

# Transaction Designer for XML

Version 4

User's Guide  
Revision 1.5

CLE|E|O

---

Copyright ©2005 Cleo Communications - Cleo Communications reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials. This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Cleo Communications.

#### GOVERNMENT RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Use, reproduction or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in the accompanying end-user agreement.

#### GOVERNMENT LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Cleo Communications has made every effort to accurately acknowledge all trademarks that appear in this document. Cleo Communications, however, cannot attest to the accuracy of this information.

Transaction Designer, Transaction Processor, and Transaction-based Processing are trademarks and Cleo is a registered trademark of Cleo Communications. All other brand names are trademarks or registered trademarks of their respective owners.

---

# TABLE OF CONTENTS

<b>INTRODUCTION.....</b>	<b>5</b>
OVERVIEW .....	7
COMPONENT DIAGRAM.....	8
COMPONENT DESCRIPTIONS.....	9
KEY FEATURES .....	10
<b>USING THE TRANSACTION DESIGNER.....</b>	<b>11</b>
TRANSACTION DESIGNER WORKBENCH.....	11
<i>Transactions Perspective</i> .....	12
<i>Recorder Perspective</i> .....	14
MENU BAR DESCRIPTIONS.....	15
<i>File</i> .....	15
<i>Edit</i> .....	15
<i>Project</i> .....	15
<i>Window</i> .....	16
<i>Help</i> .....	17
TOOLBAR DESCRIPTIONS .....	17
PERSPECTIVE BAR DESCRIPTION.....	18
LICENSE THE TRANSACTION DESIGNER .....	19
RUN THE TRANSACTION DESIGNER.....	20
<i>Create a New Project During Startup</i> .....	20
<i>Open an Existing Project During Startup</i> .....	20
<i>Open a Project After Startup</i> .....	21
<i>Delete an Existing Project</i> .....	22
<i>Cancel</i> .....	23
IMPORT XML DOCUMENTS.....	24
RECORD XML DOCUMENTS .....	25
<i>Access the XML Recorder</i> .....	26
<i>Import or Select an XML Document to Send</i> .....	26
Optionally Edit the XML .....	27
<i>Optionally Import XML Request</i> .....	28
<i>Create an HTTP Connection</i> .....	28
<i>Delete an HTTP Connection</i> .....	29
<i>Modify HTTP Header</i> .....	29
<i>Send XML to the Server</i> .....	29
<i>Import XML Response</i> .....	30
CREATE A NEW TRANSACTION .....	32
CONFIGURE A TRANSACTION .....	33
<i>Define Requests</i> .....	34
View a Request .....	34
Configure the Resource.....	36
Add an Input Field .....	36
<i>Define Responses</i> .....	37
View a Response.....	38
Add an Identifier .....	39
Add an Output Field.....	40
<i>Element Icon States</i> .....	42
<i>Attribute Icon States</i> .....	42
<i>Using Branches</i> .....	42

Define a Branch .....	43
<i>Using Rules</i> .....	44
Define a Rule .....	44
<i>Problems</i> .....	46
MODIFY AN EXISTING TRANSACTION .....	47
<i>Cut, Copy, and Paste</i> .....	47
<i>Rename a Request or Response</i> .....	47
<i>Change the XPath</i> .....	48
<i>Rename input/output fields</i> .....	48
<i>Rename a Transaction</i> .....	49
<i>Delete an Existing Transaction</i> .....	49
PUBLISH A TRANSACTION .....	50
TRANSFER THE PUBLISHED TRANSACTION SET.....	51
<b>PULLING IT ALL TOGETHER.....</b>	<b>53</b>
<b>APPENDIX A: TROUBLESHOOTING .....</b>	<b>55</b>
PROBLEM DESCRIPTIONS .....	55
<b>APPENDIX B: TRANSACTION MAP .....</b>	<b>58</b>
SAMPLE TRANSACTION MAP.....	58
<b>GLOSSARY.....</b>	<b>59</b>
<b>INDEX.....</b>	<b>61</b>

# Introduction

Cleo's Transaction-based Processing has been extended to allow transactions to be created using XML documents, in addition to TN3270 and TN5250 screens.

Traditionally, programs that access XML data retrieved from various sources, such as database front ends, web services, and custom server applications, generally rely on XML navigation tools such as XSLT, XPath, or XQuery. Although these tools are effective, they represent a level of programming complexity that often requires a specialist. Cleo's Transaction Development Kit provides a simplified mechanism for defining and processing custom XML "transactions."

XML based transactions are created using a simple GUI interface to indicate which XML documents are included in a transaction and which fields contain input or output data. These transactions contain all the information necessary to interact with a set of XML documents. Thus, once transactions have been created, the developer can send and extract XML data using fewer calls than traditionally required. This simplifies the development process that in turn leads to reduced development time and higher reliability. The GUI interface may be used by any developer, including those not familiar with the details of exchanging or managing XML documents.

The Cleo Transaction development kit is composed of:

- A *Transaction Designer*<sup>™</sup> (TD) – that captures XML documents and creates transactions.
- A *Transaction Processor*<sup>™</sup> (TP) – that processes the transactions in real time and provides the basic API functions.
- An *Administration GUI* - that provides a Configuration tool and an Administration tool for the Transaction Processor.

The documentation set that supports the development kit consists of the following:

***Quick Start Guide for Transaction Designer***

This guide lists prerequisites and provides an installation procedure for the Transaction Designer.

***Cleo Transaction Designer for XML User's Guide***

This guide explains how to capture XML documents and create transactions.

***Quick Start Guide for Transaction Processor***

This guide lists prerequisites and provides an installation procedure for the Transaction Processor.

***Administration Guide***

This document describes the web-based administration utility for the Transaction Processor product.

***Programmer's Guide***

This guide provides information about developing applications using the Transaction Processor.

This manual, *The Transaction Designer for XML User's Guide*, contains the detailed instructions for using the Transaction Designer.

The intended audience for this document is presumed to be familiar with application development on a Windows platform, as well as having some knowledge of using XML over HTTP for accessing applications running on a remote application server.

The Transaction Designer is a plug-in to Eclipse, an open-source development environment. Although not required, some familiarity with Eclipse may also be useful.

## Overview

---

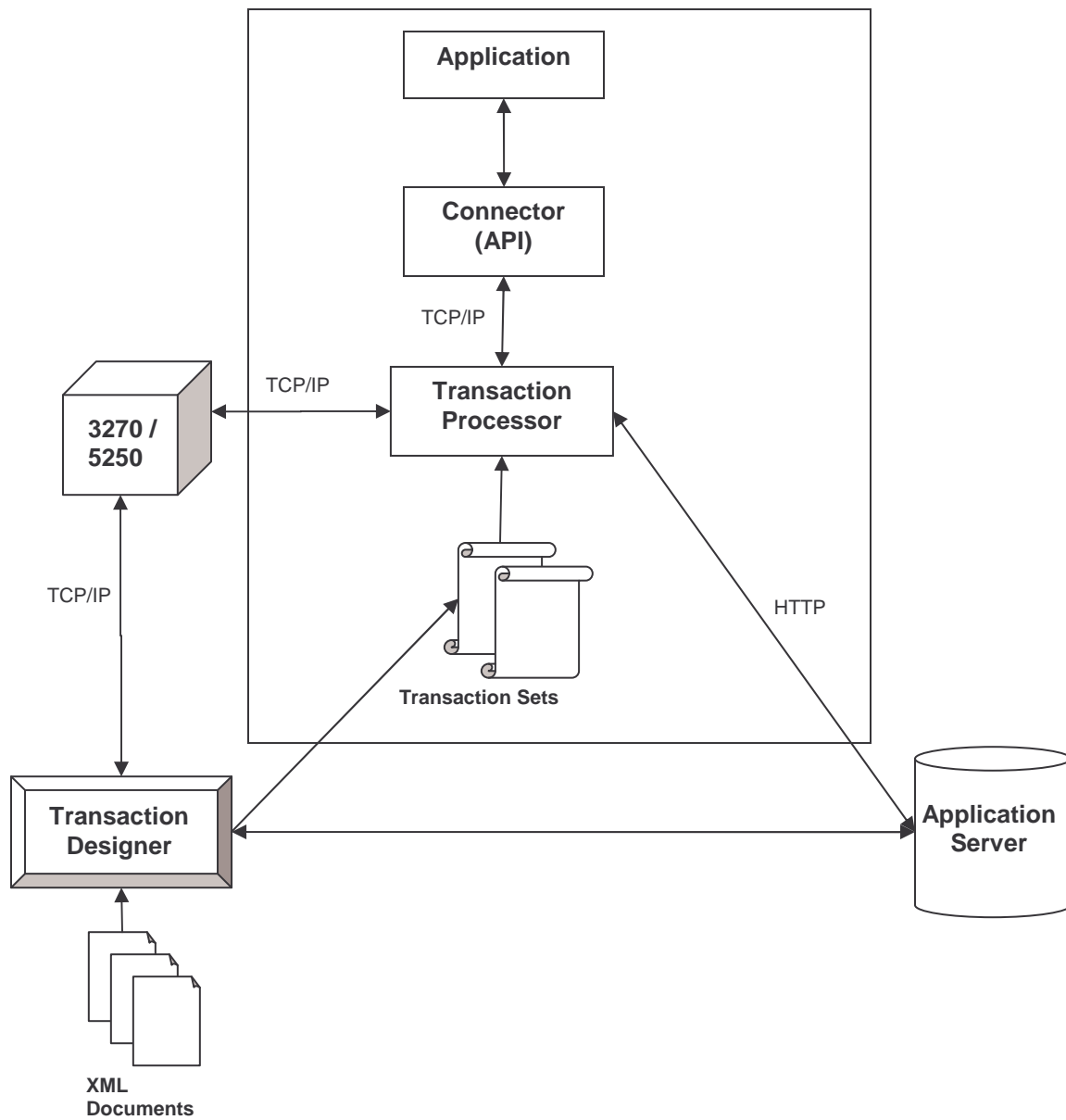
The purpose of the Transaction Designer is to allow a user to define transactions within a simple GUI based environment.

Consider the following exchange where XML is used to retrieve an account balance.

- A login request XML document containing a **login id** and **password** is sent to the server.
- A response XML document is received from the server indicating whether or not Login was **successful**.
- If successful, a balance inquiry request XML document is sent to the server containing the **account number** and **password**.
- A balance inquiry response XML document is received from the server containing an **account balance**.

The Transaction Designer allows the user to take the XML documents used in this sequence and group them together into a single transaction that can be executed from the Transaction Processor. In documents sent to the host, elements or attributes can be designated as Input Fields (e.g. **login id, password**). In the document received from the host, elements or attributes can be designated as Identifiers (Used to identify a document correctly) or Output fields (e.g. **account balance**). Grouping these documents into transactions significantly reduces the effort needed to modify a document to be sent to a host and to find the data in a document received from the host.

# Component Diagram



## Component Descriptions

---

- **Application** – User-defined program that invokes the TP through the Connector.
- **Application Server** – System that runs the application that exchanges XML messages with the TP over HTTP.
- **Connector (API)** – Provides the programming interface into the TP.
- **Transaction Designer (TD)** – Allows the user to define the properties of each incoming and outgoing XML message that make up a transaction. Generates the transaction set that can then be used by the TP to interact with the Application Server.
- **Transaction Processor (TP)** – Companion runtime component to the TD that provides the transaction based processing of 3270/5250 screens and XML messages.
- **Transaction Sets** – Contain the information (XML document definitions and transactions) required by the TP in order to recognize XML messages returned from the Application Server, parse the XML messages for output, and generate XML messages populated with user supplied input to send to the Application Server.
- **XML Documents** – represent the actual XML messages exchanged at runtime. These documents may pre-exist (or can be recorded by the TD) and are then imported.
- **3270/5250** – Mainframe host providing access to 3270/5250 applications.

## Key Features

---

The Transaction Designer provides:

- A way to create transactions using imported XML documents.
- A way to capture XML that is exchanged with a remote application.
- A display of imported documents that can be used to create transactions.
- A means to modify transactions.
- A means to publish XML document definitions and transactions to files for use by the Transaction Processor.

# Using the Transaction Designer

This section describes how to use the Transaction Designer. It includes how to:

- Use the Menu Bar and Toolbar
- Run the Transaction Designer
- Record XML Documents
- Import XML Documents
- Create a Transaction
- Publish a Transaction
- Transfer a Published Transaction Set
- Modify an Existing Project
- Modify an Existing Transaction

## Transaction Designer Workbench

---

The Transaction Designer workbench is made up of both *views* and *perspectives*. A *view* is a tabbed window on the workbench that allows the user to view, navigate, and modify the resources used in the project.

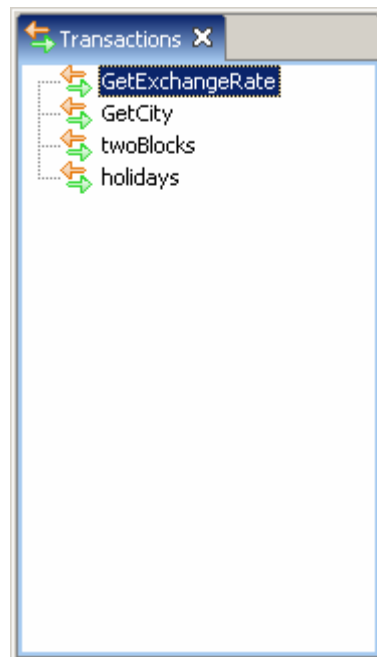
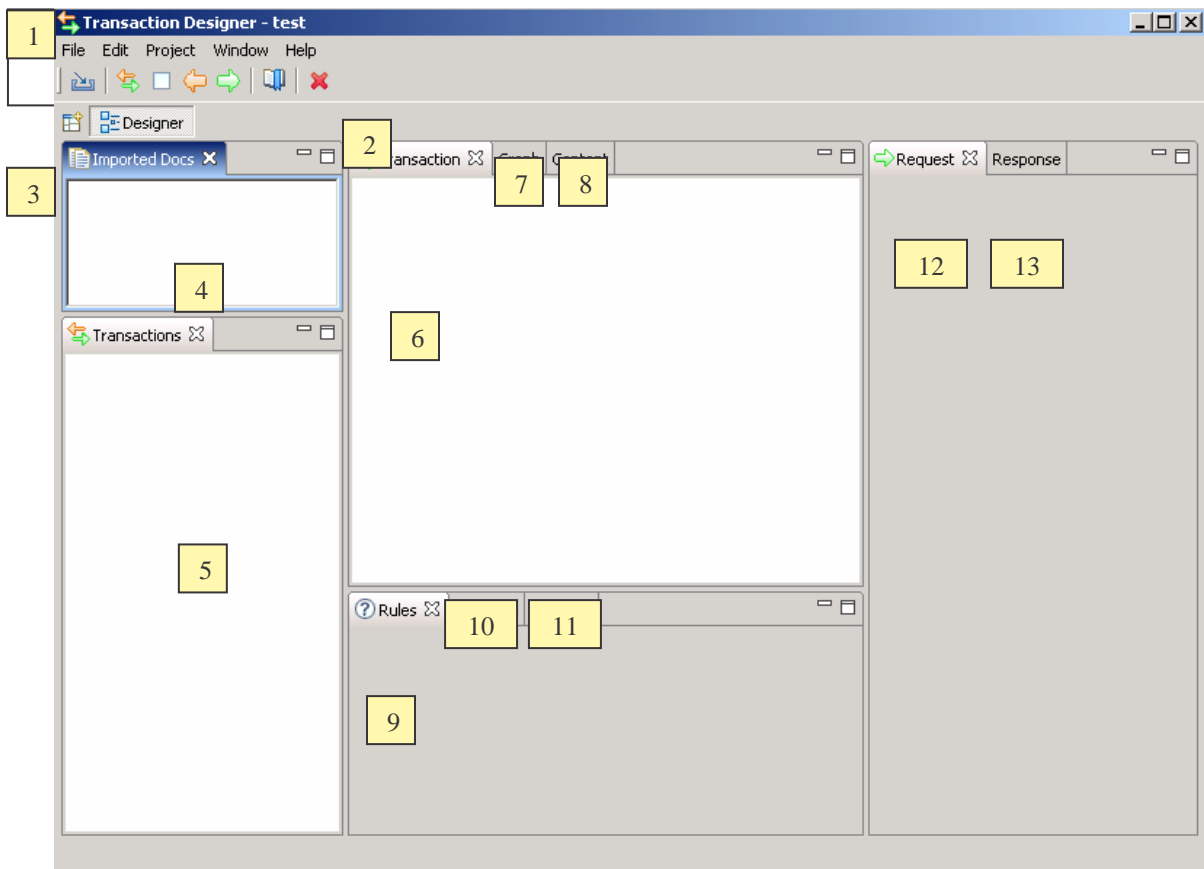


Figure 1: Transactions View

A collection of views in the Transaction Designer is referred to as a *perspective*. A perspective can also be thought of as a mode of operation. In the TD, there are two perspectives or two different modes of operation. These two perspectives are called the “Designer Perspective” and the “Recorder Perspective” and allow the user to accomplish different tasks in the TD. The “Designer Perspective” is where XML documents are grouped together to define transactions, while the “Recorder Perspective” provides the user with a mechanism for capturing XML documents received from a host. Each component of the “Transactions Perspective” and the “Recorder Perspective are described below.

