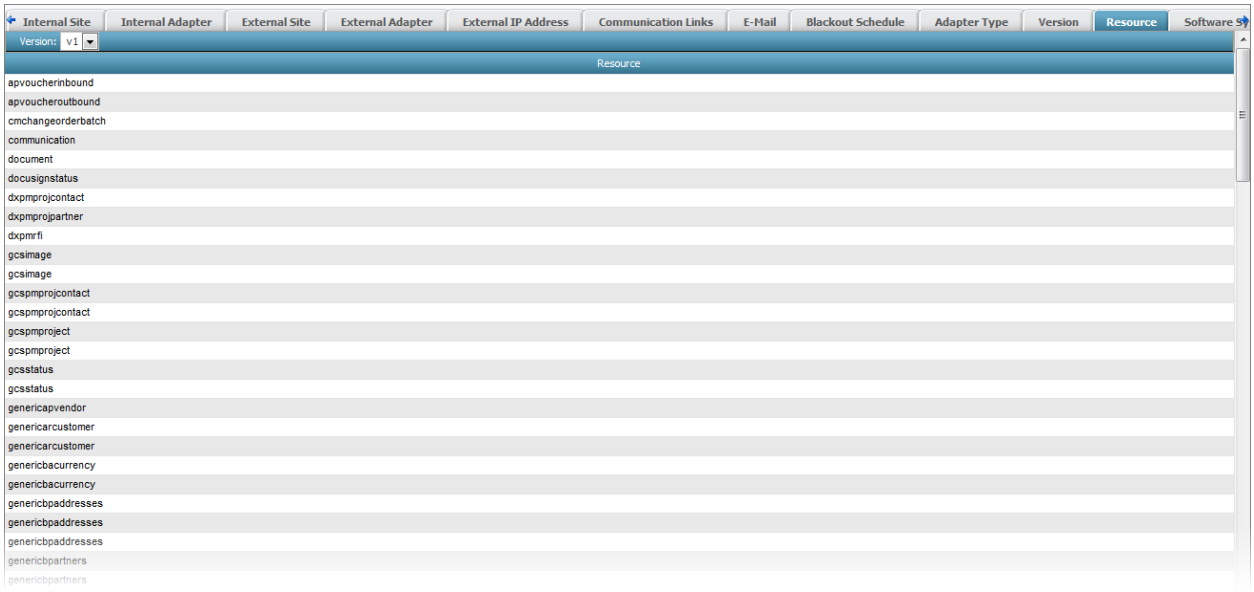
This display-only tab shows which adapters are available for the version selected using the **Version** drop-down list.

Version



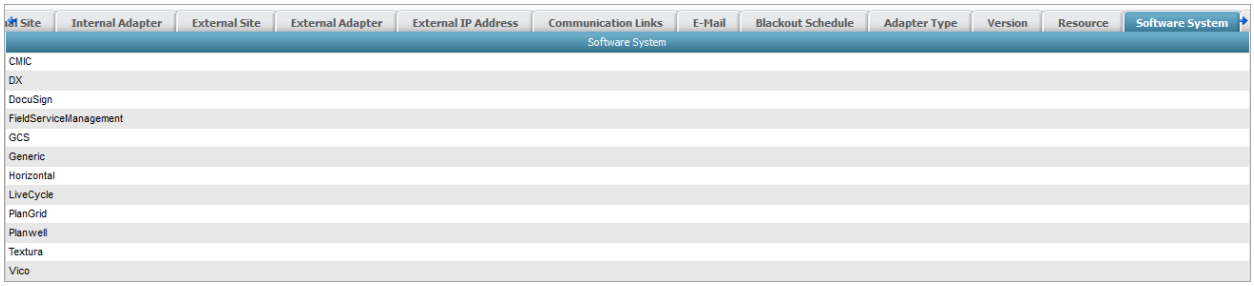
This display-only tab shows the versions available.

Resource



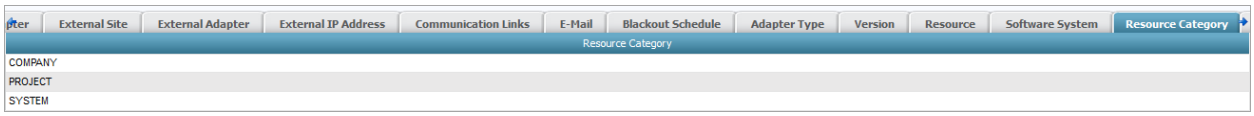
This display-only tab shows the types of objects that may be transmitted using the version selected via the **Version** drop-down list.

Software System



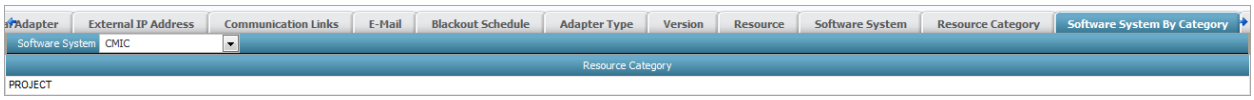
This display-only tab shows the Software Systems available.

Resource Category



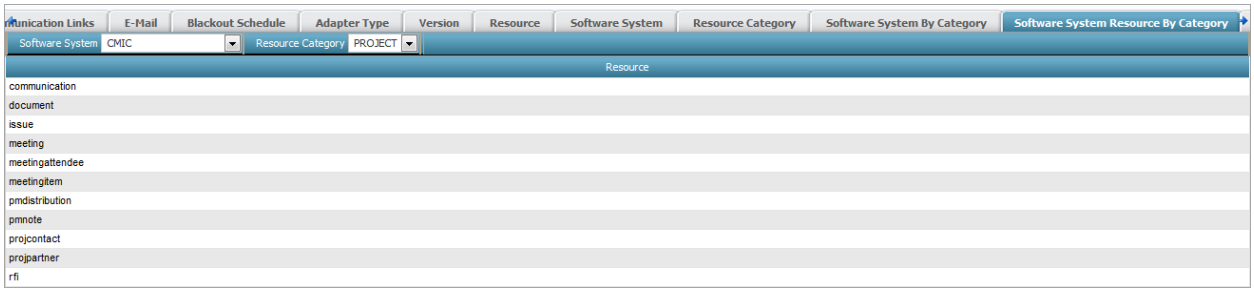
This display-only tab lists Resource Categories in use by the various systems.

Software System By Category



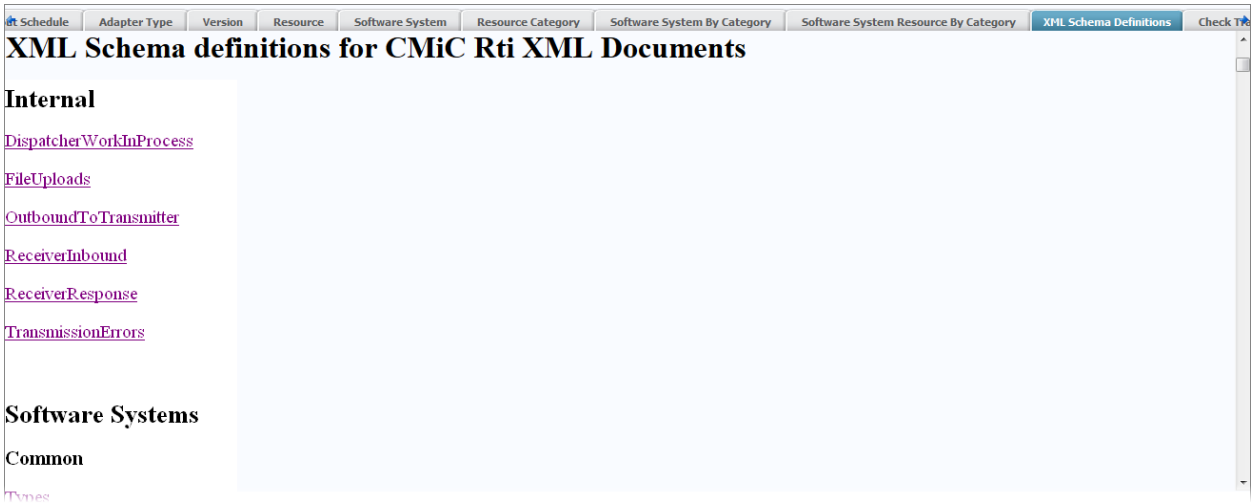
For the system selected via the **Software System** field, this display-only tab lists its related Resource Categories.

Software System Resource By Category

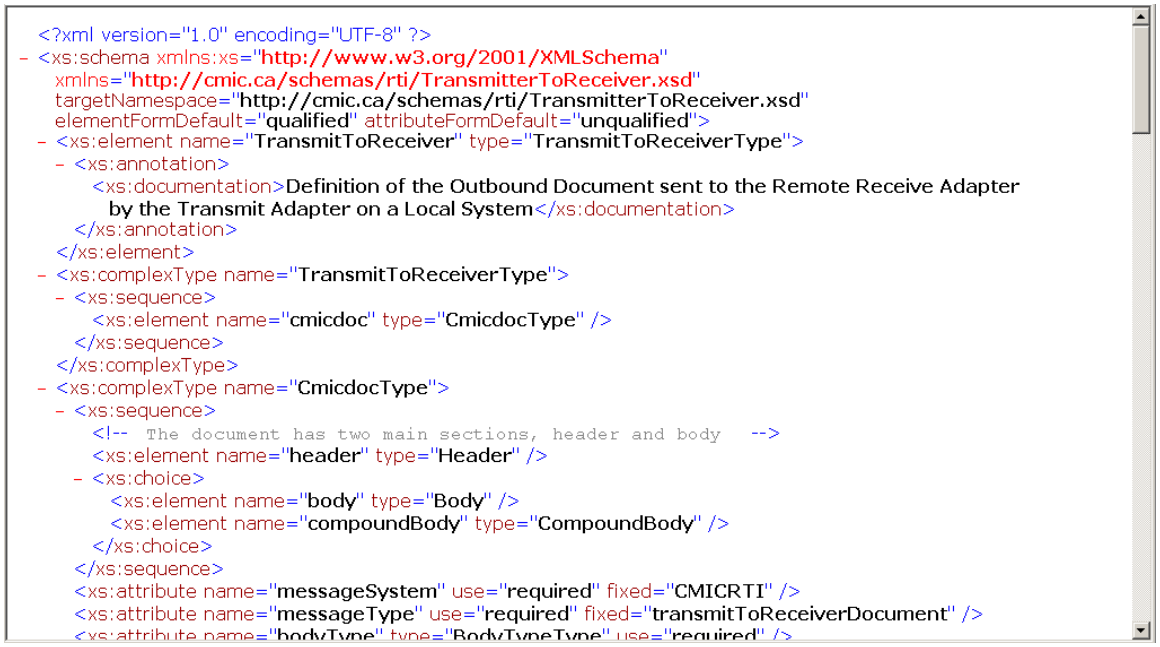


Display only-listing of the System Resources in use by the various Resource Categories for each of the Software System types.

XML Schema Definitions

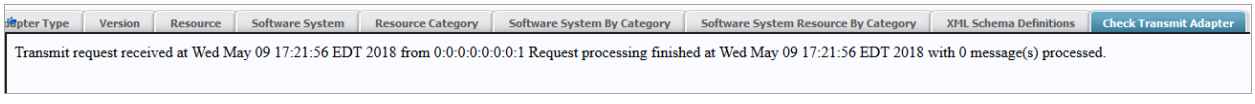


The tab is used to see the definitions being used in the XML interchange of the various items.