## Transactions Perspective

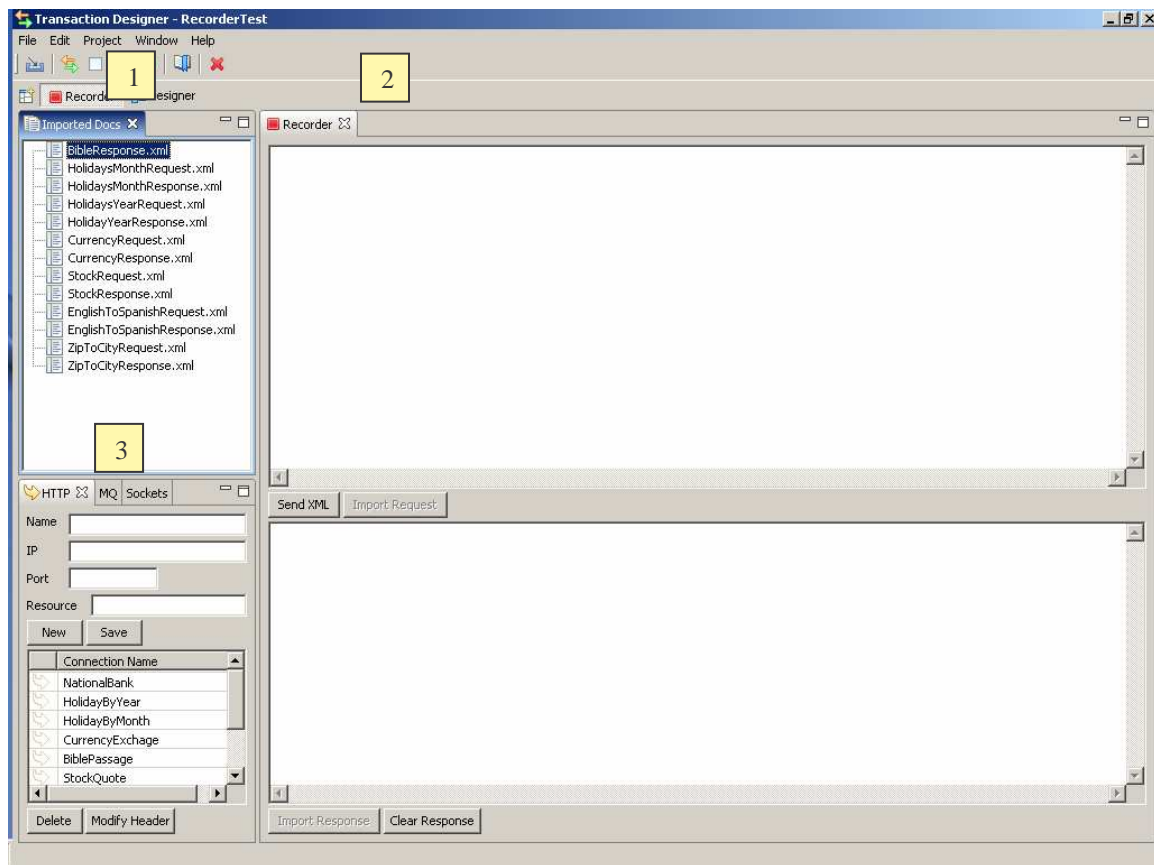


**Figure 3: The Transactions Perspective**

1. **Menu Bar:** provides access to the main functions and is described in the next section.

2. **Toolbar:** provides access to common functions and is described in the next section.
3. **Perspectives Bar:** allows the user to switch between the Designer perspective and the Recorder perspective.
4. **Imported Docs View:** Any XML documents that are imported are displayed in this area.
5. **Transactions View:** Open transactions are listed here.
6. **Transaction View:** Displays a representation of the selected transaction in the form of a tree containing the Request and associated Responses.
7. **Graph View:** Displays a representation of the selected transaction in the form of a graphical chart showing the flow of the transaction.
8. **Content View:** Displays the raw content of the XML that the selected imported doc, request, or response originated from.
9. **Rules View:** Provides a means to create rules to tie together transactions in sequence based on output field values. See section on *Creating Rules* for more details.
10. **Branches View:** Provides a means to create rules used to tie together transactions in sequence based on the response received from the host. See section on *Creating Branches* for more details.
11. **Problems View:** Lists errors contained in the current project.
12. **Request View:** Shows the details of the selected Request in a tree form. In this view, the user can designate certain elements or attributes of the document as Input Fields.
13. **Response Views:** Shows the details of a Response in a tree form. In this view, the user can designate certain elements or attributes of the document as Output Fields or Identifiers.

## Recorder Perspective



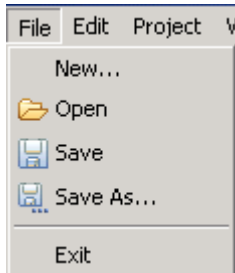
1. **Imported Docs View:** Displays a list of XML documents that can be used to create transactions.
2. **Recorder View:** This view contains two windows. XML to be sent to the host appears in the upper window by either clicking on an imported document or by pasting XML into the window. When the send button is pressed, the response is displayed in the lower window.
3. **Http View:** This is where connections to servers along with any associated header information and/or resources are configured for Http.

## Menu Bar Descriptions

---

The menu bar is located at the top of the layout and provides the following functions:

### File



**New:** Create a new Project

**Open:** Open an existing Project

**Save:** Save the current Project

**Save as:** Save the current Project under a different name

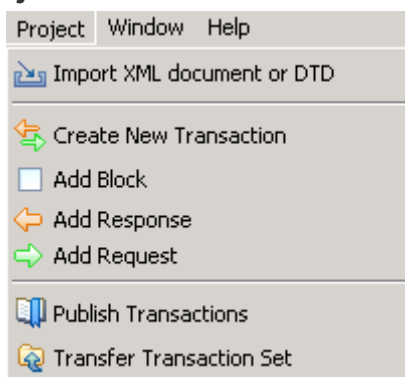
**Exit:** Exit the application

### Edit



**Delete:** Deletes the selected transaction, component of a transaction, or imported document.

### Project



**Import XML document:** Imports an XML document to be used in transactions for the current Project.

**Create New Transaction:** Creates a New Transaction

**Add Block:** Adds a Request / Responses Block to a Transaction.

**Add Response:** Adds a Response to a Transaction.

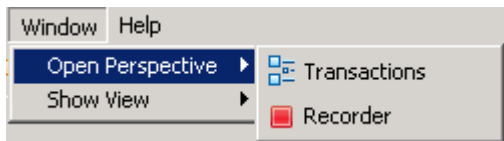
**Add Request:** Adds a Request to a Transaction.

**Publish Transactions:** Publishes Transactions for use by the Transaction Processor.

**Transfer Transaction Set:** Sends the current Transaction Set to a specified local folder or remote server.

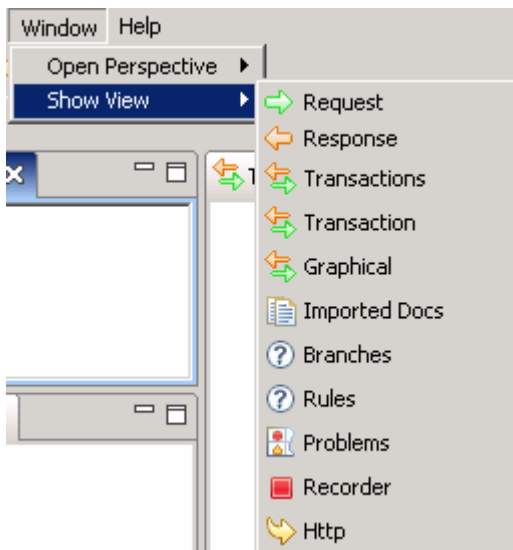
## Window

### Open Perspective



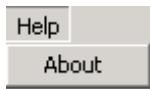
- **Transactions:** the default perspective upon opening the Transaction Designer. In this perspective, transactions are created and defined.
- **Recorder:** Changes the perspective from Transactions to Recorder. The recorder allows you to capture XML documents as they are passed to and from a server. To go back to the main Transaction perspective, click Window > Open Perspective > Transactions.

### Show View

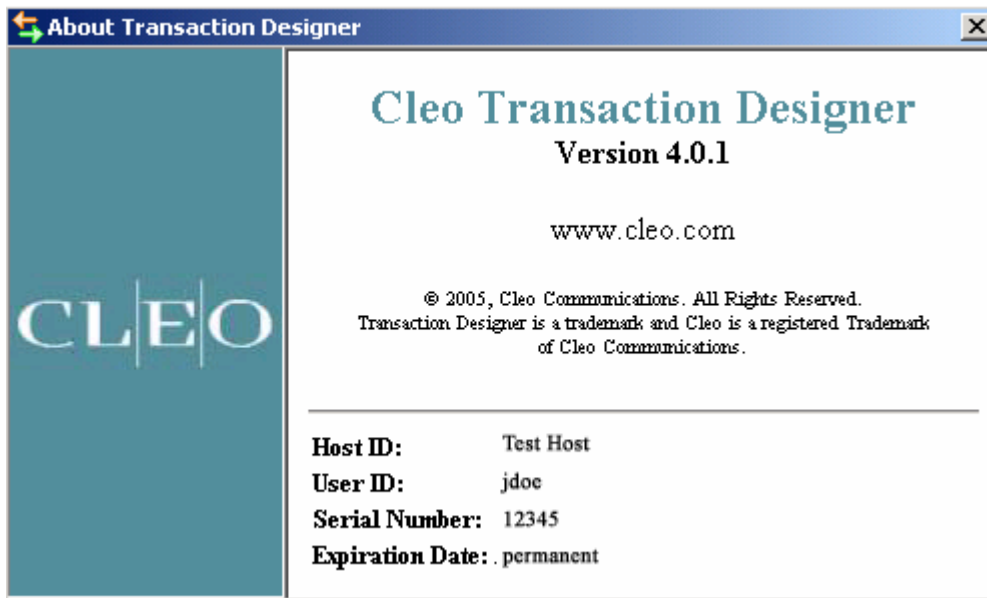


Show view restores views or panels if they are closed and brings the selected view into the forefront.

## Help



**About:** Displays product, version, and manufacturer information as well as licensing information.



## Toolbar Descriptions

---

The toolbar provides access to common functions.



Import XML Document



Create a new transaction



Add a Block to the current transaction



Add a response to the selected block



Add a request to the selected block



Publish transaction



Delete item



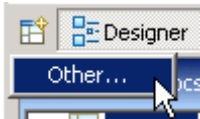
Switch between Transaction perspective and Recorder perspective.

## Perspective Bar Description

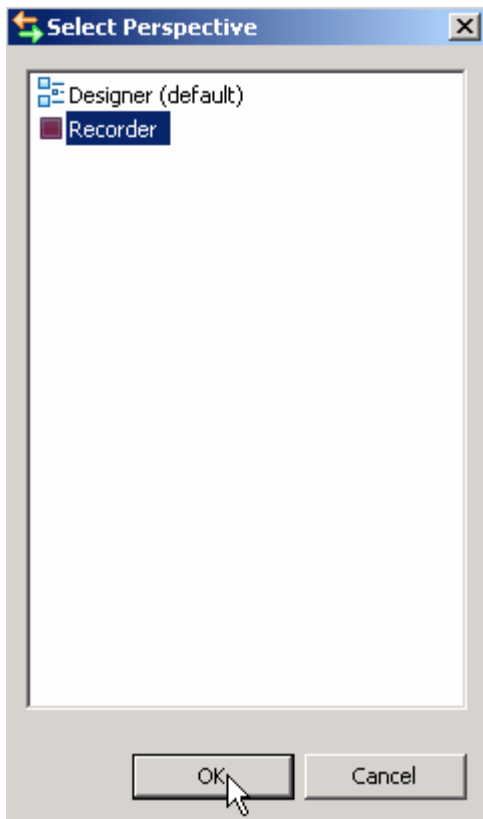
---



The Perspective Bar provides another means to switch between the Transactions perspective and the Recorder perspective. See “Transaction Designer Workbench” for an explanation of Perspectives. To switch from the current perspective to a new perspective, click the Perspective Icon and select **Other**.



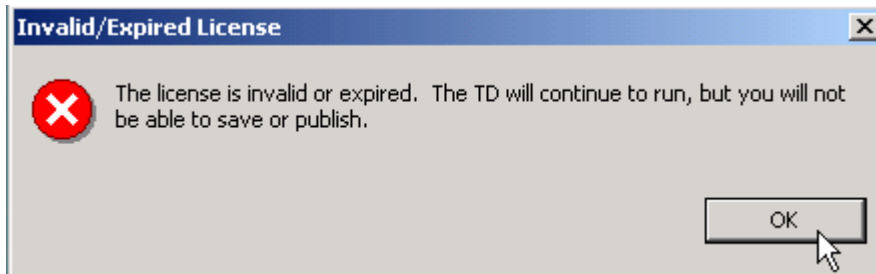
Choose the desired perspective and click **OK**.



## License the Transaction Designer

---

Without a valid license key, you will not be able to save or publish transactions.



To begin using all the functions of the Transaction Designer, you will need to obtain a license key. If you do not already have a key, email [salesEN@cleo.com](mailto:salesEN@cleo.com) to request your license key. You will need to provide your host id, user id and serial number. To obtain this information, open the Transaction Designer and in the File menu, click on **Help > About**.

Once you have received your license key, save it in the following location:

`%XTD_HOME%\plugins\com\cleo.td_4.0.0`

(e.g. C:\TDesigner\plugins\com.cleo.td\_4.0.0). The Transaction Designer will now be fully functional.

## Run the Transaction Designer

---

Project files contain useful information about which transactions are defined and other properties such as last published date.

**Note:** All Project files have a .cxp extension and are stored in the associated transaction directory (e.g. C:\Tdesigner\trans\test123).