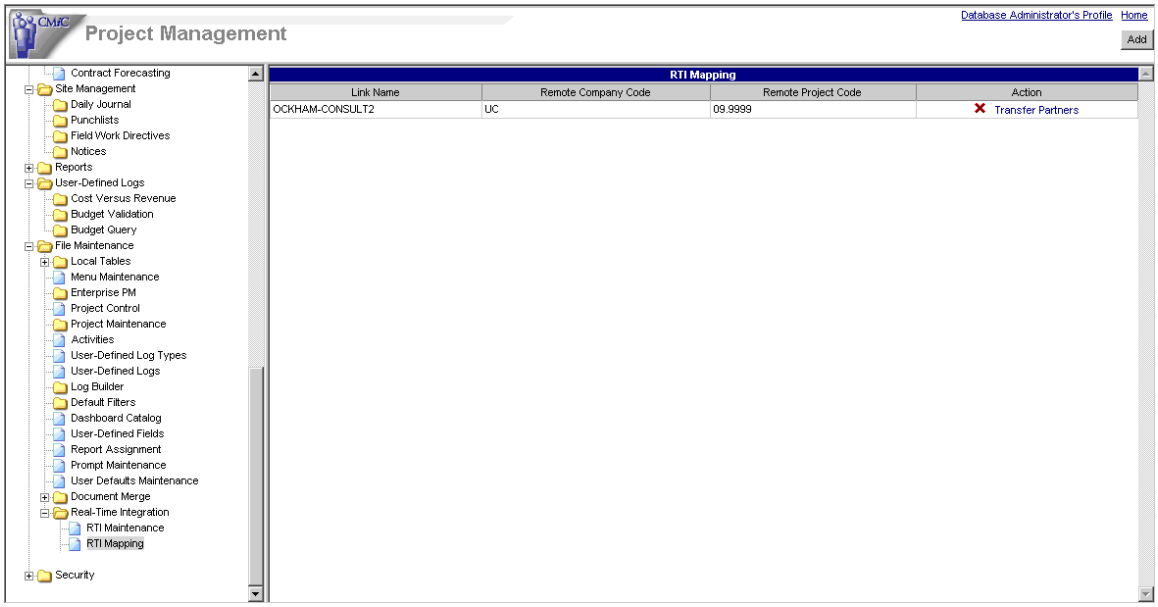
Example of XML Definition for CMiC Transmit To Receiver

Check Transmit Adapter



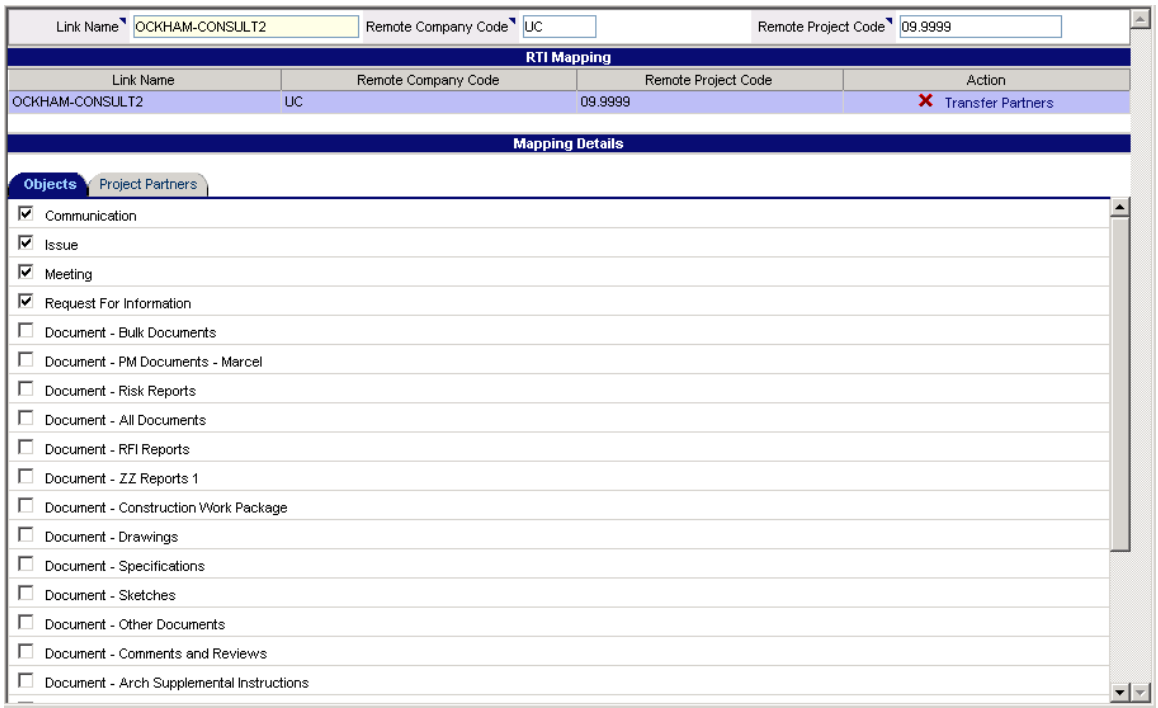
When selected, this tab triggers the adapters to process anything in queue. If the adapters on your site are working, a message will appear about the processing. If there is an issue with adapter, the appropriate error message will be shown.

RTI Mapping for CMiC to CMiC Usage

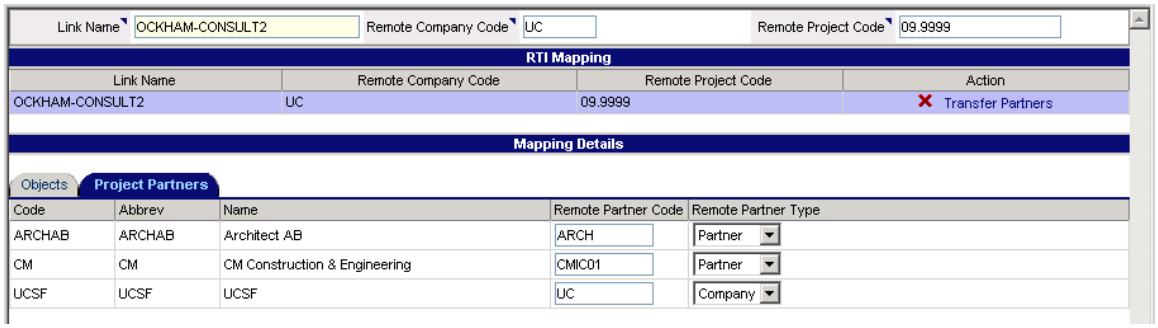


RTI Mapping is what allows the adapters to understand where to insert data received or transmitted between the various systems. The RTI Mapping screen is defined to the Project you are displaying in PM JSP when opened. This maintenance is required for any Project that will be using the RTI System to connect and update another user Database/Project.

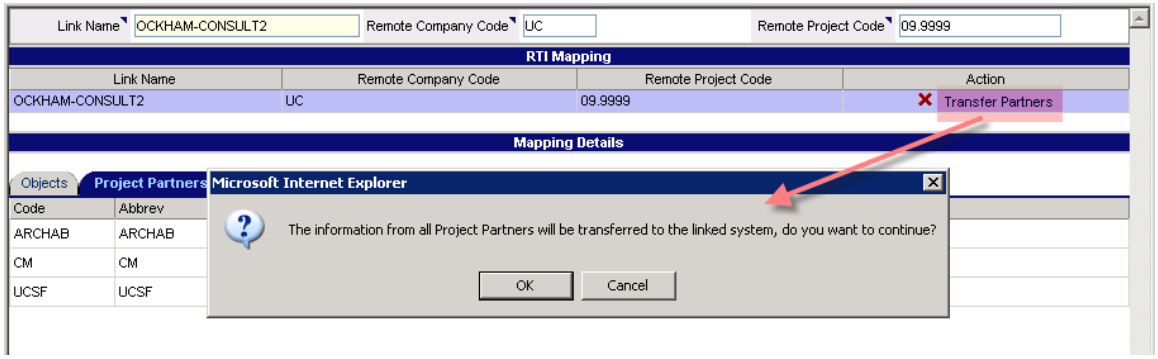
For example, your Project XYZ is in your database within your Company 10. The corresponding Project at the other user's database is Project ABC in their Company 01. Mapping setup allows the transmit and receive adapters to understand that an RFI entered in your Company 10 Project XYZ would need to be inserted in the other database in Company 01, Project ABC.



By clicking on the Project Link from the initial display, the user will see 2 tabs for configuration of what Objects may be transferred, and also the Mapping of Project Partners to enable matching your Contacts on your Partners with their Partners and Contacts.

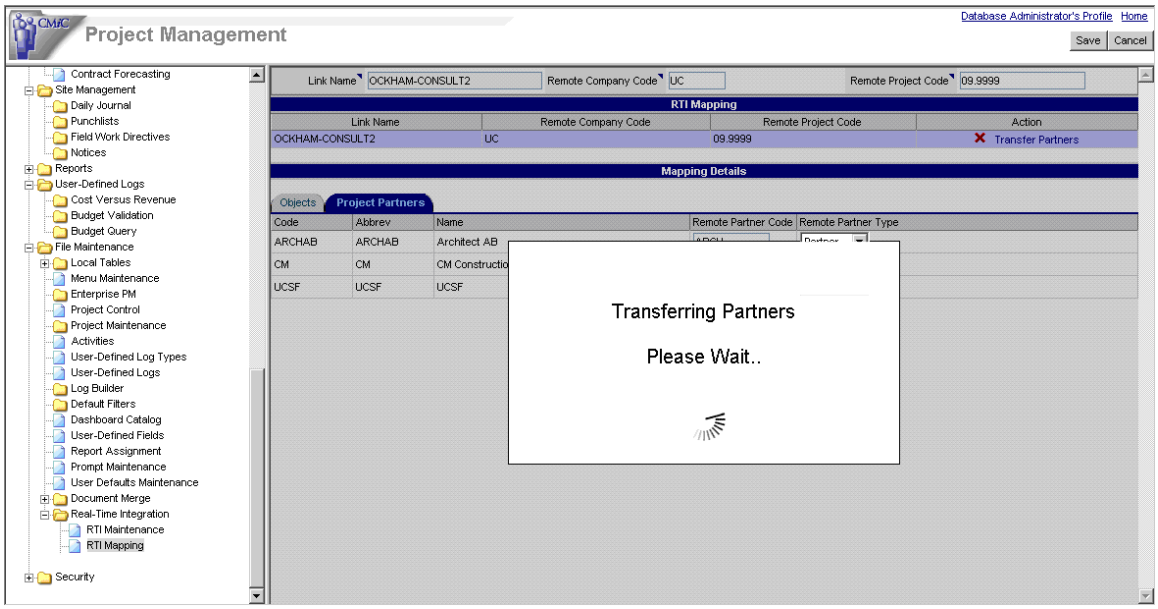


The Project Partners in your database will unlikely have the same Partner Codes in the other user Database. To ensure connections are made and the correct contacts setup on the other side, the Project Partner Mapping shows all Project Partners in your Project and allows you to enter the Partner/Company Codes relevant for the other database and to identify if it is a Partner, or a Company record on their database.



There is also an option to bulk transfer the Partner information to the other system by selecting the [**Transfer Partners**] option under the Action for the Project link.

When using this option, the following will appear while the transfer is being made:



Partners are mapped by link if entered, and if not, will be mapped by first searching for the matched Phone Number data, and secondly for a match on the Partner Codes.

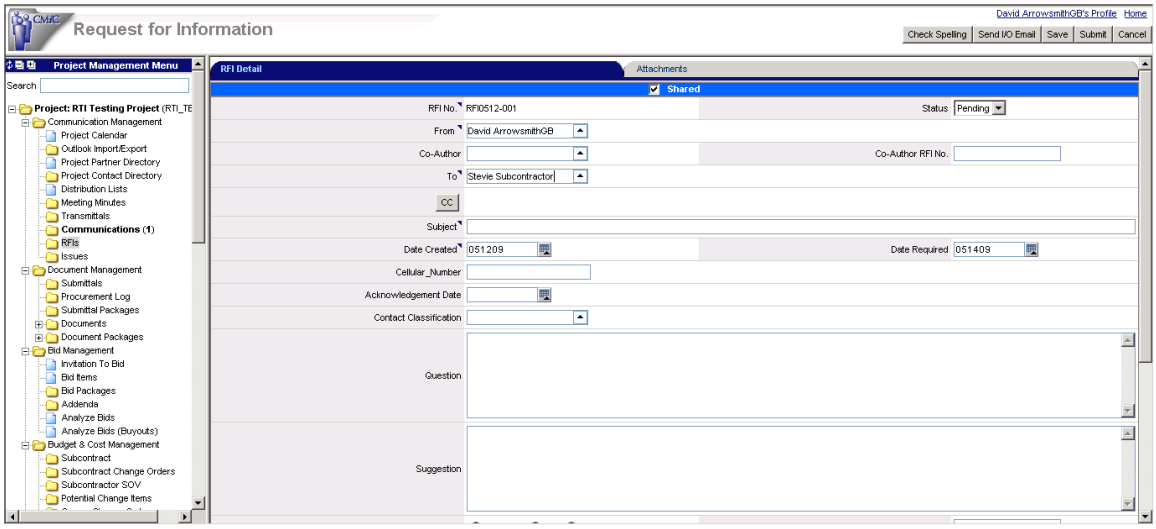
Similarly, the Contacts for that Business Partner will be mapped by searching for matches in the Email field, and if not found, a second search on Contact Code match will be made.