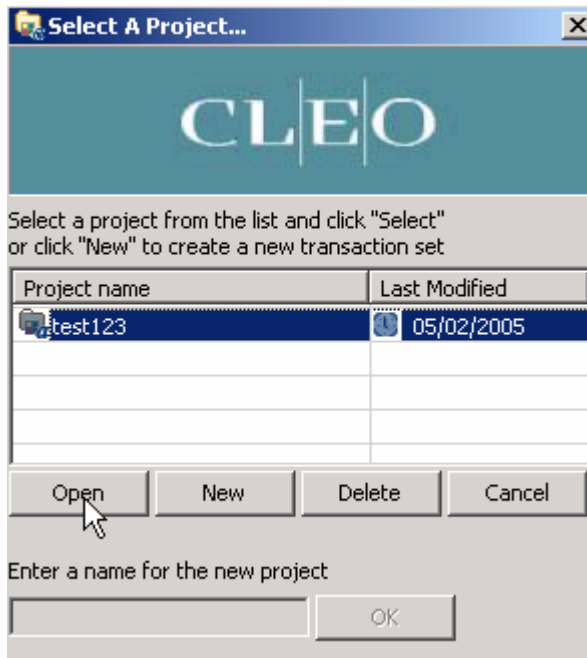
### Create a New Project During Startup

1. Open the Transaction Designer. A project menu screen will appear.
2. Enter a project name and click **OK**. You will then be taken to the main Transaction Designer screen.



### Open an Existing Project During Startup

1. Open the Transaction Designer. Existing projects will appear under the **Project Name** header along with their associated **Last Modified** dates.
2. Select the desired project and click Open or double click the desired project name. You will then be taken to the initial Transactions perspective of the TD Workbench.



### Open a Project After Startup

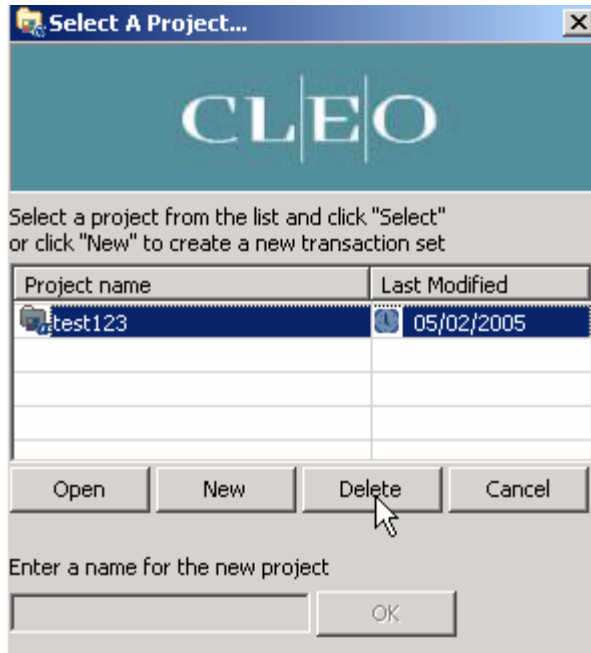
1. In the menu bar choose **File > Open**.
2. A file browser window will appear that displays the default location for your project files. Select the appropriate project you wish to open.

## Delete an Existing Project

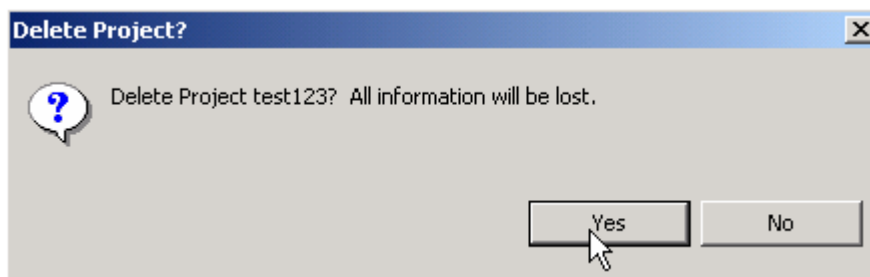
In the event a project is no longer needed. It can be deleted.

To delete a project:

1. Open the Transaction Designer. Existing projects will appear under the **Project Name** header along with their associated **Last Modified** dates.
2. Select the project you wish to delete and click **Delete**.

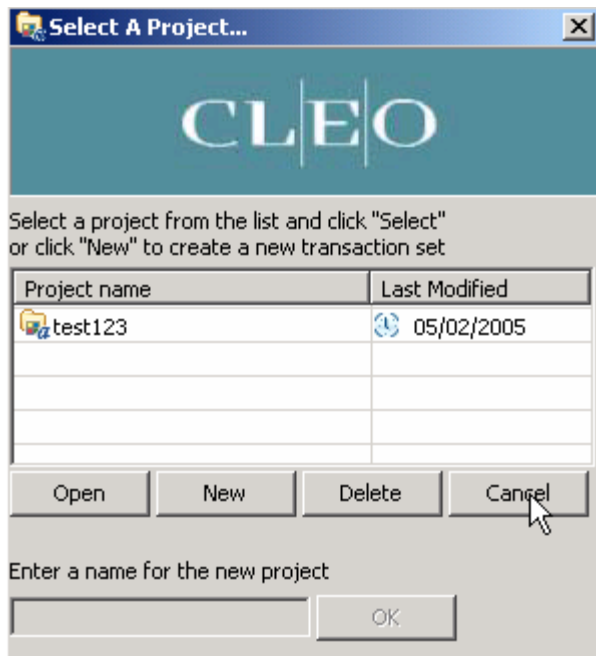


3. You will be asked to confirm your request to delete the project. Click **Yes**.

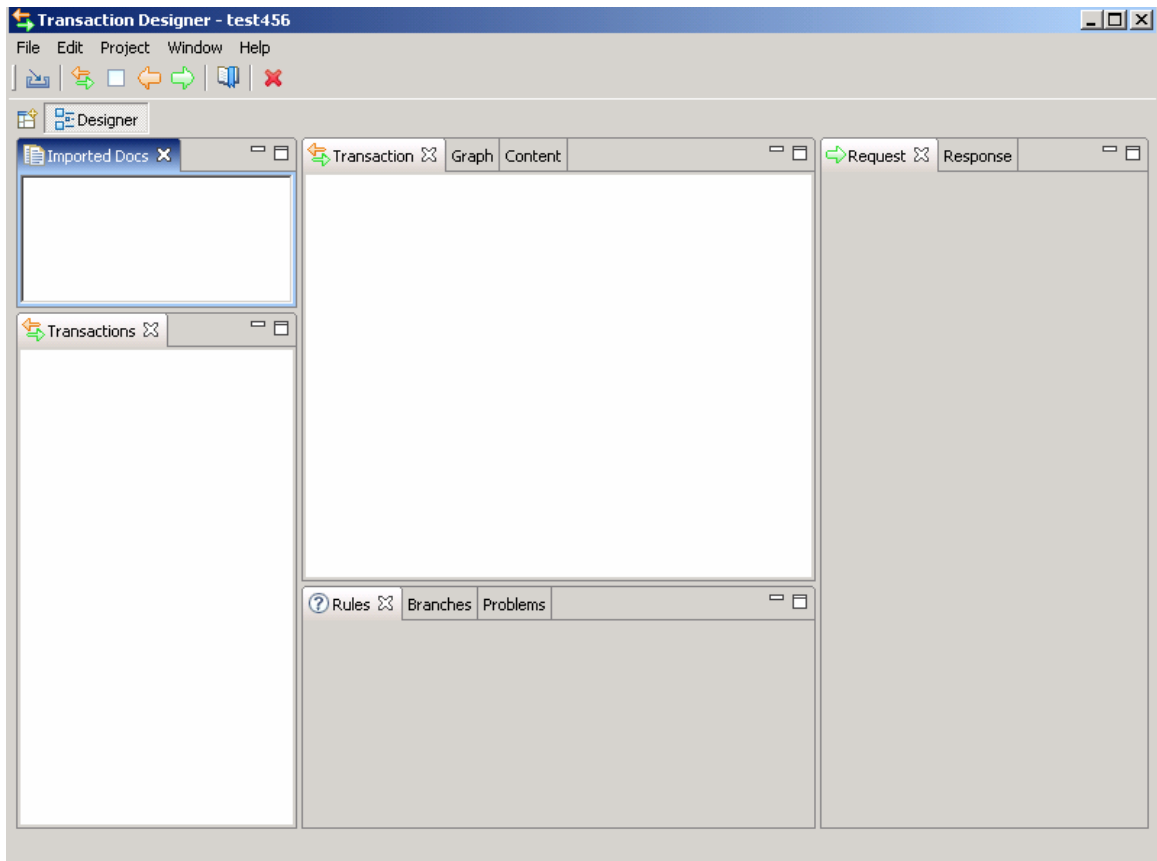


## Cancel


Click **Cancel** to exit the Transaction Designer

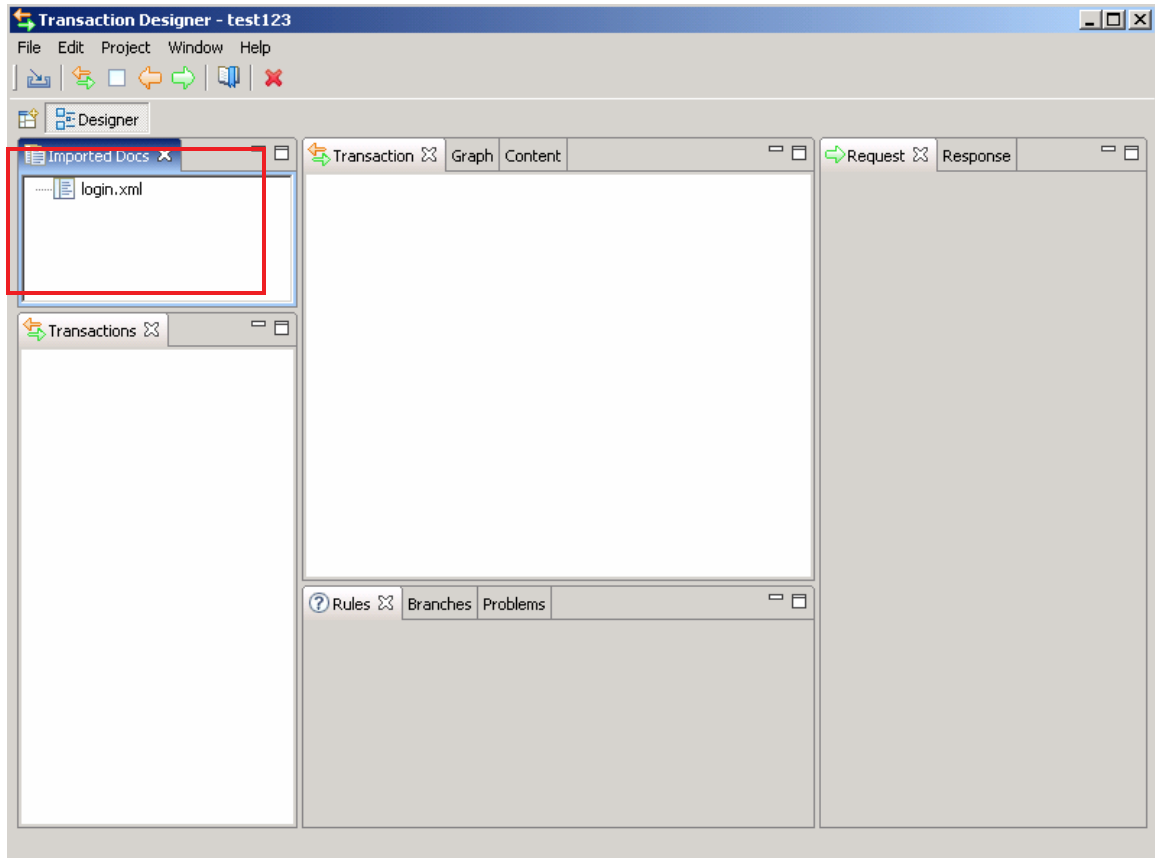


## Import XML Documents



To import an XML document:

1. Click on the Import button: .
2. Browse to find the location of your XML document and click Open. The file will appear in the Imported Docs view.




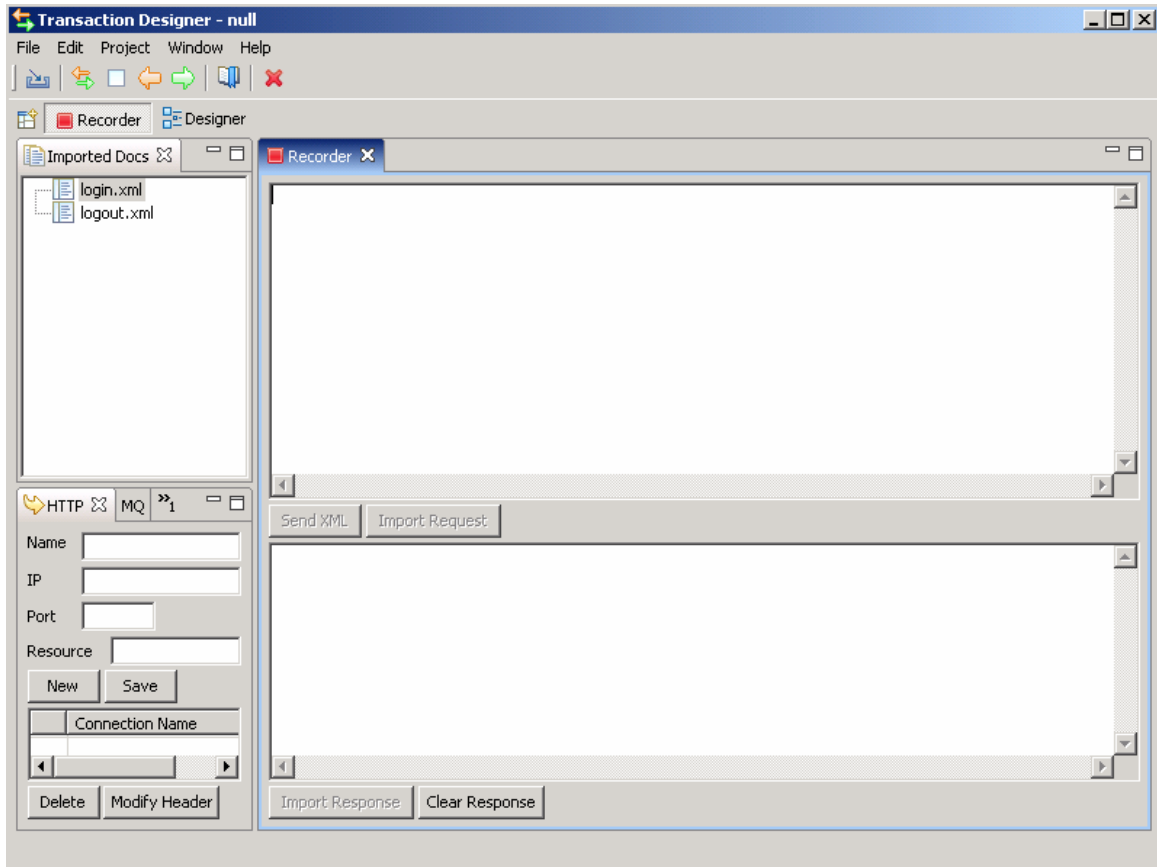
## Record XML Documents

---

If the XML documents exchanged with the server are not available for import (i.e. in file format), the Recorder can be used to save the required XML documents to disk after an interactive session with the server.

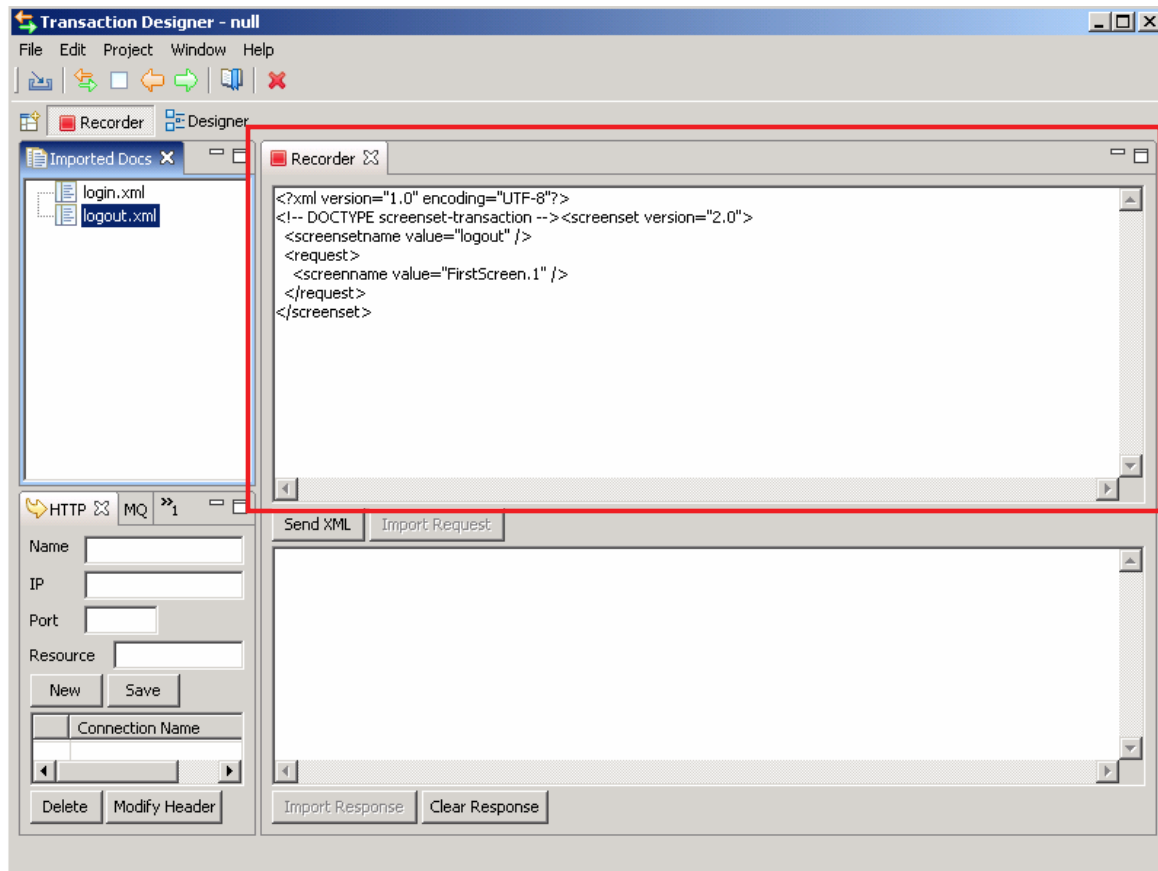
## Access the XML Recorder

To access the XML Recorder, in the toolbar, click on the perspective icon  and choose **Other > Recorder**, or from the menu bar choose, **Window > Open Perspective > Recorder**. This switches the TD to the Recorder perspective.



## Import or Select an XML Document to Send

To begin the recording process, **import** an XML document (see “Import XML Documents”) or, from the Imported Docs view, **select** an XML document that you wish to send to the server. The raw XML will appear in the top text pane of the Recorder view.



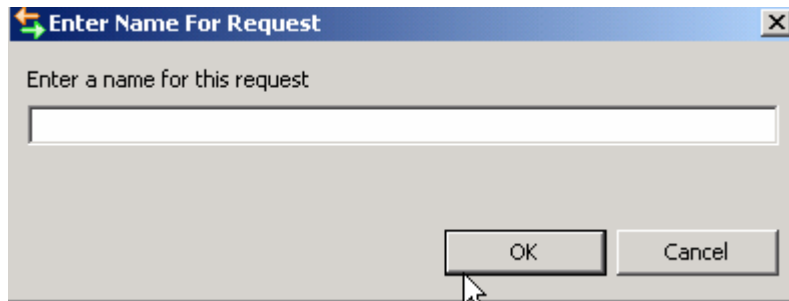
### Optionally Edit the XML

If needed, you may edit the data within the top text pane by either typing directly into the text pane or cutting and pasting text into the text pane.

**Note:** If you make a change to the XML document that renders it not well-formed, such as leaving out an angle bracket, the text will change to red and the Send XML button will become disabled. Fix the error and continue.

## Optionally Import XML Request

1. To save the modified XML as an imported document (for later use in a request), click on **Import Request**.
2. Name the request and click **OK**.



3. The request will appear in the Imported Docs view.

## Create an HTTP Connection

In order to send XML over HTTP to a server, the connection information for the server must be provided. Connection information includes the IP address, Port, and optional Resource (which is the relative location of the application on the server).

To create an HTTP connection:

1. Click on the HTTP view at the lower left, if it is not already displayed.
2. Enter a **Name** for the connection. This name will appear in the "Connection Name" box near the bottom of this panel and is used only as a means for you to distinguish your connections.
3. Enter the **IP address**.
4. Optionally enter the Port.
5. Optionally enter the Resource (e.g. ending URL path).
6. Click **Save**. This saves the connection and will make it available for all projects.

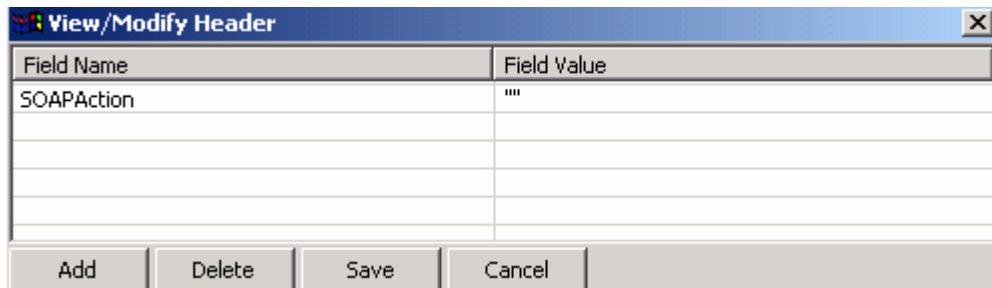
You may continue to create additional connections by clicking **New**. A dialog box will appear asking you to name the connection. The connection will then appear in the selection list. Select the connection to define the settings as above.

## Delete an HTTP Connection

To delete a connection, select the connection name by clicking on it, and then click **Delete**.

## Modify HTTP Header

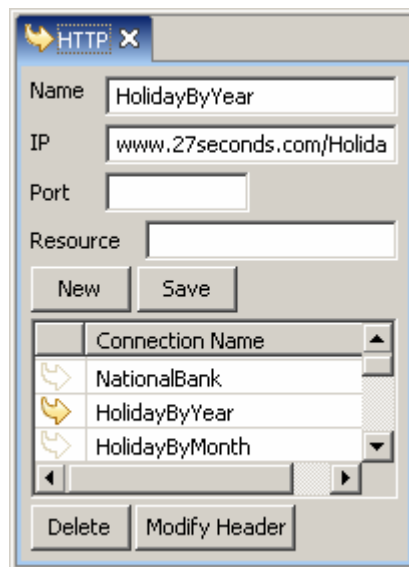
All HTTP connections include a header. Should you need to modify the header, click **Modify Header**. Add the desired field names and field value pairs to the HTTP header. This is an optional tool.



## Send XML to the Server

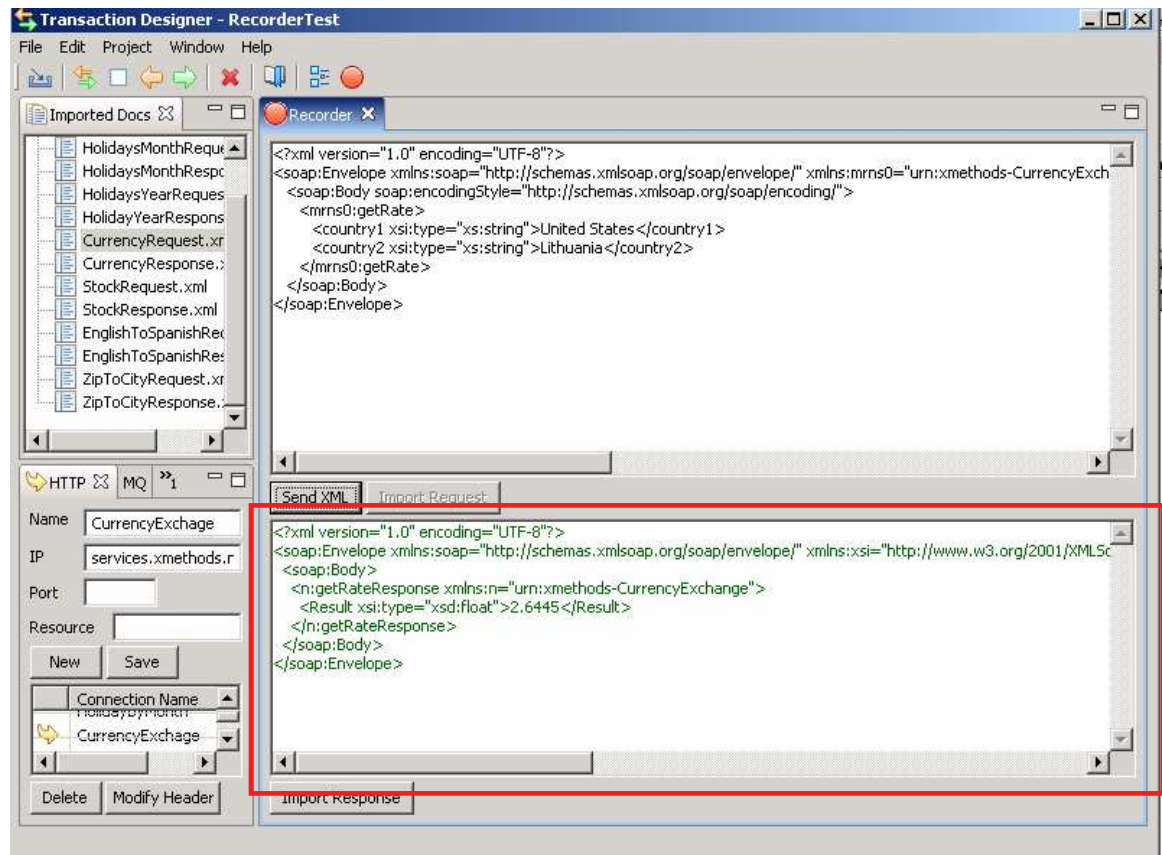
Once the XML document has been selected and your connection has been created, you can send the data to the server.

1. Choose the connection you wish to use by clicking on the connection name from the Connection Name list.



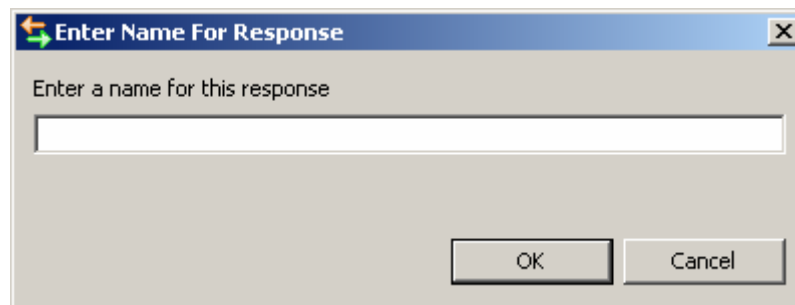
**Note:** Active (selected) connections will be preceded by a yellow arrow. Inactive connections will be preceded by a white arrow.

2. Click the button **Send XML**.
3. The returned response will appear in the bottom text pane.

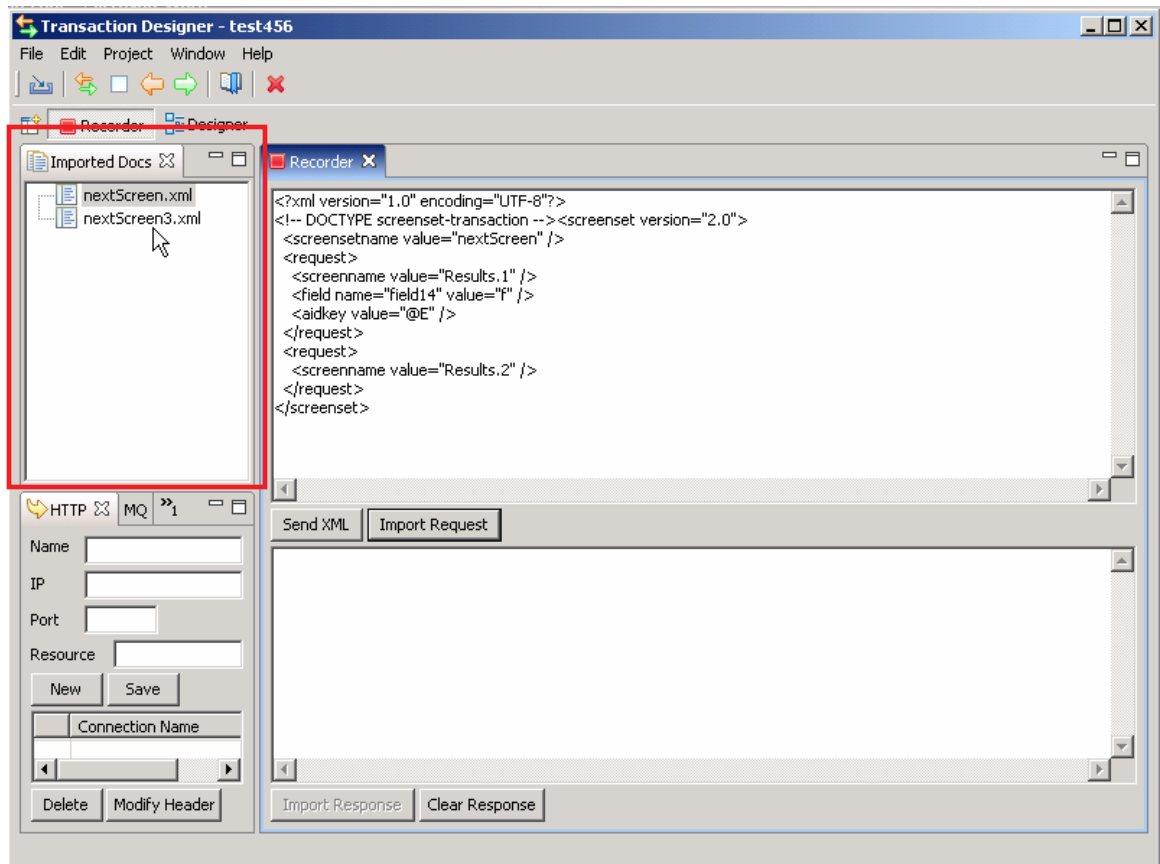



## Import XML Response

1. To save the XML returned by the server as an imported document (for later use in a response), click the **Import Response** button located at the bottom of the screen.
2. Name the response and click **OK**.