The process of Mapping Partners should be run from both the local and remote databases.

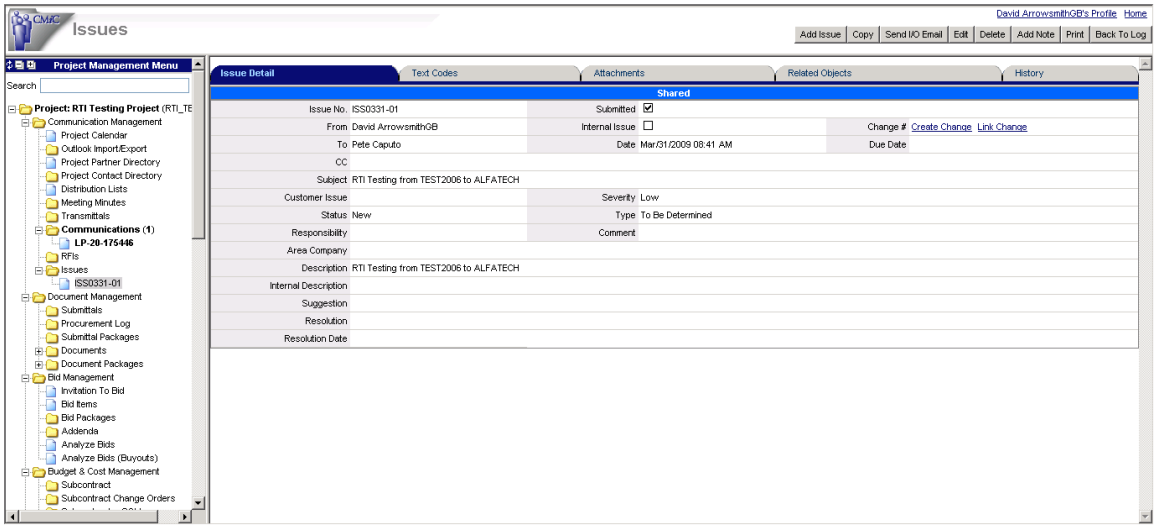
CMiC to CMiC RTI Transmissions

Overview – Using RTI (CMiC to CMiC)

The basic CMiC to CMiC RTI Function is provided to allow specific objects such as RFI, Communications, Meeting Minutes and others to be entered in one database and through RTI appear in the other users' database. By default, if the project is mapped for RTI functionality, there will appear a 'Shared' flag in items that can be transmitted. By unchecking this flag, the item when saved or submitted for the first time will remain only within your database. If it was checked to be shared, on your submitting the record, it will be transmitted to the other databases as configured.



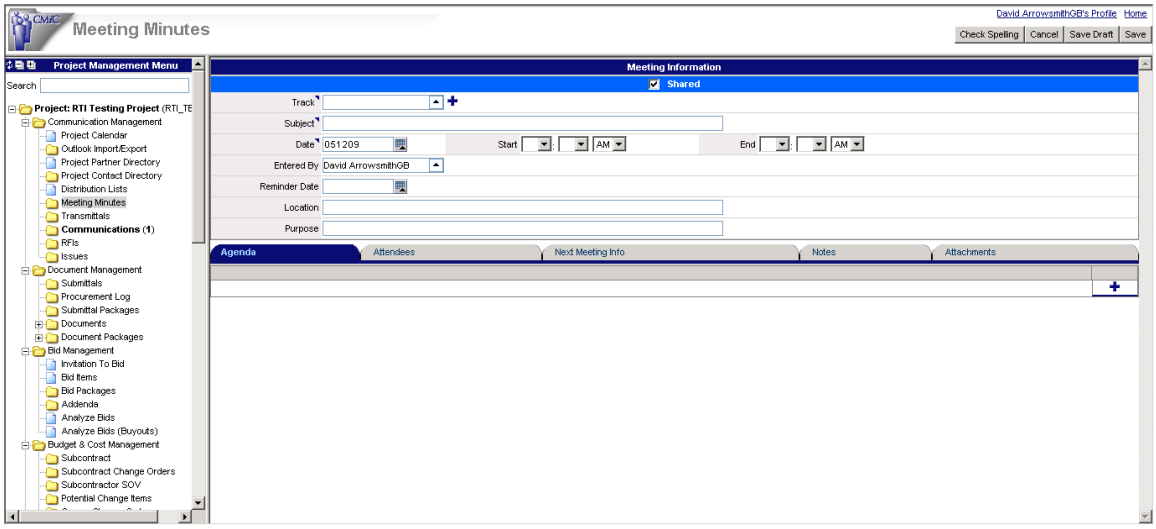
Sample RFI Record – New RFI Create with the RTI Mapping setup for the Project.



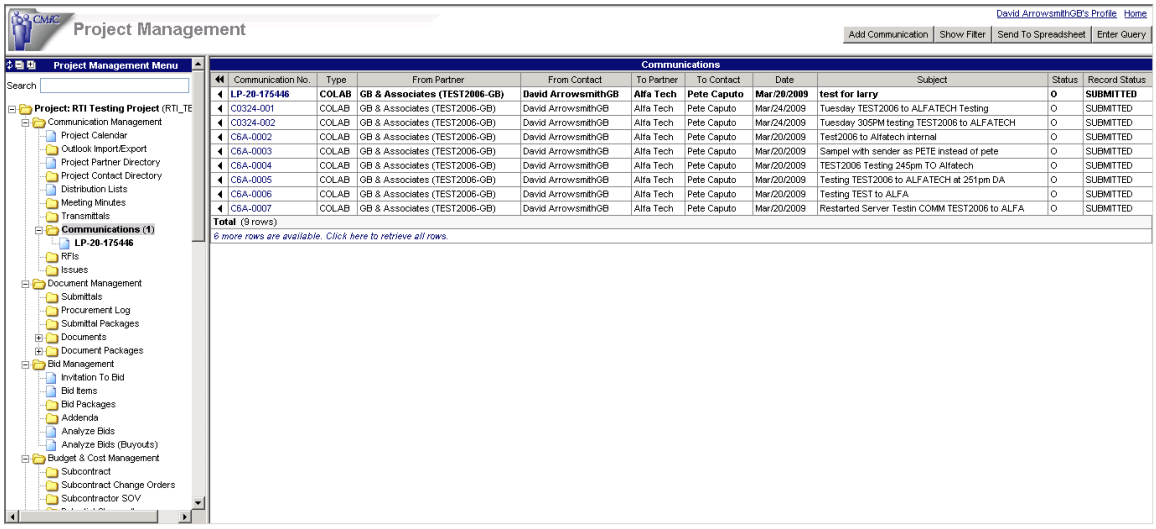
Submitted Issue where Shared Option for RTI Transmission was Checked

Once an RTI Item has been saved/submitted, there is no changing the Shared Option. Ensure the correct value was set prior to submitting any Object when working in an RTI Enabled Project. By default, the option is always checked for sharing.

If the user in one of the Projects then proceeds to update any data in the item, it will appear as a Note in the matched record on the other database (non-editable and shaded).

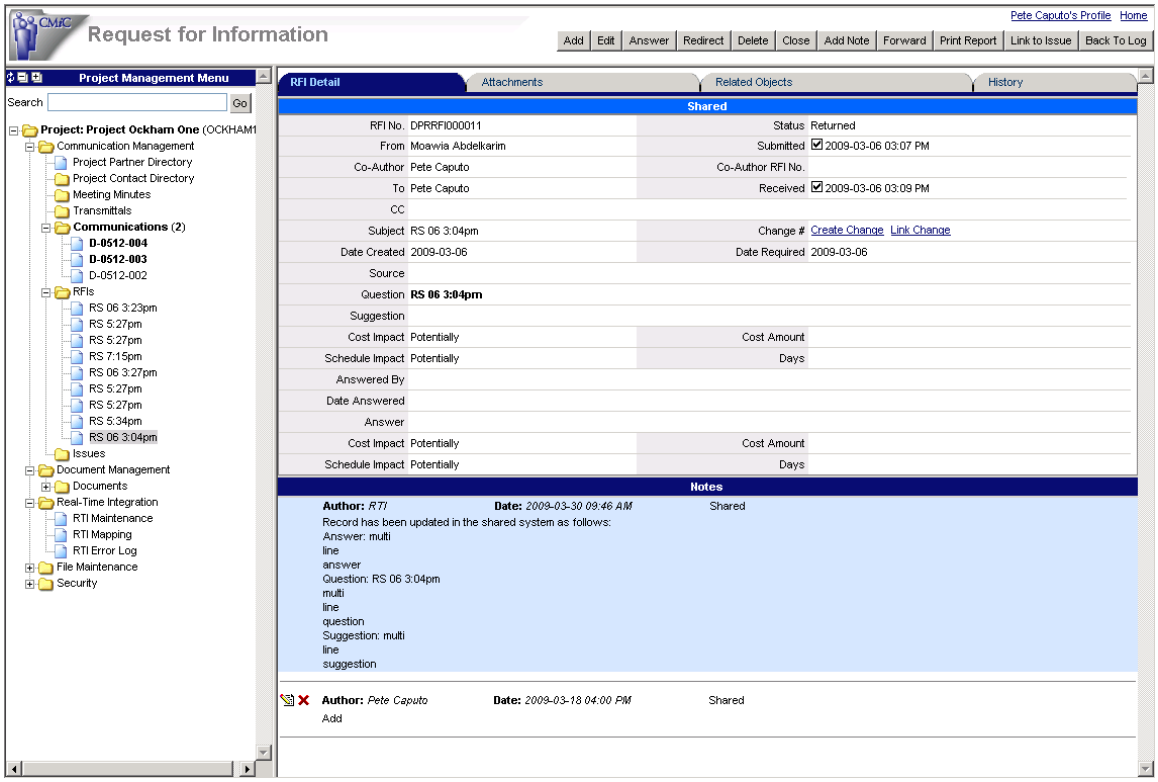


Meeting Minutes are another RTI Enabled item, and can be shared manually by saving just the header and then later transmitting additional data by later updates, or by entering the entire Meeting and details and submitting at once.



When reviewing the Project Logs, items created from the other database will likely have a different ID Code, which makes for easy identification of the records originating from the other database. Additionally, the From/To Partner codes will assist in noting which database created the original records.