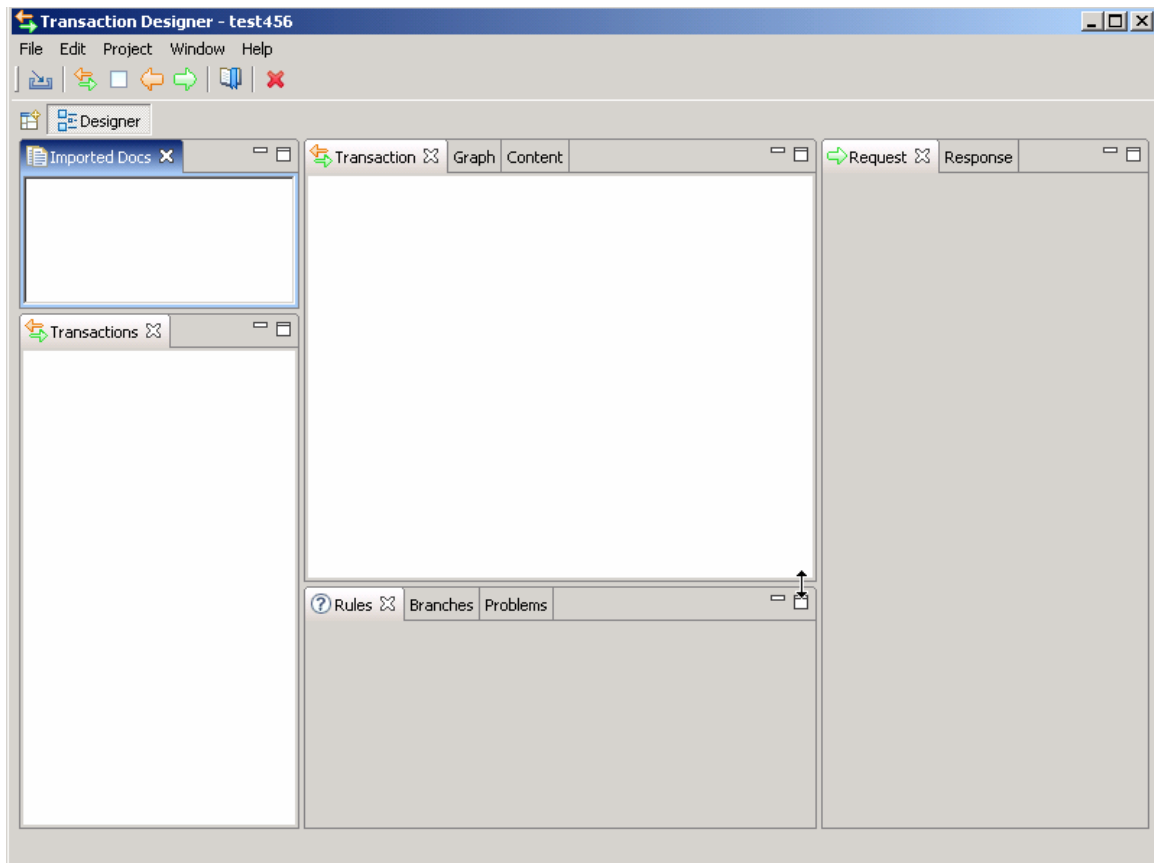



3. The response will appear in the Imported Docs view.

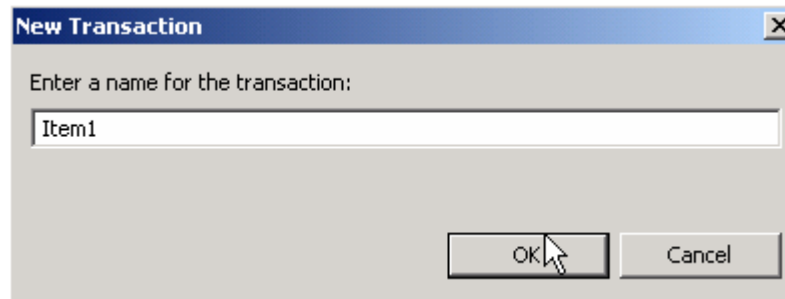


- To switch back to the Transactions view in order to utilize the XML document in your transaction, click on the open a perspective button  and choose Other > Designer or from the menu bar choose, Window > Open Perspective > Transactions.

## Create a New Transaction



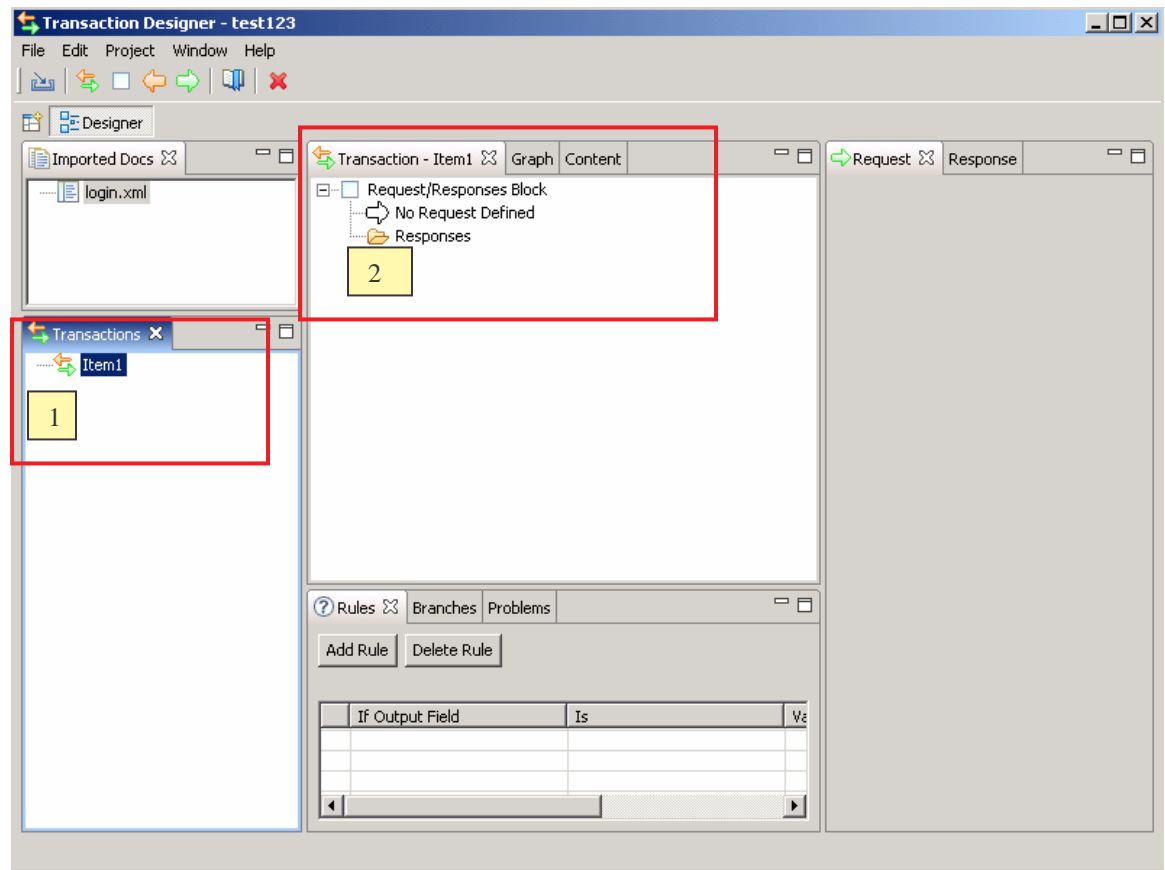
1. In the toolbar menu, click on the New Transaction button:  or in the file menu, choose **Project > New Transaction**.
2. Enter a name for the Transaction and click **OK**.






3. The transaction will appear in the Transactions view.

## Configure a Transaction

When a Transaction is selected, the transaction will appear both in the Transaction view and the Graph view.



The transaction is initially set up with one Request/Responses Block (or simply Block). A Block is a request followed by one or more response. To add a Block to a Transaction, click on the Add Block icon in the toolbar. 

A Transaction is initially set up with one Block containing one Response. To add a Response, select the imported document to be used and click on the Add Response button in the toolbar. . To add a Request, select the imported document to be used and click on the Add Request button in the toolbar. 

**Note:** A Transaction Block *must* contain at least one Request and can also contain one or more Responses.

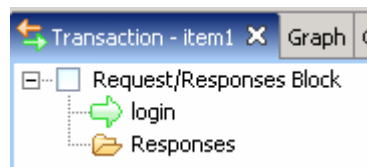
## Define Requests

Requests are the XML documents used to exchange information with the host or server. Defining the Requests uses the element and attribute information contained in the XML or DTD.

In a Request, elements and attributes can be used as *Input Fields*. Input fields are populated by the calling application (which invokes `addInput` on the Transaction Processor) with any data that will be sent to the server application.

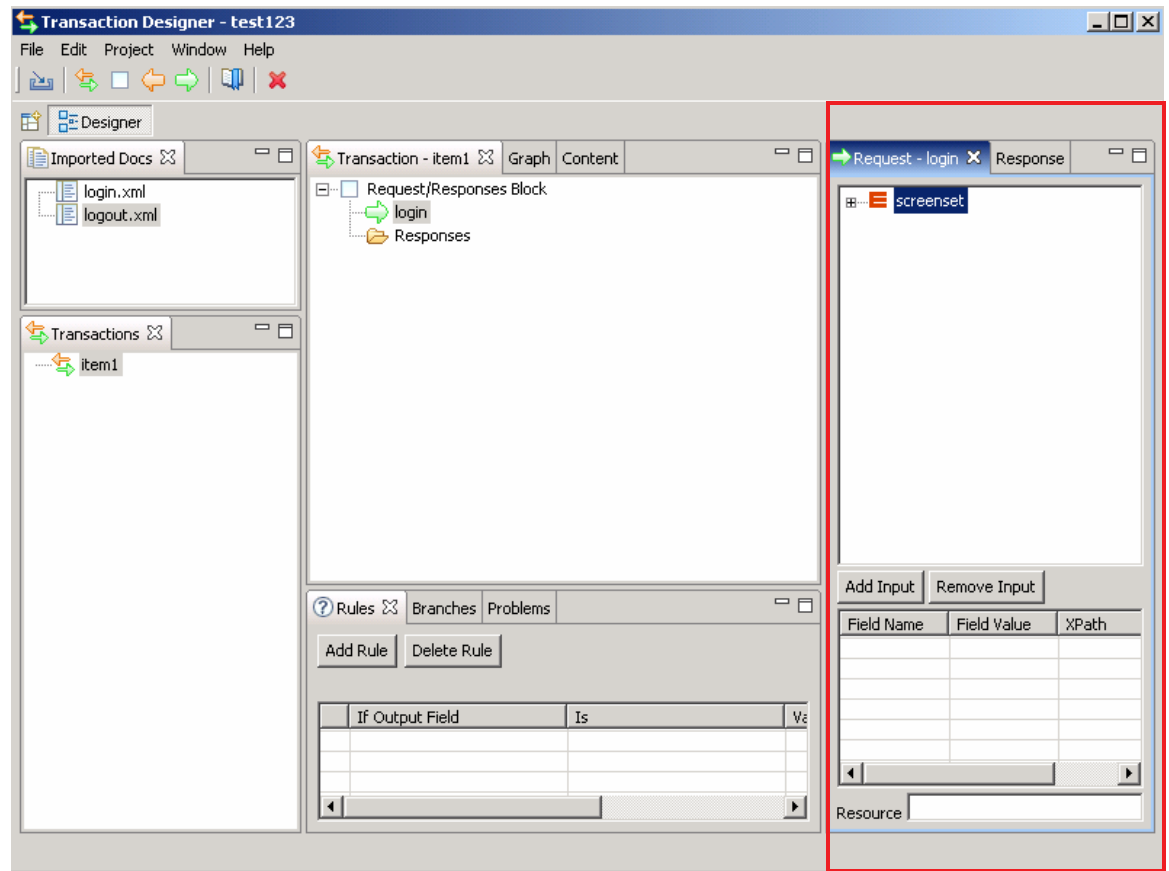
To define a Request, you must first have an XML document imported. See “Import XML Documents” for details.



1. Select an XML document from the Imported Docs view. Click on the desired file and drag it **onto** the Request Arrow.
2. The arrow will change from white to green and the name of the XML document or DTD will appear next to the arrow.

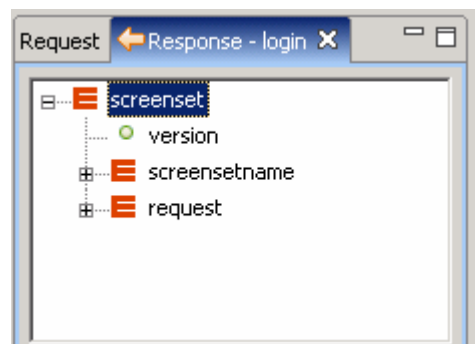


## View a Request

1. In the Request view, you will see the XML root element.



- Click on the + sign to the left of the Element to expand and view the other Elements and Attributes (Sub-Elements). An Element is represented with the  icon and an Attribute is represented with the  icon.



**Note:** A Request must contain at least one Input field. Follow the instructions in the following section to add an input field.

### ***Configure the Resource***

A resource is the relative path (after the IP address and port of a URL) to the host application. Each separate request may be sent to a different resource on the same host.



A new Request will be assigned a default resource (“resource1”).

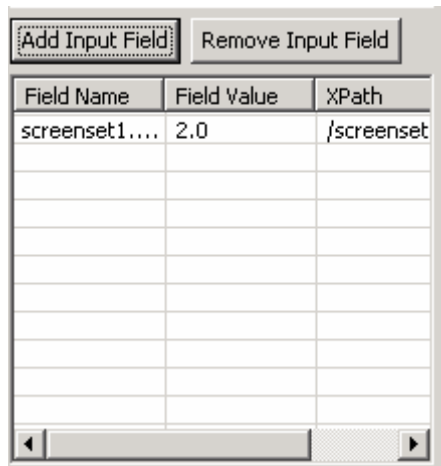
The resource can be edited. However, the value given to the resource must match exactly the resource name as configured in the Transaction Processor.

### ***Add an Input Field***

First you will need to view the request, as described in the View a Request. To designate an Element or Attribute as an Input Field, select the Element or

Attribute in the Request panel and click on **Add Input Field**

The Element or Attribute Field Name, Field Value and XPath will appear in the pane below:



Use the scroll bar at the bottom, to view the entire information.

To remove the Input Field, click on **Remove Input Field**.

## Define Responses

Responses are the XML documents used to exchange information with the host or server. Defining the Responses uses the element and attribute information contained in the XML.

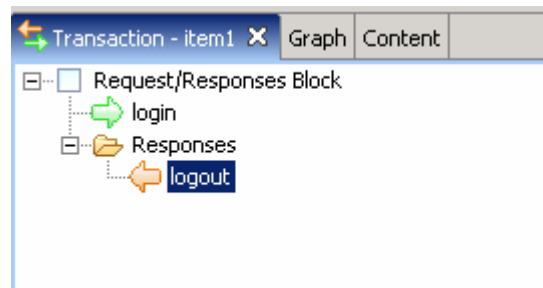
In a Response, elements and attributes can have any of the following functions:

- *Output field*: contains data that will be returned to the calling application (which invokes `getOutput` on the Transaction Processor).
- *Identifier*: used to uniquely identify a particular XML document returned by the server application.

**Note:** An Element or Attribute can be used as both an output field and an identifier.

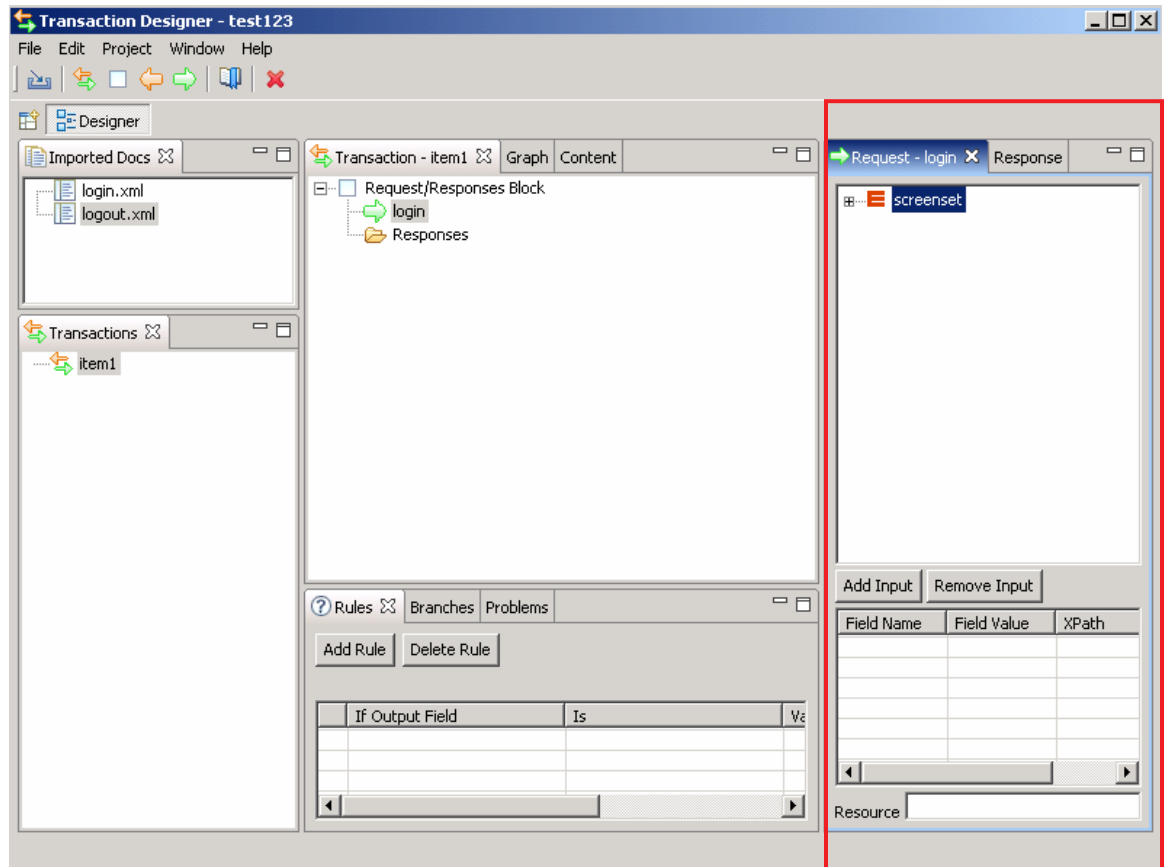
To define the Responses, you must first have an XML document or DTD imported. See “Import XML Documents” for details.

1. To begin defining the output fields and identifiers, you must first select an XML document from the imported docs panel. Click on the desired file and drag it **onto** the Responses folder.
2. An arrow will appear below the folder along with name of the XML document.

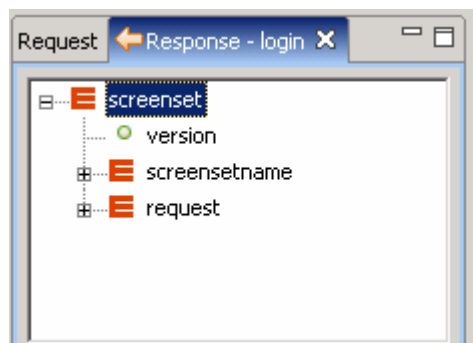


## View a Response

1. In the Response panel, you will see the XML root element.



2. Click on the + sign to the left of the Element to expand and view the other Elements and Sub-Elements. An Element is represented with the (insert) icon and an Attribute, or Sub-element, is represented with the (insert) icon.



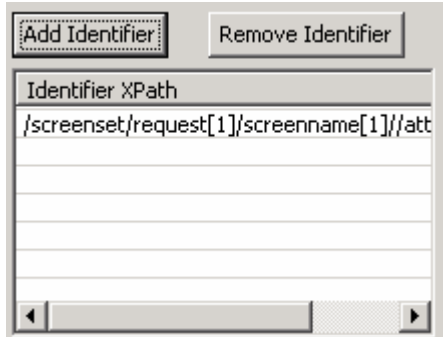
**Note:** A Response must contain at least one Identifier and can contain one or more Output Fields. Follow the instructions in the following sections to Add an Identifier and an Output Field.

***Add an Identifier***

First you will need to view the response, as described in the

View a Response. To designate an Element or Attribute as an Identifier, select the Element or Attribute in the Response panel and click on **Add Identifier**

The XPath for the Element or Attribute will appear in the pane below.



To remove the Identifier, click on **Remove Identifier**.

### ***Add an Output Field***

First you will need to view the response, as described in the

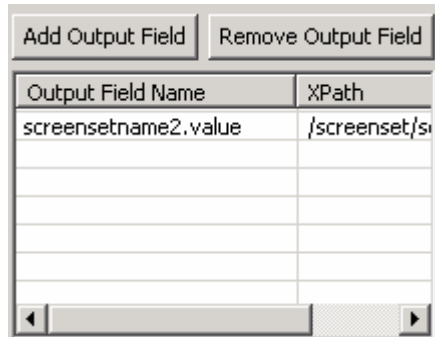
View a Response. To designate an Element or an Attribute as an Output field,

select the Element or Attribute in the Response panel and click on

**Add Output Field**

The Element or Attribute Output Field Name and XPath will appear in the pane

below:











Output Field Name	XPath
screensename2.value	/screensename2.value









Use the scroll bar at the bottom, to view the entire information.

To remove the Output Field, click on **Remove Output Field**.

## Element Icon States

- When an Element has been designated as an Input field, the icon changes from  to .
- When an Element has been designated as an Output field, the icon changes from  to .
- When an Element has been designated as an Identifier, the icon changes from  to .
- When an Element has been designated as an Output field and Identifier, the icon changes from  to .

## Attribute Icon States

- When an Attribute has been designated as an Input field, the icon changes from  to .
- When an Attribute has been designated as an Output field, the icon changes from  to .
- When an Attribute has been designated as an Identifier, the icon changes from  to .
- When an Attribute has been designated as an Output field and Identifier, the icon changes from  to .

**Note:** You can also hover over any Element or Attribute to view the Usage and Value.

## Using Branches

A “branching response” is any response returned from the host or server that will be different depending on what data is originally sent to the server from the user. For instance, suppose there is a single request document that contains only one input field:

- If the user sends INPUTA in the request document, then the response document that is returned from the server will be RESPONSEA.
- If the user sends INPUTB in the request document, then the response document that is returned from the server will be RESPONSEB.

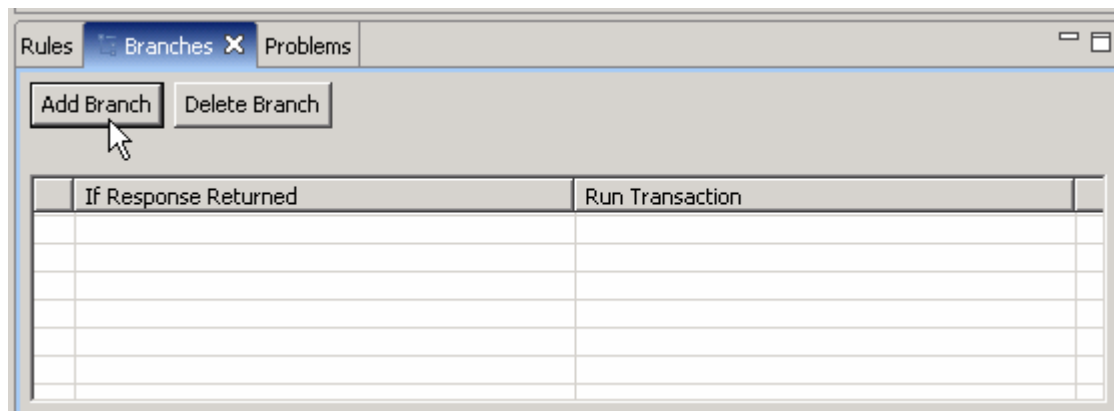
- If the user sends INPUTC in the request document, then the response document that is returned from the server will be RESPONSEC.

In the above case, RESPONSEA, RESPONSEB, and RESPONSEC are uniquely identified from each other.

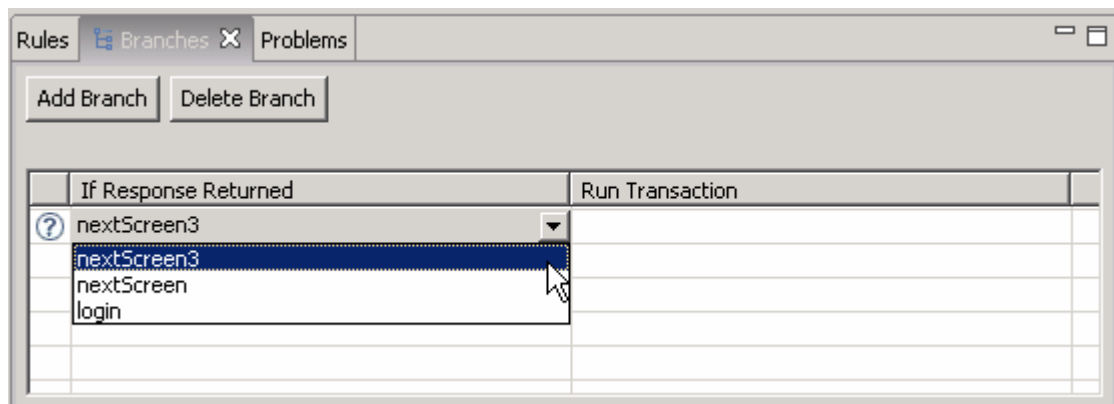
### **Define a Branch**

Before a Branch can be defined, a Transaction containing at least one Response must be selected. Branches cannot be defined when Rules are defined.

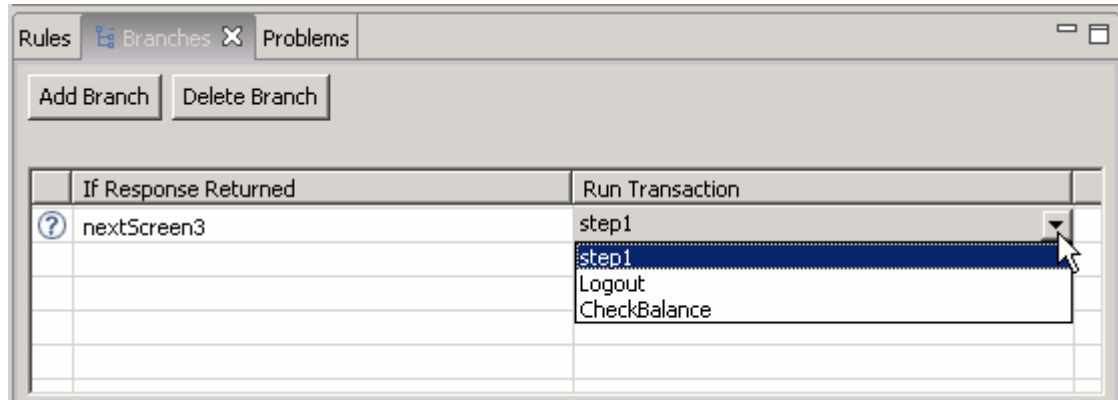
1. Click on **Add Branch** in the Branch Panel:



2. Click in the **first cell** underneath “If Response Returned.”
3. From the drop down box, select the desired Response:



4. Click on the **corresponding cell** underneath “Run Transaction.”
5. From the drop down box, select the Transaction that will be invoked.



## Using Rules

Rules provide conditional control on the transaction. They define which transitions to follow next based on a comparison on Output fields. (Rules are optional, but may be used as an alternative to building similar IF-THEN logic into the user application.) For instance, if there is an Output field that contains a BALANCE and two transitions each leading to two different request nodes (one which lists premium account options and another which lists standard account options), a set of rules might be defined that states:

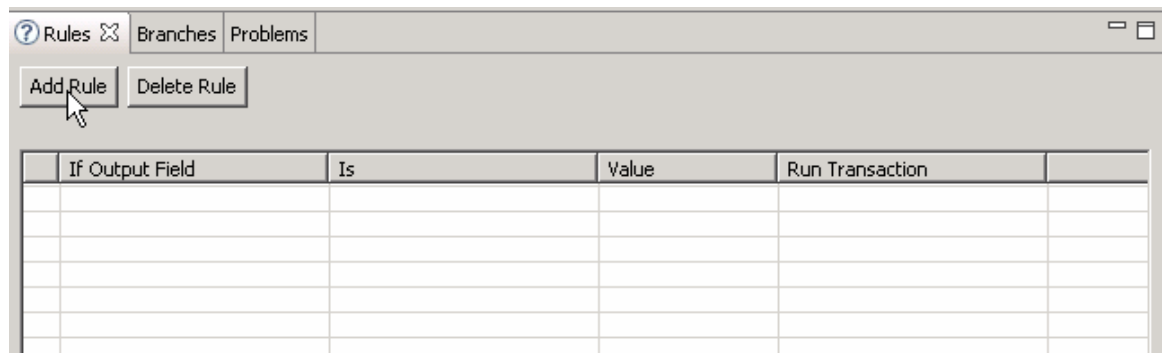
“If BALANCE is greater than or equal to 100000, then follow transition 1”

“If BALANCE is less than 100000, then follow transition 2”

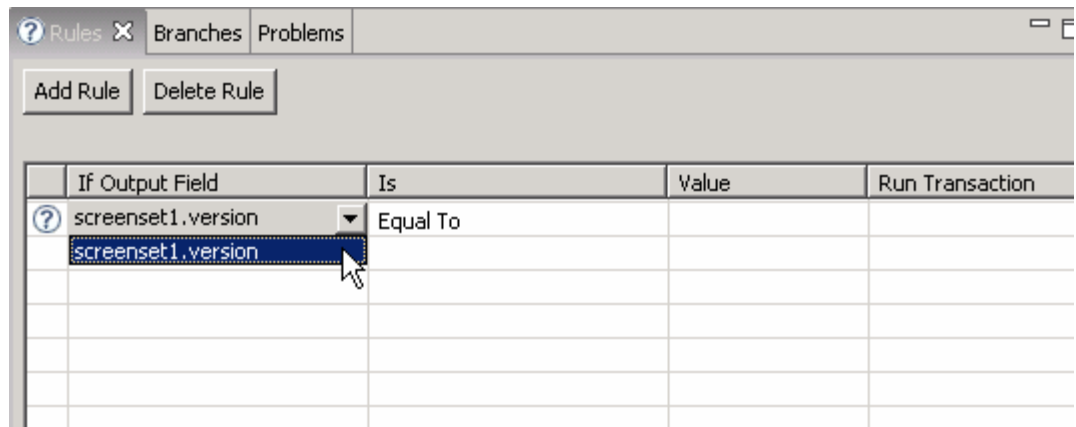
### **Define a Rule**

Before a Rule can be defined, a Transaction containing at least one Output field must be selected. Rules cannot be defined when Branches are defined.