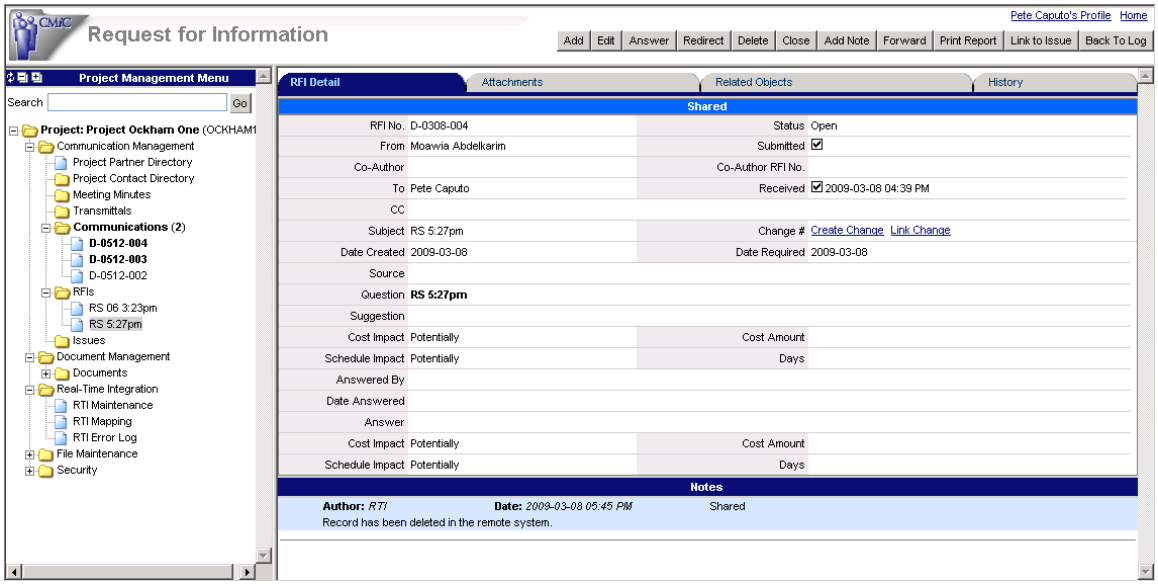
Updated Record



This sample shows an RFI that has been updated in both databases. In the current database, the user Pete added a Note. In the remote database, additional fields were populated such as the Answer. Note that the answer does not appear in the RFI, but only in the Note section.

The blue shaded, ‘display only’ notes are transmitted from the remote database when a matched object is updated in any way. The note will identify the changes made including identification of which fields were changed. It is then up to the user of the receiving database to copy/paste whichever data they wish to have stored in their copy of the record. The ‘remote update’ note is also not available for deletion to ensure audit ability.

Deleted Record



This sample RFI shows in the database it was transmitted to, and in the ‘Notes’ shows that the record was deleted in the Remote Database.

When a record is deleted in one database, it is not deleted in the other database, but rather a note identifying when it was deleted and that the record was deleted will show in your database.

Once any object is transferred, any changes, including deletion is shown as a note. There is no direct delete option from your database that would occur in the other database.

This logic is specific to CMiC PM and may be different for other RTI Integrations.

Important Conditions

In order for data to be transmitted from one database to another, specific care must be given to things that may be unique in your database. Types, Status Codes, etc, must exist in both databases for an object to be transferred.

For example, when transmitting a Communication Record, there is a type of communication that is user definable. If the type is not found in the remote database, the record transmission will fail. Correction will be to adjust to a value that both databases contain.

Documents and Attachments

The transmission of objects that have attached PM Documents is permitted. Only the Document types that are included in the RTI Mapping will be included allowing control of what documents may or may not be transmitted.

In order for Documents to transfer however, the matching Document Type and Status codes must exist in each database. If there is already a document in the remote database that has the same Type/Document ID, then an error will be presented in the transfer as overwriting/replacing of the document on the remote database is not permitted.

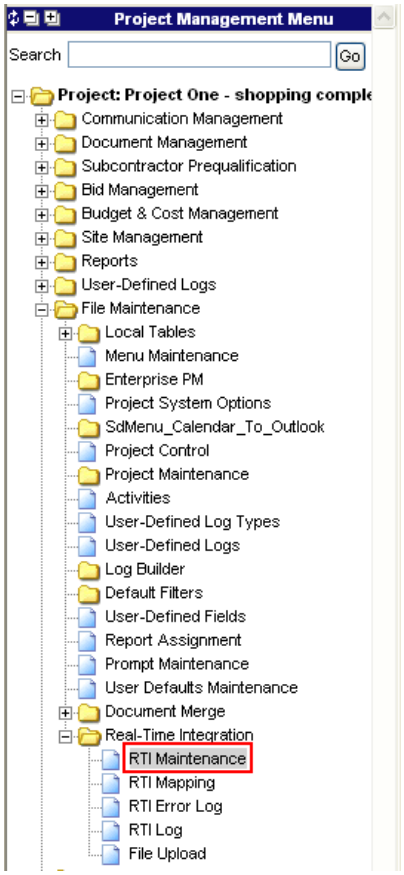
CMiC RTI & Horizontal Glue

Overview

This section describes the steps necessary to enable the Horizontal Glue RTI adapter in CMiC for those with the appropriate License for this functionality.

RTI Maintenance

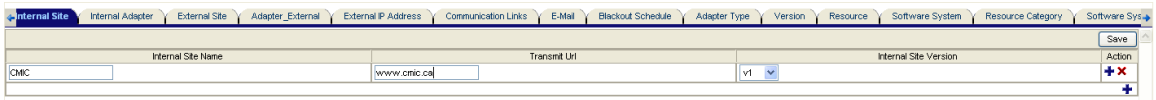
Login to PM and select the RTI Maintenance menu option.



NOTE: In this screen you are doing a global RTI setup that is NOT specific to a particular project.

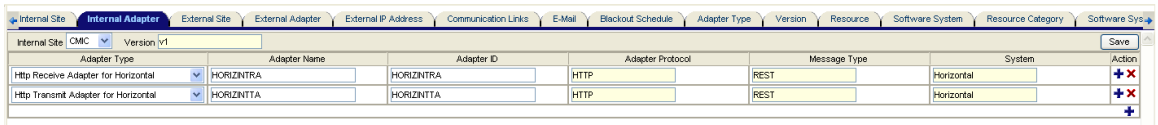
Internal Site

In the Internal Site tab, create an internal site to represent your CMiC software.



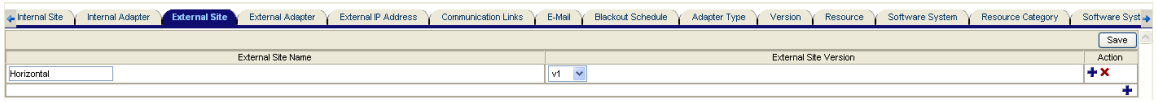
Internal Adapter

In the Internal Adapter tab, create internal adapters for receiving from and transmitting to Horizontal. Make sure the Internal Site you created for CMiC is selected in the dropdown list. You can set the Adapter Name and Adapter ID as in the example below or to any value you wish, these values will be used by Horizontal Glue to communicate with your CMiC software.



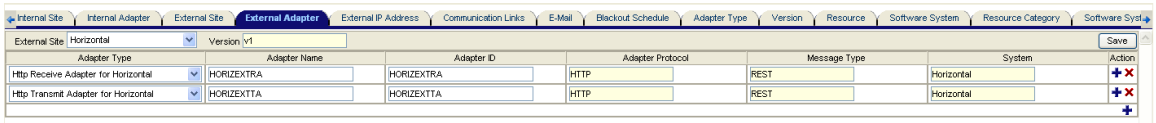
External Site

In the External Site tab, create an external site to represent Horizontal Glue.



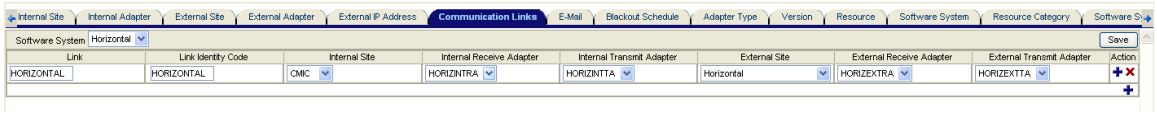
External Adapter

In the External Adapter tab, create external adapters for receiving from and transmitting to Horizontal. Make sure the External Site you created for Horizontal is selected in the dropdown list. You can set the Adapter Name and Adapter ID as in the example below or to any value you wish, these values will be used by Horizontal Glue to communicate with your CMiC software.

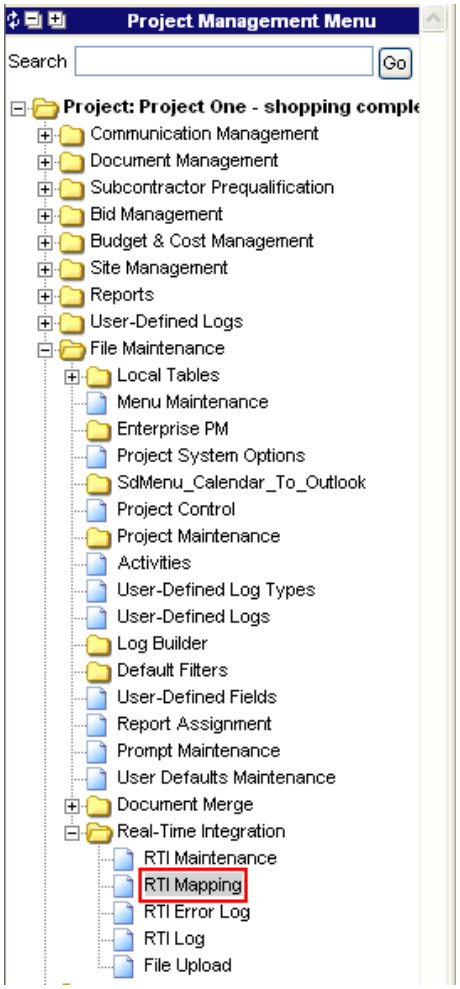


Communication Links

In the Communication Links tab, select Horizontal from the Software System dropdown list. You can set the Link and Link Identity Code as in the example below or to any value you wish, these values will be used by Horizontal Glue to communicate with your CMiC software. Fill in all fields using the values you created in the previous tabs. Be sure to select the proper receive and transmit adapters from the dropdown lists.



RTI Mapping



Select the RTI Mapping menu option.

NOTE: In this screen you are doing project-specific RTI setup, linking the project you are currently logged into in PM to the Communication Link you created in the RTI Maintenance Communication Link tab. You will need to do this step for each project that will be communicating with Horizontal Glue.

Press the [**Add**] button to create a new RTI Mapping.

Select Horizontal from the Software System dropdown list. Select the Communication Link you created in RTI Maintenance from the Link Name dropdown list. Save the record.

RTI is now setup to communicate with Horizontal Glue.

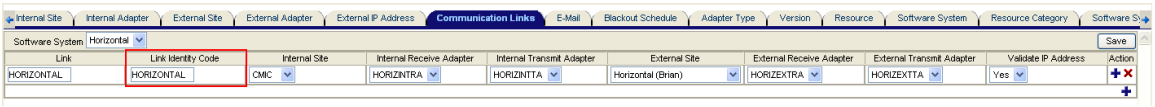
Values Required by Horizontal

You must provide Horizontal with the following values to enable them to communicate via RTI with your system.

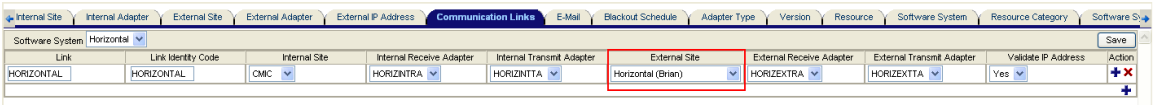
URL

Open PM (JSP). Horizontal can use this URL to derive the URL needed to communicate with RTI.

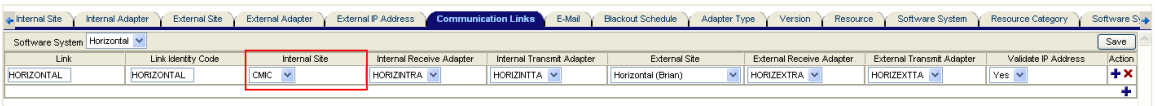
Link Identity Code



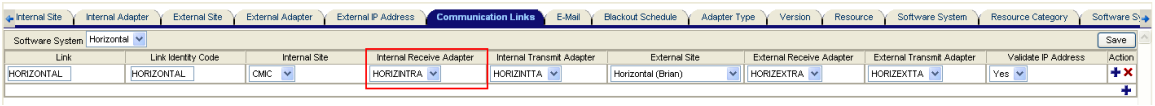
From Site Name



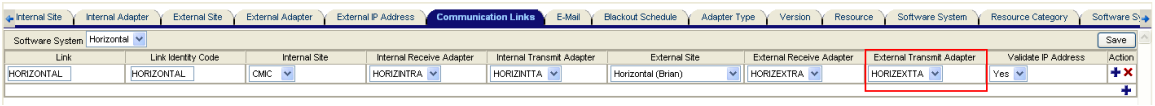
To Site Name



Receive Adapter



Transmit Adapter



CMiC RTI & Textura

Overview – CMiC RTI & Textura

Textura and CMiC have collaborated to integrate CMiC Enterprise's Project Management, Subcontract Management and Accounts Payable modules with their web-based Construction Payment Management ™ (CPM ™) solution to improve the operational efficiency of invoicing, compliance management, lien waiver collection and payments related to subcontracts and change orders. This is done by enabling collaboration between a CMiC client, Textura and subcontractors.