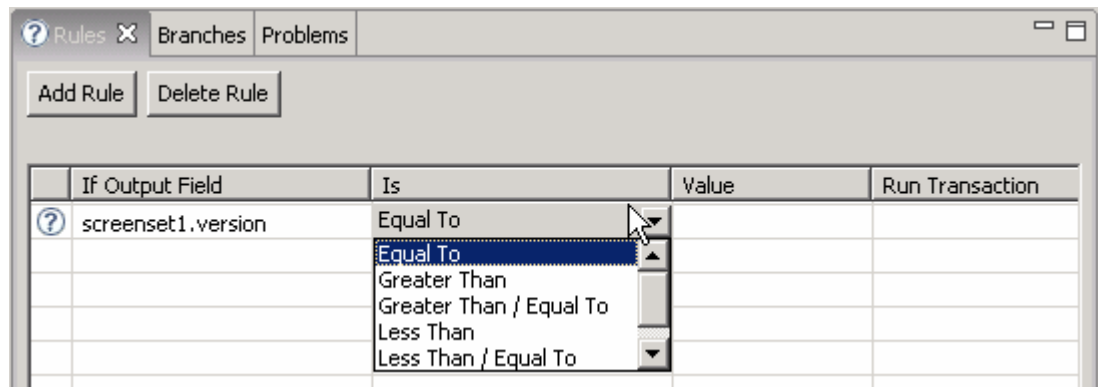
1. Click on **Add Rule** in the Rules Panel.



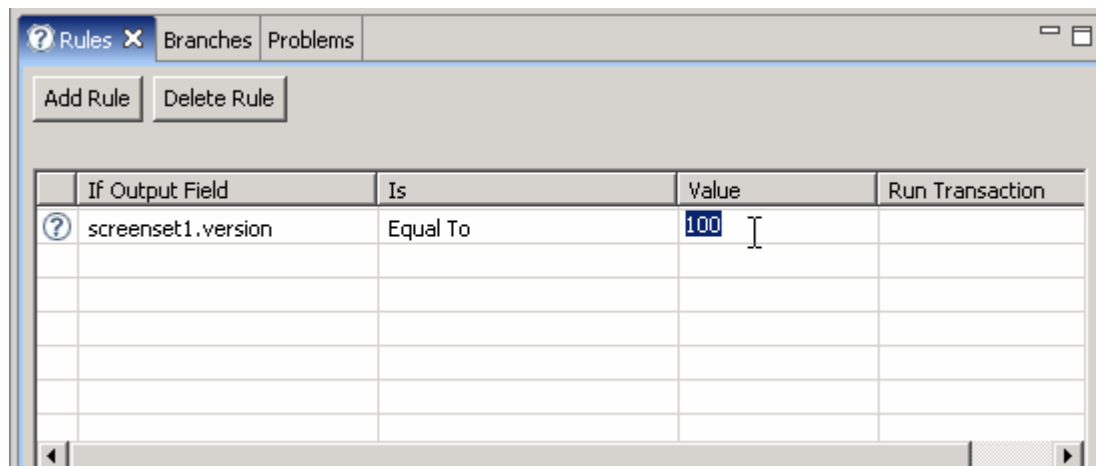
2. Select the desired Output Field.



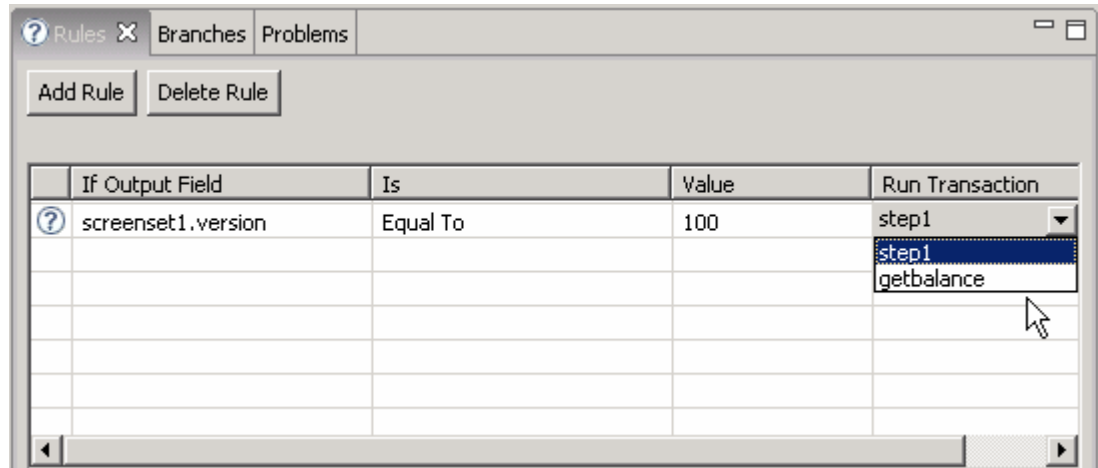
3. Choose the condition.



4. Enter the value.

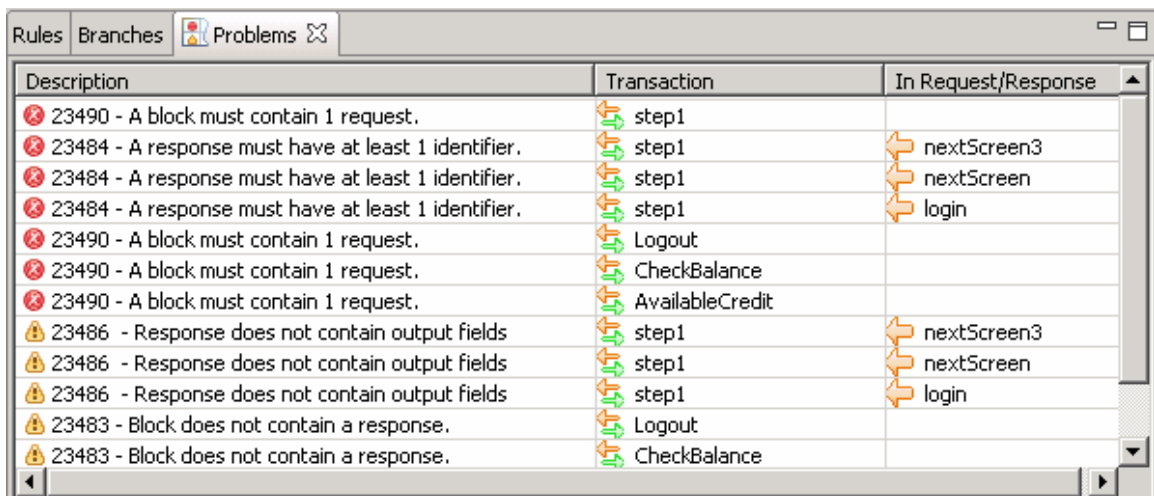


5. Select the Transaction that you wish to run based on the Rule.



## Problems

The problems tab lists any errors in the current Project that will cause it to be invalid, or not able to be published.

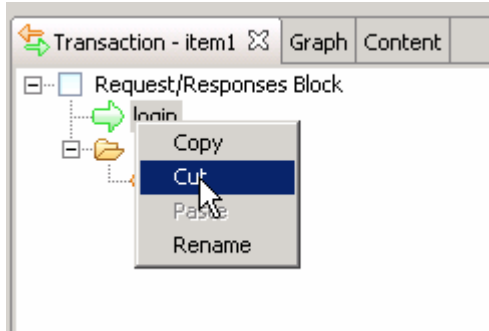


For details on problems and resolution, see “Appendix A: Troubleshooting”.

## Modify an Existing Transaction

### Cut, Copy, and Paste

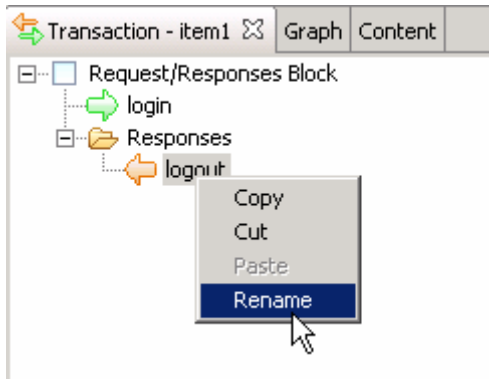
Requests and Responses can be Cut, Copied, and Pasted by right-clicking on the specific Request or Response.



**Note:** Responses cannot be pasted into Requests and Requests cannot be pasted into Responses.

### Rename a Request or Response

Any request or response can be renamed by right-clicking on the specific Request or Response in the Transactions view and choosing Rename.



## Change the XPath

In some cases, it may be desirable to change the XPath for input fields, output fields, or identifiers to something more advanced in order to account for repeating, optional, or moving attributes or elements. Simply click on the XPath cell in a table containing identifiers, output fields or input fields and type in the new XPath.

Output Field Name	XPath
Book20	booksResult[1]/Book[20]

◀ ▶

Add Identifier Remove Identifier

Identifier XPath

/Envelope

◀ ▶

For more information on XPath refer to <http://www.w3.org/TR/xpath>

## Rename input/output fields

In order to rename an input or output field, simply click on the cell containing the name of the field and type in the new name.

Output Field Name	XPath
Book20	/soap:Envelope/soap:...

◀ ▶

Add Identifier Remove Identifier

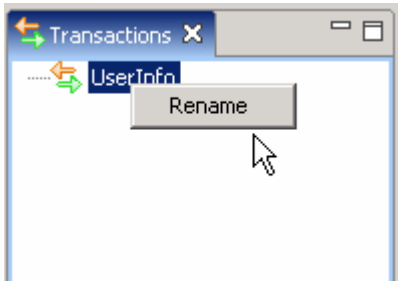
Identifier XPath

/Envelope

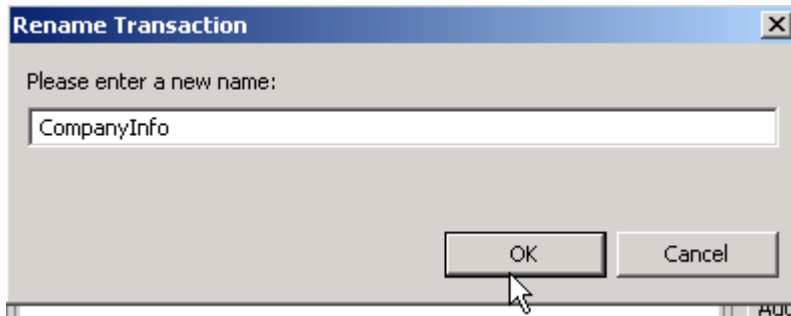
◀ ▶

## Rename a Transaction


To change the name a Transaction, right click on the transaction in the Transaction view and choose **Rename**.



Enter the new name of the Transaction and click **OK**.



## Delete an Existing Transaction

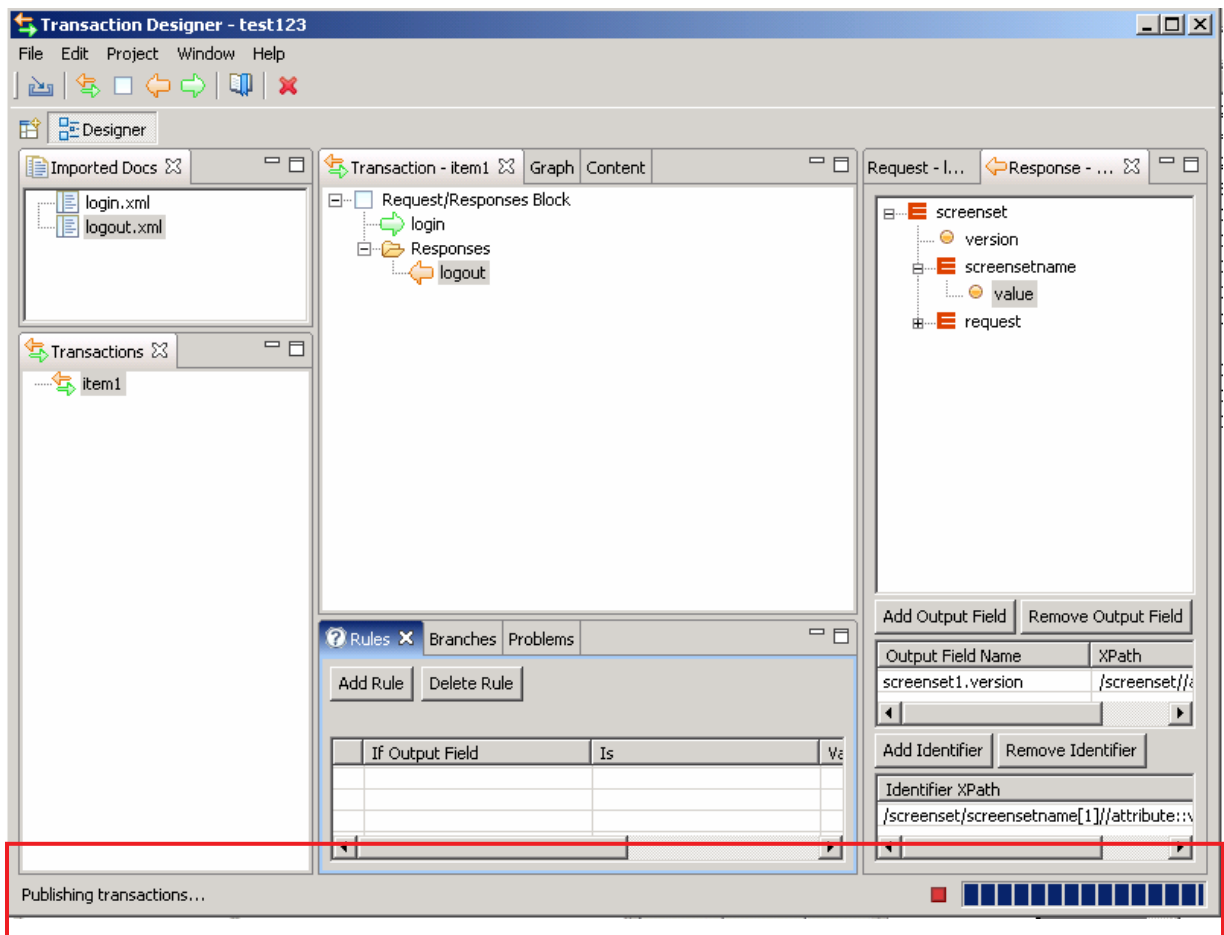
1. Click on the transaction in the Transactions view.
2. Click on the delete button located in the toolbar , or use the menu bar, **Edit** > **Delete**.

## Publish a Transaction

When a transaction is complete, it can be published. The associated transactions are first checked for errors and then the required files used by the Transaction Processor to run the transaction are created. To publish a transaction click on the Publish icon

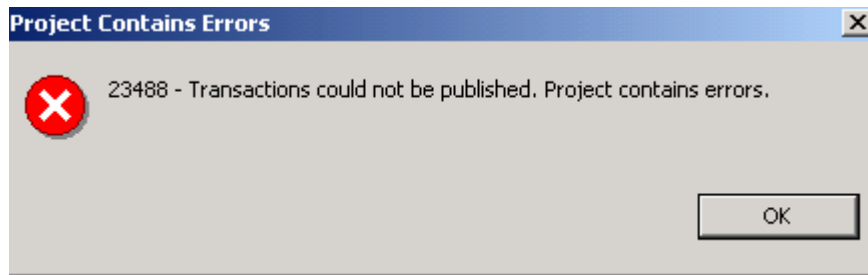
 in the toolbar or use the program menu, **Project > Publish**.

The status bar at the bottom of the workbench will show the publishing status.



Transactions are now ready to be used by the Transaction Processor (see the Transaction Processor Admin Guide and Transaction Processor Programmer's Guide for details on using the Transaction Processor.)

If the transaction contains errors, a warning message will be displayed:



If this occurs, review and fix the errors as identified in the Problems Panel.

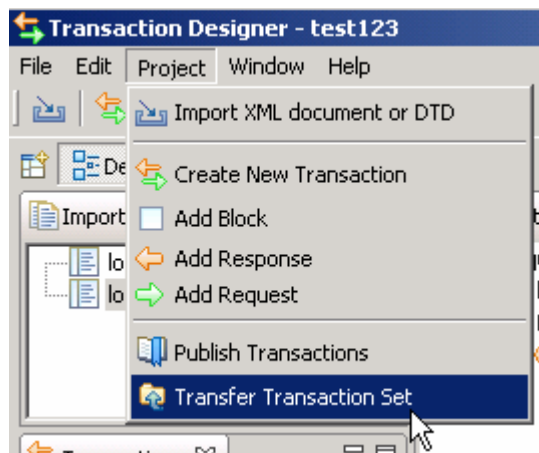
For details on problems and resolution, see “Appendix A: Troubleshooting”.

## Transfer the Published Transaction Set

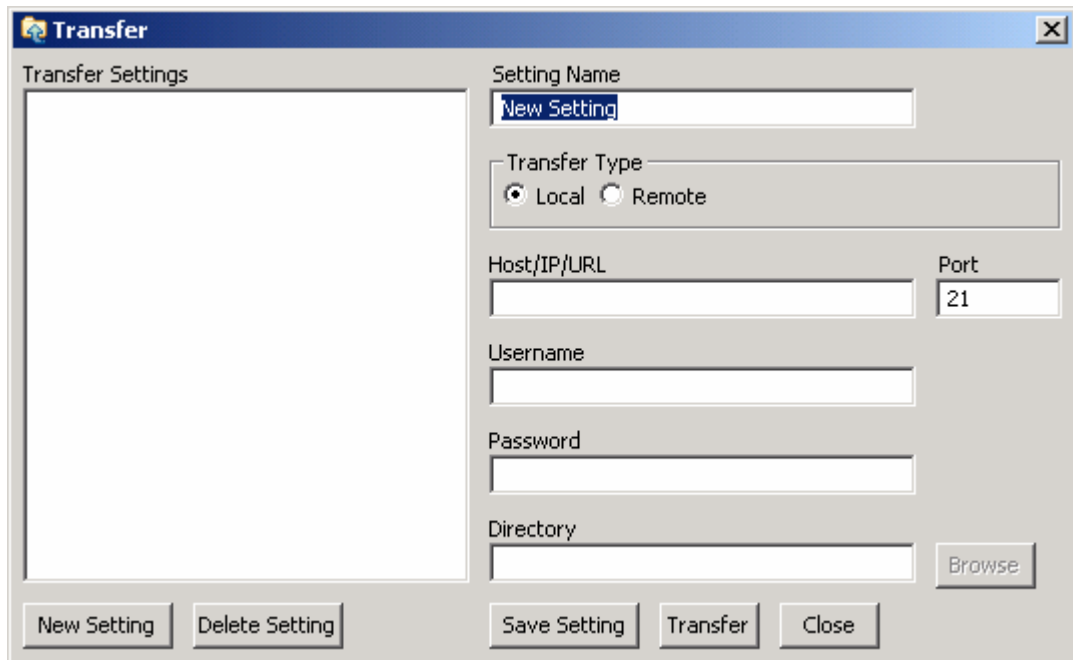
---

Once the Transaction Set has been published, the TD can be used to copy the files required by the Transaction Processor into their runtime locations.

To transfer the published transaction set to either a local network drive or a remote location, from the file menu, choose **Project > Transfer Transaction Set**.



A new window will open. The first time you utilize the Transfer feature, you will need to name and define the transfer settings. In the future, you will be able to utilize saved transfer settings.



1. Enter a **name** for the transfer setting.
2. Select either **Local** or **Remote** for the Transfer Type.
3. If Local, type in or Browse to find the directory path.
4. If Remote, enter the Host/IP/URL, Port, Username, and password.
5. Click **Save Setting**. The saved setting will now appear in the left side of the window under 'Transfer Settings.'
6. Click on the Transfer Setting you wish to use and click **Transfer**.

To Delete a Transfer Setting, click on the setting to be deleted and click **Delete Setting**. To create an additional transfer setting, click **New Setting**. To exit the window, click **Close**.

## Pulling It All Together

Now that you have read about all the different features provided by the Transaction Designer for XML, you should be ready to create your first Transaction Set project.

In this section, we have provided a simple example that should help pull much of what you've learned together. For this example, you will be creating a transaction that will obtain the state for a given ZIP code.

To create a Transaction Set project:

1. If the TD is not licensed, install a license (see “License the Transaction Designer”).
2. Launch the TD for XML.
3. Enter a name for the new project – ZipService (see “Create a New Project During Startup”).
4. Switch to the Recorder Perspective (see “Access the XML Recorder”).
5. Create a new HTTP connection (see “Create an HTTP Connection”).
  - a. Name: ZipService
  - b. IP: <http://webservices.imacination.com/distance/Distance.jws>
6. Modify Header for ZipService (see “Modify HTTP Header”).
  - a. Content-Type = application/soap+xml; charset=utf-8
  - b. SOAPAction = “”
7. Record the XML documents that will be used in the transaction.
  - a. Send the following XML request to the configured Web Service (see “Send XML to the Server”):

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body>
  <ns1:getState soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
xmlns:ns1="http://DefaultNamespace">
    <ns1:zip>48103</ns1:zip>
  </ns1:getState>
</soapenv:Body>
</soapenv:Envelope>
```

- b. The following XML response should be returned:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body>
  <ns1:getStateResponse xmlns:ns1="http://DefaultNamespace"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <ns1:getStateReturn xsi:type="xsd:string">MI</ns1:getStateReturn>
  </ns1:getStateResponse>
</soapenv:Body>
</soapenv:Envelope>
```

- c. Import the request (see “Optionally Import XML Request”). Name the request “GetStateRequest”.
- d. Import the response (see “Import XML Response”). Name the response “GetStateResponse”.

- e. Clear the response.
8. Switch to the Designer Perspective.
9. Create a new transaction (see “Create a New Transaction”). Name the transaction “GetState”.
10. Define the Request and Response that will be exchanged with the Web Service when the ZipTrans transaction is run.
  - a. Add a request to the first Request/Responses Block using the GetStateRequest imported doc (see “Define Requests”). Rename request to “GetStateRequest” (see “Rename a Request or Response”).
  - b. Add a response to the first Request/Responses Block using the GetStateResponse imported doc (see “Define Responses”). Rename the response to “GetStateResponse”.
11. Define Input Fields that will be sent to the Web Service, Output Fields that will be returned from the Web Service, and Identifiers that will be used to verify that the correct response is returned from the Web Service.
  - a. Add an input field to GetStateRequest using the “ns1:zip” element (see “Add an Input Field”). Rename the input field to “ZipCode”.
  - b. Add an output field to GetStateResponse using the “ns1:getStateReturn” element (see “Add an Output Field”). Rename the output field to “State”.
  - c. Add an identifier to GetStateResponse using the “ns1:getStateResponse” element (see “Add an Identifier”).
12. Publish the transaction set (see “Publish a Transaction”).
13. Transfer the transaction set (see “Transfer the Published Transaction Set”).
14. View the transaction map (see “Appendix B: Transaction Map”).

The “GetState” transaction can now be run in the Transaction Processor.

# Appendix A: Troubleshooting

## Problem Descriptions

---

The following warnings (that may or may not need to be corrected) and errors (that prevent transactions from being published) are documented in the Problems view.

Description	Transac
23490 - A block must contain 1 request.	New1
23483 - Block does not contain a response.	New1

### **23477- A transaction must be selected before adding a block.**

Simply click on the transaction you wish to add a block to, then add the block.

### **23478- Cannot add block, conditions are defined.**

Since conditions such as Branches or Rules can only exist on the last block in a transaction, blocks cannot be added to a transaction after conditions have been defined. If you wish to add additional blocks to this transaction, you must first delete the condition.

### **23479- Both an imported doc and a block must be selected before adding a request.**

Click on the imported doc to be used as a request and then click the add request button on the toolbar, or drag the document over to the request area of the desired transaction.

### **23480- Both an imported doc and a block must be selected before adding a response.**

Click on the imported doc to be used as a response and then click the add response button on the toolbar, or drag the document over to the responses folder of the desired transaction.

**23481- Failed to import document due to the following error**

Generally caused by attempting to import a file that is not well-formed XML

**23482- Only one imported document can be added as a request.**

A transaction can only have one request.

**23483- Block does not contain a response.**

A valid transaction must contain at least one response.

**23484- A response must have at least 1 identifier.**

A valid response must contain at least one identifier to distinguish it from other responses.

**23485- Request does not contain input fields.**

In most cases, but not all, requests have at least one input field to be sent to the host/server application. It is not necessary to provide an input field, but this is flagged as a warning since in most cases it is common for a request to have one.

**23486- Response does not contain output fields**

In most cases, but not all, responses have at least one output field specifying fields to be extracted from a document received from a host/server application. It is not necessary to provide an output field, but this is flagged as a warning since in most cases it is common for a response to have one.

**23484- This project does not contain any transactions.**

In order to publish transactions, there must be transactions to publish. See section on creating transactions.

**23488- The following transactions contain errors and will not be published...**

If a transaction contains errors, it will not be published. Check the problems view for a list of errors for more details.

**23489- There are warnings in the project. Continue with publish?**

Transactions that contain warnings are still publishable, however you may want to check the problems view for more details on the warnings.

**23490- A block must contain 1 request.**

A block must contain at the very least a request to be sent to the host/server application.

## Appendix B: Transaction Map

A file containing a list (i.e. “mapping”) of all transactions and request/response definitions currently defined in the Transaction Set project is published each time the Transaction Set is published. This file can be used by the developer of the application to determine which transactions to invoke on the Transaction Processor and what input to send to/output to retrieve.

To view the Transaction Map, select Window | Show View > Transaction Map. The Transaction Map can be printed by right-clicking anywhere in the text area and selecting Print.

### Sample Transaction Map

#### Transaction Set: XMLSample

<b>Transaction: GetCity</b> 0 Error(s) 0 Warning(s)	
<i>- Request/Responses Block 1 -</i>	
<i>Request = GetCityByZipRequest</i> <i>Resource = resource1</i>	
<b>Input Field - zip</b>	
<i>Value</i>	48103
<i>XPath</i>	/*[local-name()='Envelope']/*[local-name()='Body']][1]/*[local-name()='getCity']][1]/*[local-name()='zip']][1]
<i>Response = GetCityByZipResponse</i>	
<b>Output Field - city</b>	
<i>XPath</i>	/*[local-name()='Envelope']/*[local-name()='Body']][1]/*[local-name()='getCityResponse']][1]/*[local-name()='getCityReturn']][1]
<b>Identifier</b>	
<i>XPath</i>	/*[local-name()='Envelope']/*[local-name()='Body']][1]/*[local-name()='getCityResponse']][1]
<i>- End of Request/Responses Block 1 -</i>	

# Glossary

**Attribute:** In an XML document, a sub element defined within an element. In the following example, GENDER and AGE are attributes within the PERSON element:

```
<person gender="male" age="36">  
  <firstName>Bob</firstName>  
  <lastName>White</lastName>  
</person>
```

**Element:** A logical structure in an XML document that is delimited by a start and an end tag. Also called an XML object.

**Identifier:** Used only in Responses, used to uniquely identify a particular XML document returned by the server application.

**Input field:** Used only in Requests, populated by the calling application (which invokes addInput on the Transaction Processor) with any data that will be sent to the server application.

**Keyboard macro:** variable text stored in a table to be used as input in a host transaction sequence. Each entry is referenced by column number when defining “to host” field definitions.

**Output field:** Used only in Responses, contains data that will be returned to the calling application (which invokes getOutput on the Transaction Processor).

**Partial Publish:** When using the publish function to create the XML files for screen definitions and transactions, the XML file could not be generated for any transaction containing screens with no identifiers.

**Right click:** the action of using the alternate mouse button.

**Transaction Set:** a collection of transactions and XML documents used by the Transaction Processor. A transaction set may be assigned to one or more sessions. For example if “tset1” is assigned to sessions 2-11, “tset2” is assigned to sessions 12-31, the IR application “ir1” which runs “tset1” is assigned to channels 0-9, the IR application “ir2” which runs “tset2” is assigned to channels 10-19, and the IR application “ir3” which also runs “tset2” is assigned to channels 20-29, then when a call is made into channel 1, “tset1” will be run. Likewise, if a call is made into channel 11 or channel 25, “tset2” will be run. (See also Application)

**User-defined transaction:** contains host transaction sequence that is run by the Transaction Processor when requested by the IR application to retrieve information from the host. An empty user-defined transaction named “transaction1” is created by the Transaction Designer.

**XML:** short for *Extensible Markup Language*, a specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents.

**XPath:** a language that describes how to locate specific content contained within an XML document. XPATH treats an XML document as a logical ordered tree.

# Index

- Add**
  - Branch ..... 43
  - Identifier ..... 40
  - Input Field ..... 36
  - Output Field ..... 41
  - Rule ..... 44
- Administration GUI ..... 5
- Attribute
  - Definition ..... 59
- Attribute Icon States ..... 42
- Branches
  - Add ..... 43
  - Explanation of ..... 42
- Component Diagram ..... 8
- Create
  - New Project ..... 20
  - New Transaction ..... 32
- Define
  - Input Fields ..... 37
  - Output Fields ..... 37
- Delete
  - Existing Project ..... 22
  - Transaction ..... 49
- Element
  - Definition ..... 59
- Element Icon States ..... 42
- HTTP
  - Connection ..... 28
  - Header ..... 29
- Identifier
  - Assign ..... 39
  - Definition ..... 37, 59
- Import
  - Request ..... 28
  - XML Documents ..... 24
- Input field
  - Assign ..... 36
  - Definition ..... 34, 59
- Keyboard macro
  - Definition ..... 59
- License Key ..... 19
- Licensing ..... 19
- menu bar ..... 15
- Open
  - Existing Project ..... 20
- Output field
  - Assign ..... 40
  - Definition ..... 37, 59
- Partial Publish
  - Definition ..... 59
- perspective ..... 12
  - Recorder ..... 14
- Perspective Bar ..... 18
- Publish
  - Transaction ..... 50
- Recorder ..... 26
- Remove Output Field** ..... 41
- Rename
  - Request or Response ..... 47
- Requests ..... 34
- Responses ..... 37
- Right click**
  - Definition** ..... 60
- Rules
  - Add ..... 44
  - Explanation of ..... 44
- Transaction
  - Components of ..... 33
  - Delete ..... 49
  - Modify ..... 47
  - Publish ..... 50
  - Rename ..... 49
- Transaction Designer ..... 5, 6
  - Key features ..... 10
  - License ..... 19
  - Run ..... 20
- Transaction Processor ..... 5, 50
- Transaction Set
  - Definition ..... 60
- Transfer
  - Local ..... 52
  - Remote ..... 52
  - Settings ..... 51
  - Transaction Set ..... 51
- User-defined transaction
  - Definition ..... 60
- view ..... 11
  - Branches ..... 13

Content.....	13	workbench.....	11
Graph.....	13	Menu Bar .....	12
Imported Docs.....	13	Perspectives Bar.....	13
Problems .....	13	Toolbar .....	13
Request.....	13	XML	
Response .....	13	Definition .....	60
Rules .....	13	Edit.....	27
Transaction.....	13	XML Recorder .....	26
Transactions .....	13	XPath.....	60