For a more in-depth overview about this integration, please refer to the following webpage: *Textura and CMiC Deploy Real Time Integration Tool to Bring Greater Efficiency to Textura/CMiC Clients*.

Also, for further details about Textura's Construction Payment Management ™ solution, please refer to their following resources:

1. www.texturacorp.com/construction-software/payment-management/
2. http://www.texturacorp.com/texturacorp/assets/File/solution-guides/Construction%20Payment%20Management.pdf
3. https://www.youtube.com/watch?v=iv8l6juORkA

The following sections in this guide describe the steps necessary to setup the Textura adapter in RTI Maintenance, and how to use the RTI Mapping screen to turn Textura on for specific projects.

Payment Management Process Flow – Overview

Part 1: **Subcontract/Change Order Details Sent to Textura**

When a Subcontract or Change Order is posted in CMiC Enterprise, it triggers the sending of the Subcontract/Change Order to Textura to initiate its Construction Payment Management ™ solution.

Part 2: **Textura Creates Online Task Tracker for Subcontract/Change Order**

Textura uses the details of the received Subcontract/Change Order to create an online tracker for the Tasks (associated to Cost Codes) required to complete the Subcontract/Change Order, which is used by subcontractors to report how much of each Task was completed during each payment cycle for payment purposes.

Part 3: **Textura Makes Payment to Subcontractor Based on Online Task Tracker**

At the end of each payment cycle, Textura and CMiC Enterprise clients do the following:

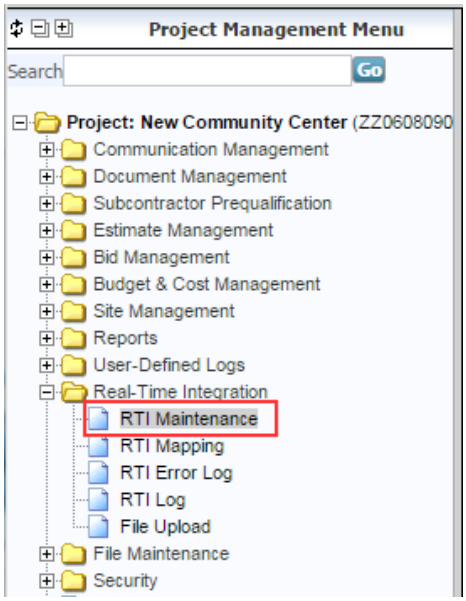
I. Textura determines the payments to the subcontractor based on the work done for each Task since the last payment cycle.

II. Textura sends a Batch of unposted Vouchers (RFPs) for the payments to CMiC client; AP clerks review the Batch and post it if everything is correct.

III. Textura makes a direct deposit payment to the subcontractor based on the work done for each Task since the last payment cycle, and sends CMiC client a Batch of unposted Checks (Payments) for the payment; AP clerks review the Batch and post it if everything is correct.

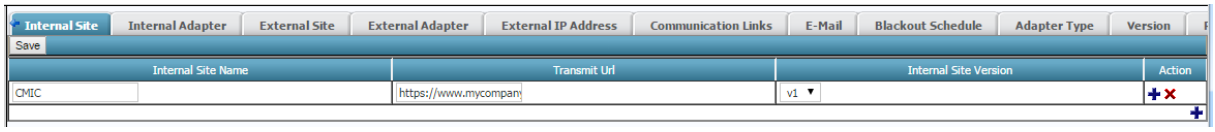
RTI Maintenance

Login to PM and select the RTI Maintenance menu option.



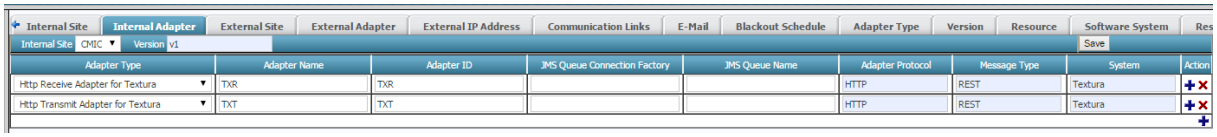
NOTE: In this screen you are doing a global RTI setup that is NOT specific to a particular project.

Internal Site – Tab



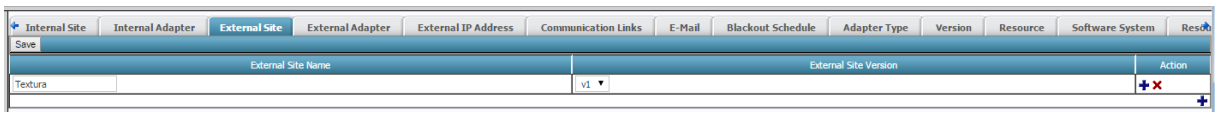
On the Internal Site tab, create an internal site to represent your CMiC software.

Internal Adapter – Tab



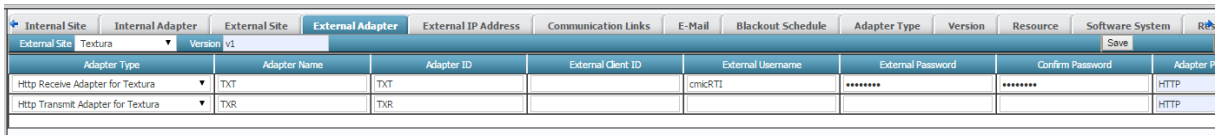
On the **Internal Adapter** tab, create internal adapters for receiving from and transmitting to Textura. Make sure the Internal Site you created for CMiC is selected in the dropdown list. You must set the Adapter Name and Adapter ID as in the example below, using "**TXR**" as the name and ID of the Receive Adapter, and "**TXT**" as the name and ID of the Transmit Adapter. These values are used to control communication between Textura and your CMiC software.

External Site – Tab



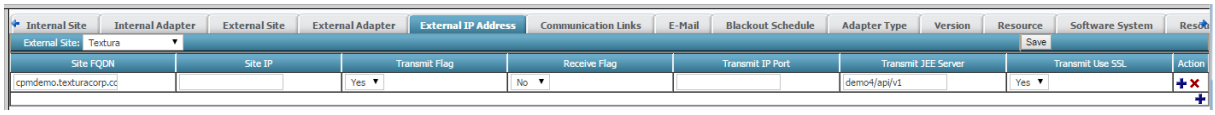
On the **External Site** tab, create an external site to represent Textura.

External Adapter – Tab



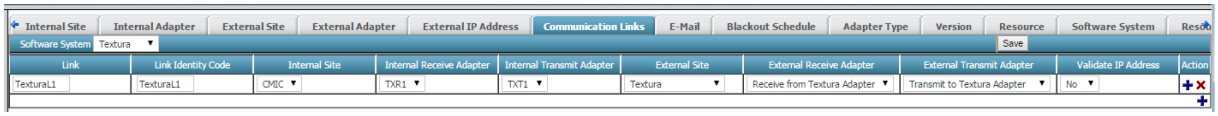
On the **External Adapter** tab, create external adapters for receiving from and transmitting to Textura. Make sure the External Site you created for Textura is selected in the dropdown list. You must set the Adapter Name and Adapter ID as in the example below, using TXT as the name and ID of the Receive Adapter and TXR as the name and ID of the Transmit Adapter. These values are used to control communication between Textura and your CMiC software. In the External Username and External Password fields, enter the values used to login to Textura. This login will be used for ALL transmissions from CMiC to Textura.

External IP Address – Tab



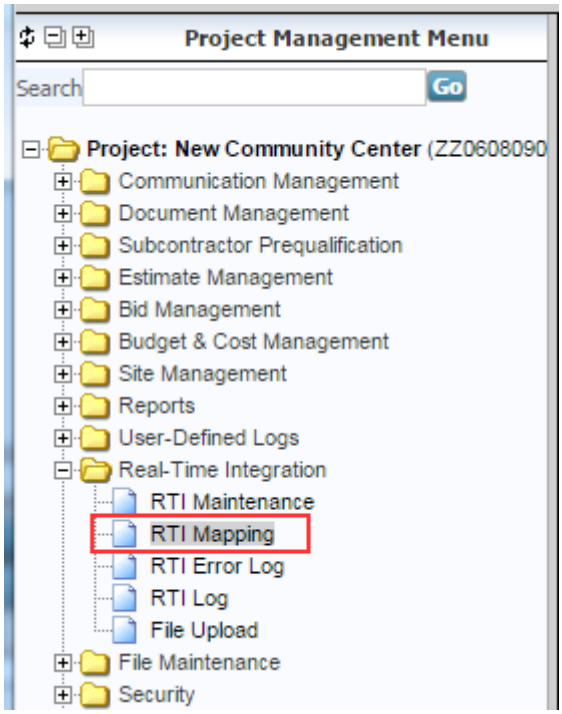
On the **External IP Address** tab, create a record which points to the Textura server being used. Enter either the FQDN Name (e.g. www.texturacorp.com) or the Site IP (e.g. 192.61.10.10). Select Yes for the Transmit Flag and No for the Receive flag (all data is retrieved by CMiC, Textura does not initiate the transfer of data to CMiC). Enter the Port if necessary and then the remainder of the URL in Transmit JEE Server (e.g. demo/api/v1). Set the value of Transmit Use SSL based on whether the communication is Secure (HTTPS), in which case Yes, or non-secured (HTTP), in which case No.

Communication Links – Tab



On the **Communication Links** tab, select Textura from the Software System dropdown list. You can set the Link and Link Identity Code as in the example below or to any value you wish, these values will be used by Textura to communicate with your CMiC software. Fill in all fields using the values you created in the previous tabs. Be sure to select the proper receive and transmit adapters from the dropdown lists.

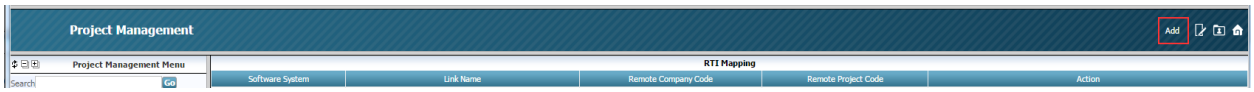
RTI Mapping



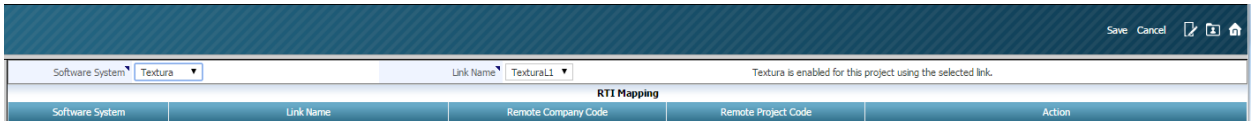
Select the RTI Mapping menu option.

NOTE: In this screen you are doing project-specific RTI setup, linking the project you are currently logged into in PM to the Communication Link you created in the RTI Maintenance **Communication Links** tab. You will need to do this step for each project that will be communicating with Textura.

Press the [**Add**] button to create a new RTI Mapping.



Select Textura from the Software System dropdown list. Select the Communication Link you created in RTI Maintenance from the Link Name dropdown list. Save the record.



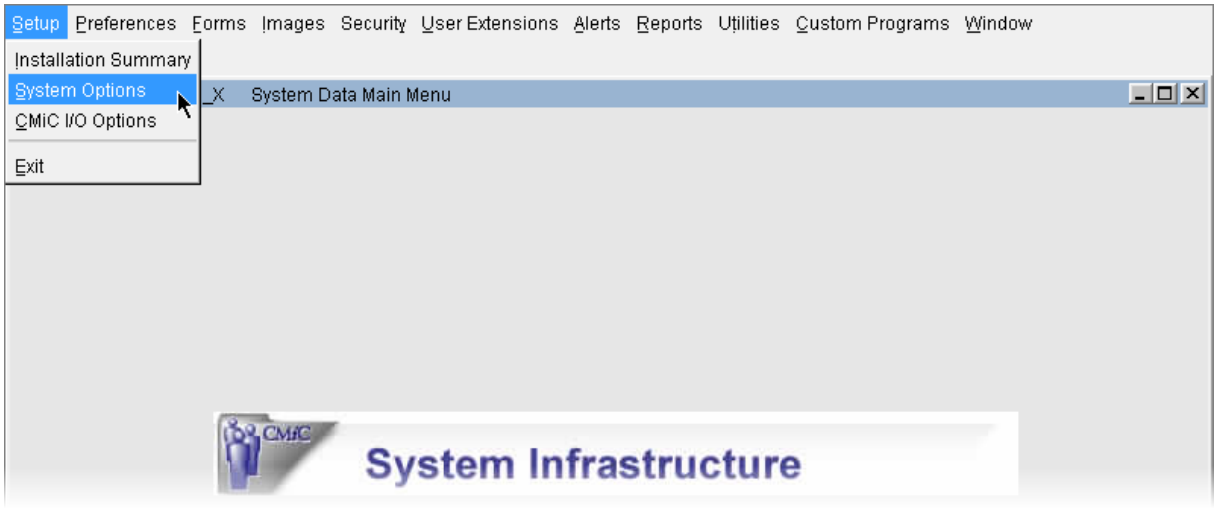
NOTE - PROJECT CODES IN TEXTURA: When creating Projects in Textura, the Project Code must be prefixed with the Company Code of the CMiC Company handling the Project, followed by a dash; otherwise, Subcontracts will not be created under their corresponding Projects.

For example, if in your CMiC system the Company Code is "**01**" and the Project Code is "**10026**", the corresponding Project Code in Textura would be "**01-10026**". For further details, contact your Textura representative.

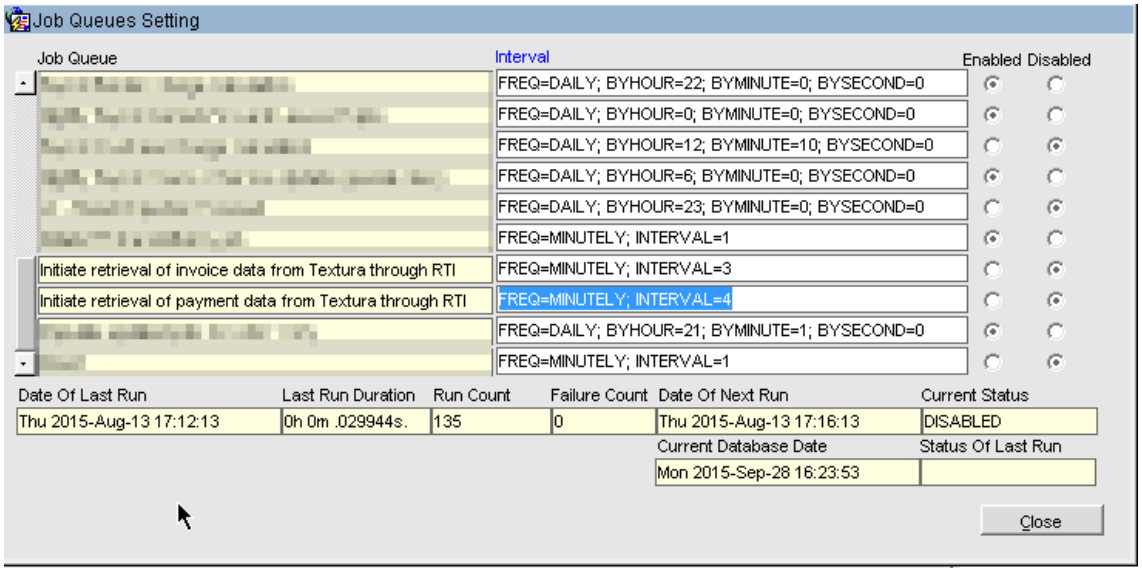
RTI Job Queue: Auto Retrieve Invoices & Payments from Textura, & Send Compliance Status to Textura

The Job Queue is used to retrieve invoices and payments from Textura, and to send compliance records to Textura. The Jobs must be enabled in order to initiate the process.

Navigate to System >> System Options:



Press the [**Job Queues**] button and scroll to the Textura jobs.

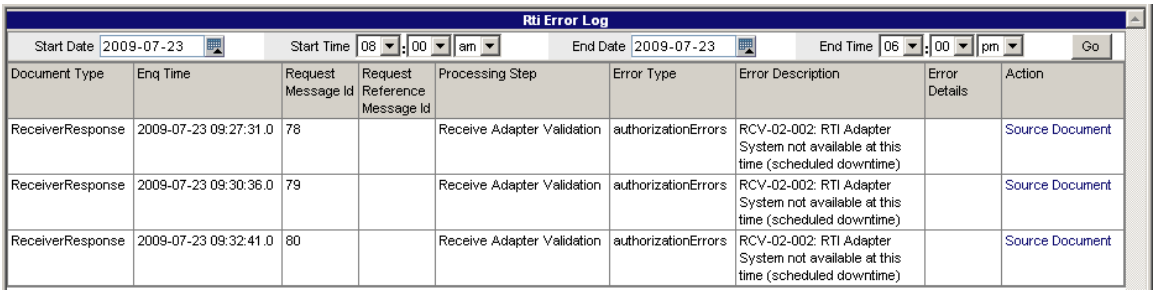


Select the **Enabled** radio button for each record.

RTI is now setup to communicate with Textura.

RTI Transmission Logs

RTI Error Log



Rti Error Log

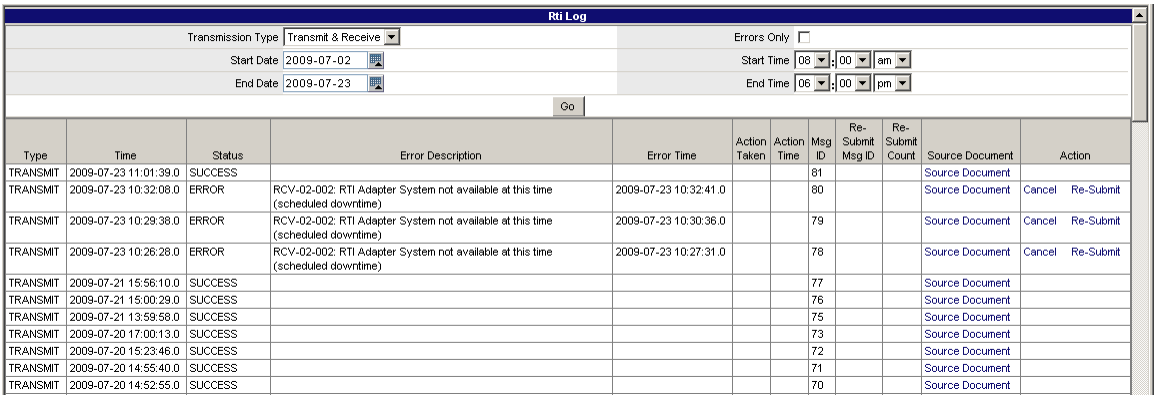
Start Date 2009-07-23 Start Time 08 : 00 am End Date 2009-07-23 End Time 06 : 00 pm Go

Document Type	Enq Time	Request Message Id	Request Reference Message Id	Processing Step	Error Type	Error Description	Error Details	Action
ReceiverResponse	2009-07-23 09:27:31.0	78		Receive Adapter Validation	authorizationErrors	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)		Source Document
ReceiverResponse	2009-07-23 09:30:36.0	79		Receive Adapter Validation	authorizationErrors	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)		Source Document
ReceiverResponse	2009-07-23 09:32:41.0	80		Receive Adapter Validation	authorizationErrors	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)		Source Document

RTI Error Log Sample

The RTI Error Log allows a review of only those transmissions that have failed. By default the log will only show current day data. To see other date/time ranges, enter the desired ranges and click the [**Go**] button to refresh the log.

RTI Log



Rti Log

Transmission Type Transmit & Receive Errors Only

Start Date 2009-07-02 Start Time 08 : 00 am

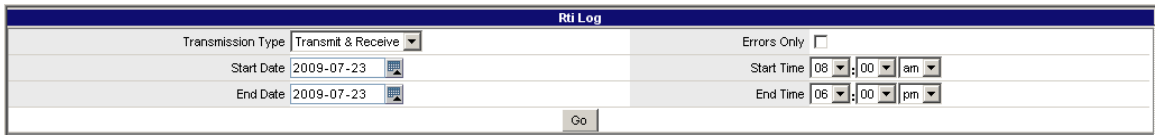
End Date 2009-07-23 End Time 06 : 00 pm

Go

Type	Time	Status	Error Description	Error Time	Action Taken	Action Time	Msg ID	Re-Submit Msg ID	Re-Submit Count	Source Document	Action
TRANSMIT	2009-07-23 11:01:39.0	SUCCESS					81			Source Document	
TRANSMIT	2009-07-23 10:32:08.0	ERROR	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)	2009-07-23 10:32:41.0			80			Source Document	Cancel Re-Submit
TRANSMIT	2009-07-23 10:29:38.0	ERROR	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)	2009-07-23 10:30:36.0			79			Source Document	Cancel Re-Submit
TRANSMIT	2009-07-23 10:26:28.0	ERROR	RCV-02-002: RTI Adapter System not available at this time (scheduled downtime)	2009-07-23 10:27:31.0			78			Source Document	Cancel Re-Submit
TRANSMIT	2009-07-21 15:56:10.0	SUCCESS					77			Source Document	
TRANSMIT	2009-07-21 15:00:29.0	SUCCESS					76			Source Document	
TRANSMIT	2009-07-21 13:59:58.0	SUCCESS					75			Source Document	
TRANSMIT	2009-07-20 17:00:13.0	SUCCESS					73			Source Document	
TRANSMIT	2009-07-20 15:23:46.0	SUCCESS					72			Source Document	
TRANSMIT	2009-07-20 14:55:40.0	SUCCESS					71			Source Document	
TRANSMIT	2009-07-20 14:52:55.0	SUCCESS					70			Source Document	

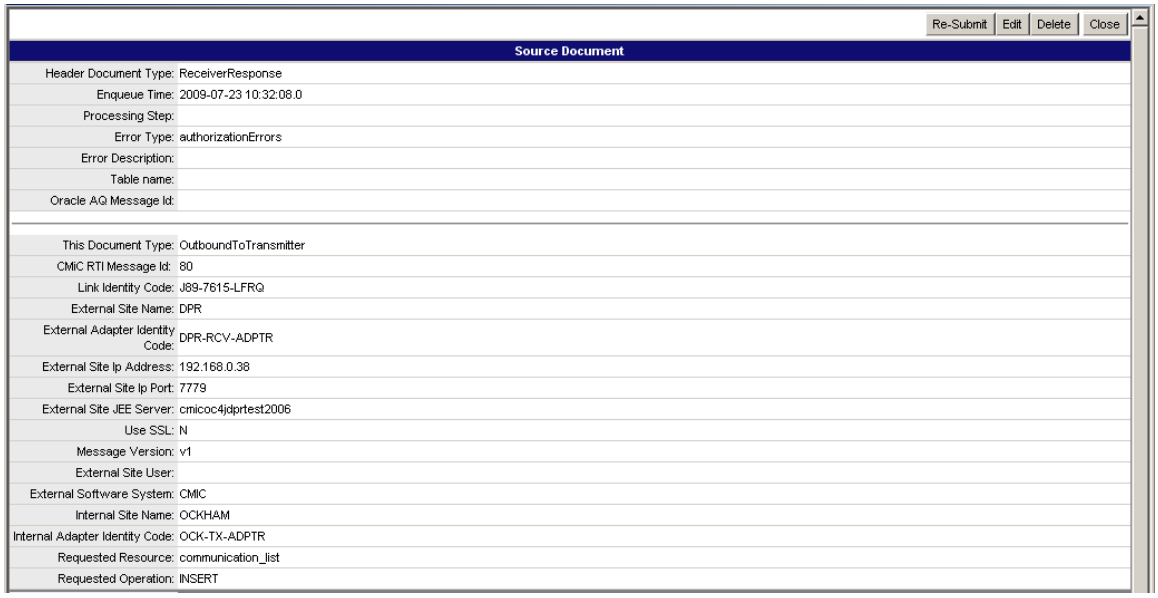
RTI Log Sample

The RTI Log is an access point to allow users to review all RTI Transmissions. By default, the current date will be shown when opening the RTI Log. By making adjustments, the data to be displayed once the user clicks the [**Go**] button will vary. Options include Transmission Type (Options include: Transmit & Receive, Transmit only, and Receive only), From/To Date and Time ranges, and also whether to show all data, or only data where an Error occurred.



If the transmission or receive was successful, the only option on those item types will be to review the Source Document (the actual XML file transmitted or received). If however there was an error or that processing was not completed, there will be options for additional activity consisting of [**Cancel**] and [**Re-Submit**]. There is also a [**Delete**] option if currently in the Source Document display.

Source Document – RTI Log and RTI Error Log



RTI Log – Source Document Sample (Header Section)

The [**Source Document**] link in the RTI Log and RTI Error Logs allow the user to review the item and shows two sections, the header information as illustrated above and the details section shown below.

Instruction Keys	Position	Name	Data Type	Format Mask	Value	Value Type
Key Info	Position	Name	Data Type	Format Mask	Value	Value Type
Queryy Info	Name	Data Type	Value			

	Row #	Name	Data Type	Format Mask	Value	Value Type
Rows of Data	1	COMM_CLASSIFIER1	VARCHAR2			new
	1	COMM_CLASSIFIER2	VARCHAR2			new
	1	COMM_CLASSIFIER3	VARCHAR2			new
	1	COMM_CLASSIFIER4	VARCHAR2			new
	1	COMM_CLASSIFIER5	VARCHAR2			new
	1	COMM_CLASSIFIER6	VARCHAR2			new
	1	COMM_COMMUNICATION_ID	VARCHAR2		A-0723-003	new
	1	COMM_COMP_CODE	VARCHAR2		ZZ	new
	1	COMM_DATE	DATE	DD-MON-RRRR HH24:MI:SS	23-JUL-2009 10:31:59	new
	1	COMM_FOLLOWUP_CONTACT_CODE	VARCHAR2			new
	1	COMM_FOLLOWUP_DATE	DATE	DD-MON-RRRR HH24:MI:SS		new
	1	COMM_FOLLOWUP_DUE_DATE	DATE	DD-MON-RRRR HH24:MI:SS		new
	1	COMM_FOLLOWUP_PARTN_CODE	VARCHAR2			new
	1	COMM_FOLLOWUP_PARTN_TYPE	VARCHAR2			new
	1	COMM_FOLLOWUP_REQUIRED_FLAG	VARCHAR2		N	new
	1	COMM_FOLLOWUP_TEXT	VARCHAR2			new
	1	COMM_FROM_CONTACT_CODE	VARCHAR2		PETE	new
	1	COMM_FROM_PARTN_CODE	VARCHAR2		ZZ	new
	1	COMM_FROM_PARTN_TYPE	VARCHAR2		C	new
	1	COMM_PROJ_CODE	VARCHAR2		OCKHAM1	new
	1	COMM_STATUS	VARCHAR2		O	new
	1	COMM_SUBJECT	VARCHAR2		KT 10:31 am	new
	1	COMM_TEXT	VARCHAR2		KT 10:31 am	new
	1	COMM_TO_CONTACT_CODE	VARCHAR2		MOAWIA	new
	1	COMM_TO_PARTN_CODE	VARCHAR2		DPR	new
	1	COMM_TO_PARTN_TYPE	VARCHAR2		P	new
	1	COMM_TYPE	VARCHAR2		COLAB	new

File Uploads	File #	Path	Application	File Type	Document Type

RTI Log – Source Document Sample (Details Section)

The data shown will vary based on the record type (e.g. Communication Record vs. RFI Record) and the details section will also include if there were any related files being transmitted.

Appendix

Enabling RTI in v10x

Complete the following steps to enable RTI in v10x:

1. Take a backup of opmn.xml under MIDTIERJSP_HOME\asinst_midtierjsp\config\OPMN\opmn.
2. Edit opmn.xml and find entry line <process-type id="CMICRTI_ENV" module-id="CUSTOM" status="disabled">.
3. Change line to <process-type id="CMICRTI_enabled" module-id="CUSTOM" status="enabled">.
4. Restart the OPMN on that Oracle Home.

NOTE: This process only needs to be done on 1 JSP server and only 1 JSP server for that environment.

Configuring to Communicate to an SSL Server

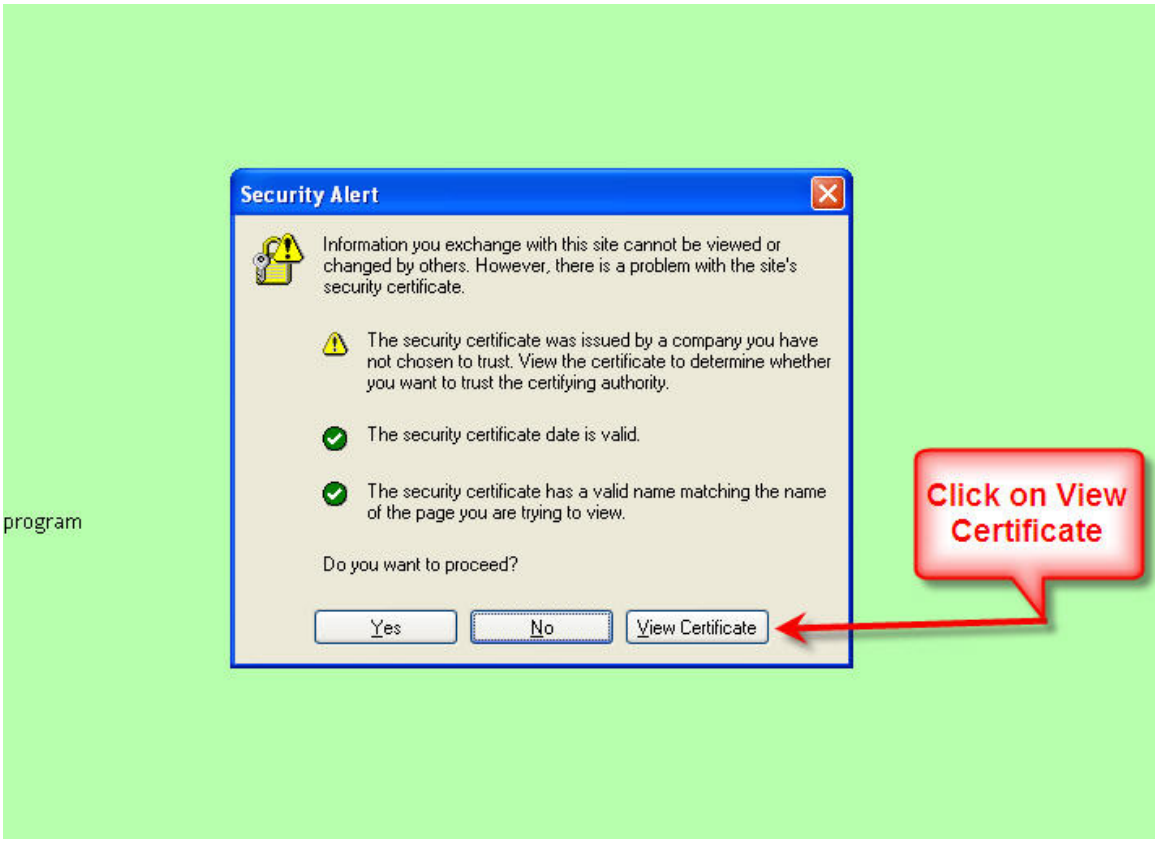
The following information is provided to assist in setting up your environment to Use an SSL certificate with RTI.

Downloading the certificate:

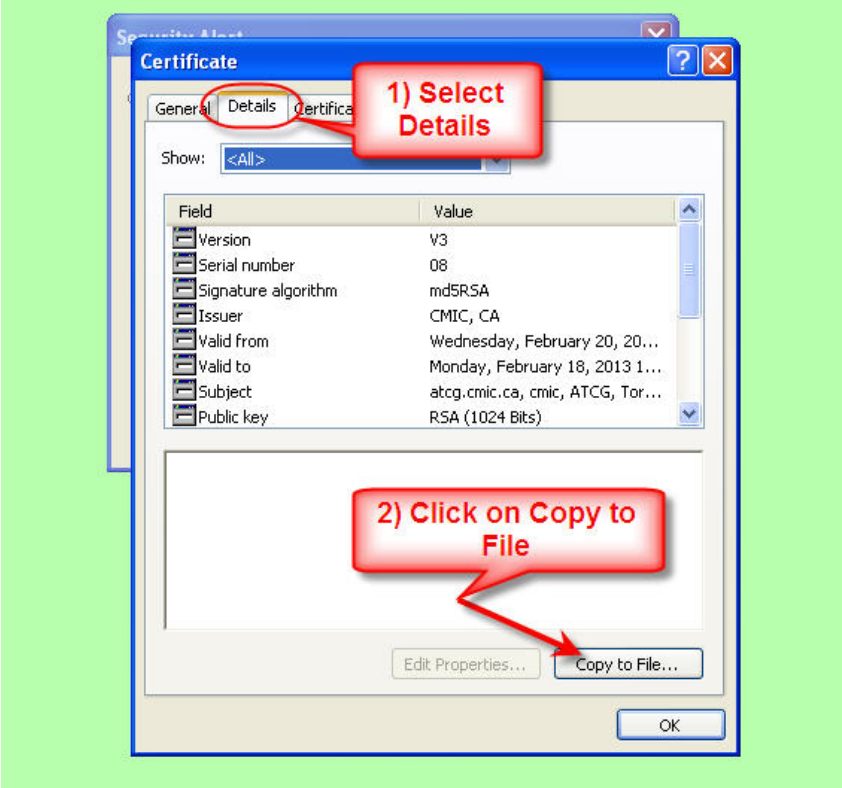
Open a browser and try to access a JSP from the remote environment:

https://atcg.cmic.ca/cmicpublicprod/CMiCPublic/cmicpublic.jsp

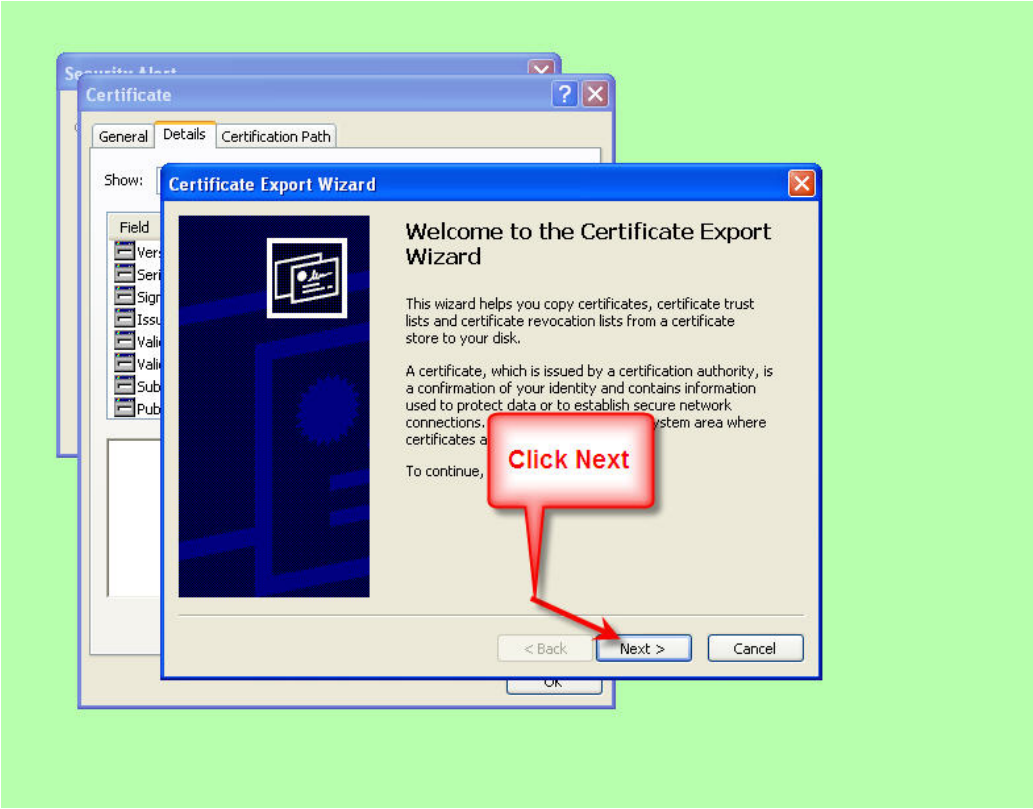
Once the pop-up appears, click on [**View Certificate**]



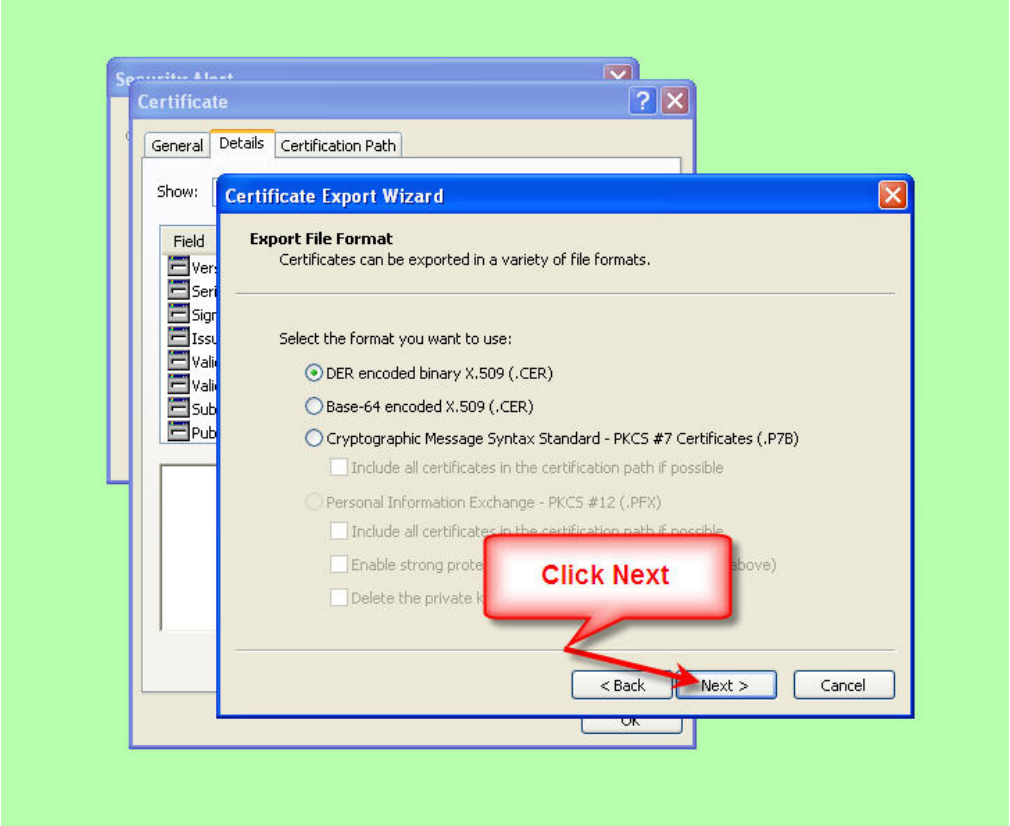
Select Details tab, then click on [**Copy to File**]:



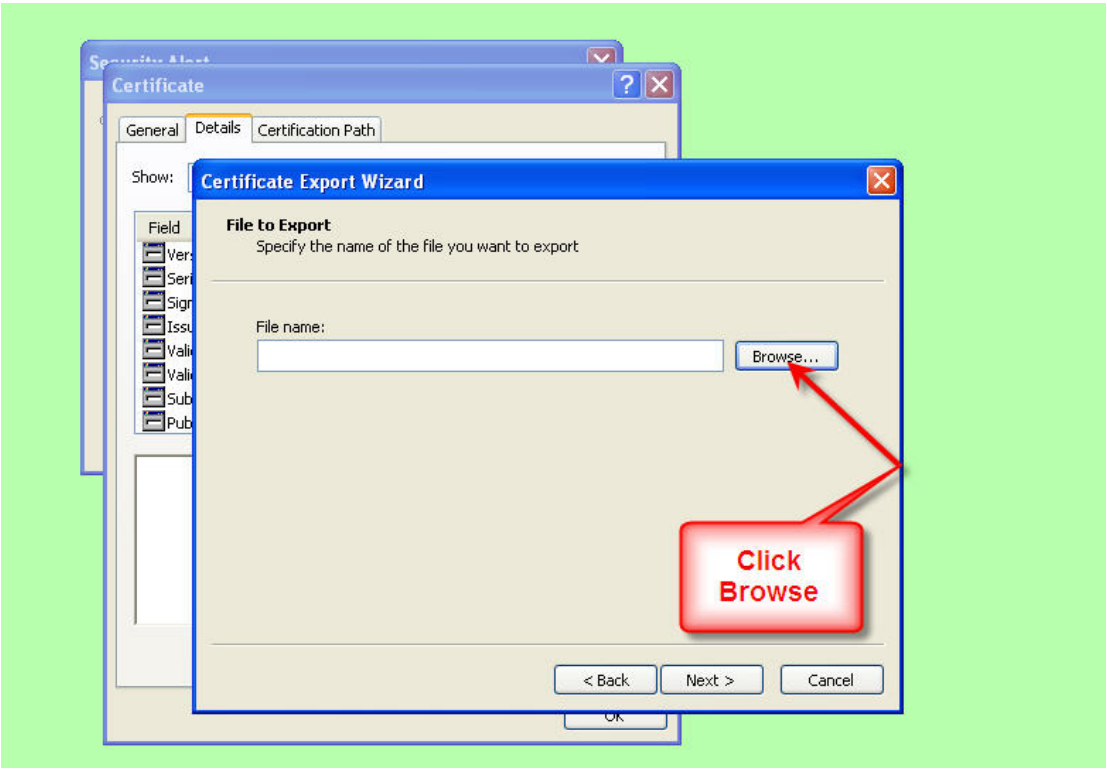
Click [**Next**]



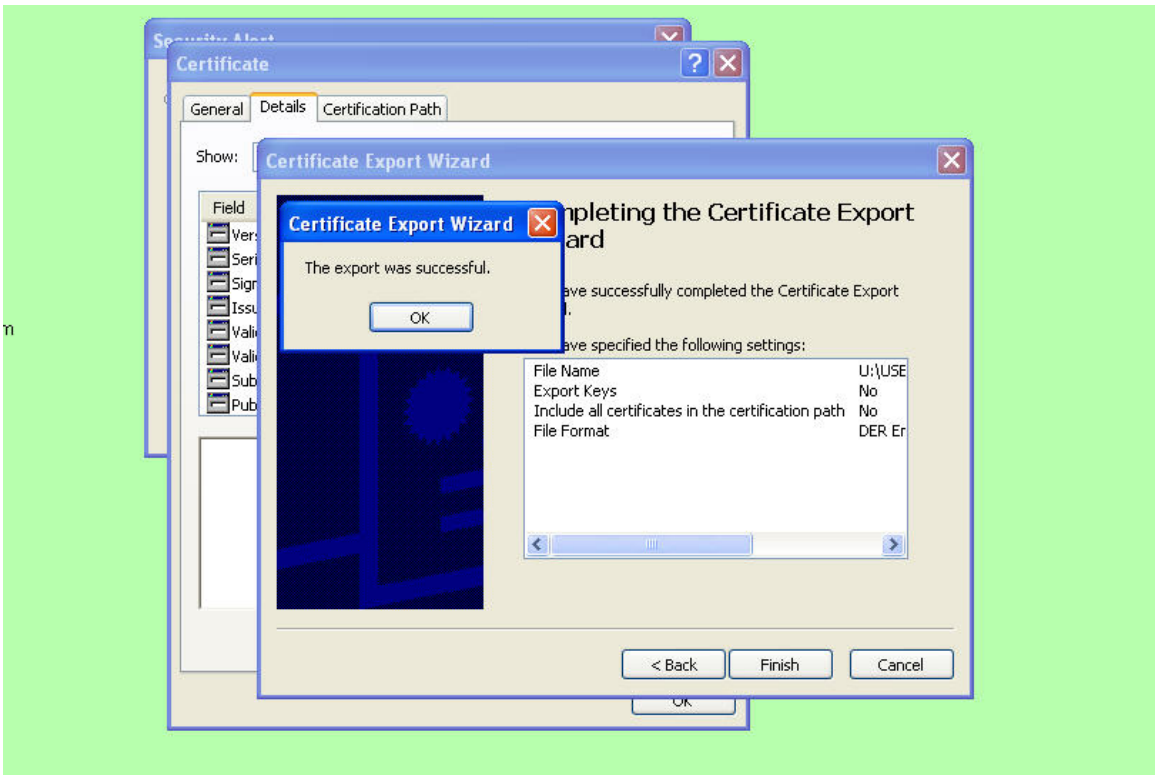
Click [**Next**] again:



Click [**Browse…**] and select a file and save it:



Once the file is saved you'll receive the following message:



Import the SSL Certificate into a Keystore

Assume the certificate was saved into a file called certificate.cer, import the certificate into a keystore (call it aKeystore) using the following command:

- If keytool is from JRE 1.4 or higher use the following command:

keytool –importcert -trustcacerts -file certificate.cer -alias certificate.cer -keystore aKeystore -storepass <passwd>

- If keytool is from JRE before 1.4 use the following command:

keytool -import -trustcacerts -file certificate.cer -alias certificate.cer -keystore aKeystore -storepass <passwd>

Add Keystore in opmn.xml

1) find process-type tag with id="<OC4J_instance>"
2) find sub tag <module-data>
3) find sub tag <category id="start-parameters">
4) find sub tag <data id="java-options" and add the following property:

   -Djavax.net.ssl.trustStore=*aKeystore*

Where *aKeystore* is the full path to the file where the certificate was imported.

Dropping and Recreating DB Queues

The following section provides steps for dropping and recreating DB queues.

Stop and Drop the Queues

Run the following statements to stop and then drop each RTI queue and queue table:

```
EXECUTE dbms_aqadm.stop_queue(Queue_name => 'RTI_ERROR_Q');
EXECUTE dbms_aqadm.drop_queue(Queue_name => 'RTI_ERROR_Q');
EXECUTE dbms_aqadm.drop_queue_table(Queue_table => 'RTI_ERROR_Q_T');

EXECUTE dbms_aqadm.stop_queue(Queue_name => 'RTI_INBOUND_Q');
EXECUTE dbms_aqadm.drop_queue(Queue_name => 'RTI_INBOUND_Q');
EXECUTE dbms_aqadm.drop_queue_table(Queue_table => 'RTI_INBOUND_Q_T');

EXECUTE dbms_aqadm.stop_queue(Queue_name => 'RTI_OUTBOUND_Q');
EXECUTE dbms_aqadm.drop_queue(Queue_name => 'RTI_OUTBOUND_Q');
EXECUTE dbms_aqadm.drop_queue_table(Queue_table => 'RTI_OUTBOUND_Q_T');
```

Recreate the Queues

As user UIG, run script rtiqueue.que, which would typically be found on the server at d:\cm\v10\<environment>\uig\sql\. This script creates and enables all queues, so nothing else should need to be done.

Index

A

B

C

D

E

F

I

L

O

P

R

S

T

U

V

X