

IBM WebSphere® process integration V6.0.2 – Lab exercise

ManualClean – Human tasks

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	3
Description of the Module.....	3
Business objects.....	4
Interfaces.....	4
Exercise instructions	5
Part 1: Initialize the workspace for this lab exercise and create a module.....	7
Part 2: Create manual cleanse	12
Part 3: Prepare for unit testing and assemble manual cleanse	20
Part 4: Test manual cleanse	23
What you did in this exercise	32
Solution instructions	33
Task: Adding remote server to WebSphere Integration Developer test environment	34

What this exercise is about

In many situations focused on the automation of integration between different technologies, there is still a need for human interaction. WebSphere Integration Developer provides developers with the ability to easily add human interaction into integration applications. Utilizing the Service Component Architecture (SCA) programming model, human tasks can be defined as services (SCA components) and then flexibly wired into integration applications. This lab exercise will show the basic steps for defining a human task as a service component and testing the service using the End-to-End Test Framework.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Integration Developer V6.0.2 installed
- WebSphere Process Server V6.0 test environment installed
- Sample code in the directory C:\Labfiles602 (Windows®) or /tmp/LabFiles602 (Linux®)

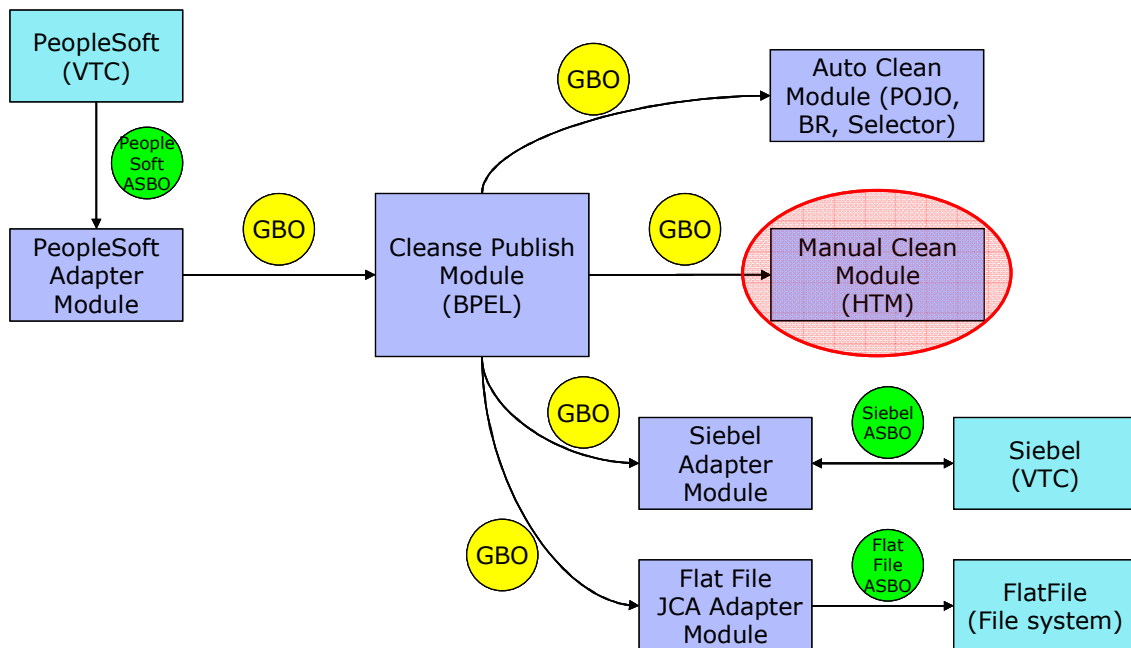
What you should be able to do

At the end of this lab you should be able to:

- Create a Participating Task
- Assemble the Participating Task following the SCA programming model
- Component test the Participating Task using the End-to-End Test Framework

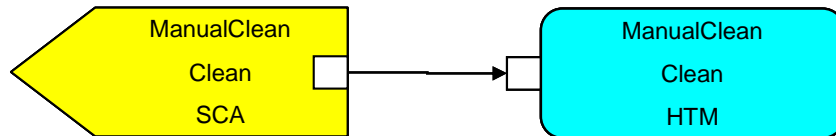
Introduction

The following diagram highlights the part of the overall scenario that will be addressed in this lab. You will create the manual clean human task which utilizes the clean interface. The human task can then be assembled as an SCA Component. The SCA component can then be easily wired as a step in the Cleanse Publish business process. At this step, the human task allows for the general business object (GBO) which was passed into the business process from the PeopleSoft Adapter to be manually updated or cleaned after an automated clean has occurred from a Business Rule, Selector, or Java™ object. After the GBO is manually cleaned and the task is completed, the business process can continue and publish the GBO to the Siebel and Flat File adapters. Because the human task is wired in as a SCA component, at any time the implementation can be changed to any type that matches the clean interface. This provides flexibility to the integration application as the business scenario changes.



Description of the module

The following diagram shows the ManualClean module. It highlights the parts of the module that will be addressed in this exercise.



Business objects

The following generic business object (GBOs) will be used in this exercise.

- **Clip** – The parent business object describing characteristics of a particular clip. This generic business object originates from the application specific business object (ASBO) from the PeopleSoft system. In the complete solution, a map and mediation is used to transform the application specific business object from the PeopleSoft system to the generic business object which the Manual Clean human task can operate upon.
- **ClipItem** - The child business object describing a retail packing option for this type of clip. This generic business object is also defined from the application specific business object (ASBO) from the PeopleSoft system and transformed to the generic business object using a map and mediation.

In addition to these business objects, there will be a business graph generated to contain these business objects.

- **ClipBG**

Although this business graph is generated by the tools, it is only associated with the parent business object. The business graph for the child business object is not used in the scenario.

Interfaces

The following interface is generated by WebSphere Integration Developer and uses the business graph and business objects.

- **Clean** – Request/Response interface which passes a single part, named inClipBG of type ClipBG, on the request and a single part on the response, named outClipBG of type ClipBG. With the use of this interface, it is expected that any changes made to the ClipBG will be passed directly back. This interface will be used to define the ManualClean human task and will be part of the exported SCA binding which is called by other SCA components

Exercise instructions

Some instructions in this lab might be specific for Windows platforms. If you run the lab on a platform other than Windows, you will need to run the appropriate commands, and use appropriate files (for example .sh in place of .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references as follows:

Reference variable	Windows location	Linux location
<WID_HOME>	C:\Program Files\IBM\WebSphere\ID\6.0	/opt/IBM/WebSphere/ID/6.0
<WPS_HOME>	<WID_HOME>\runtimes\bi_v6	<WID_HOME>/runtimes/bi_v6
<LAB_FILES>	C:\Labfiles602	/tmp/Labfiles602
<WORKSPACE>	C:\Labfiles602\eXchange\ManualClean\workspace	/tmp/Labfiles602/eXchange/ManualClean/workspace
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles602\eXchange\ManualClean\Solution	/tmp/Labfiles602/eXchange/ManualClean/Solution

Windows users' note: When directory locations are passed as parameters to a Java program such as wsadmin, you must replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles602\ would be replaced by C:/LabFiles602/.

Note that the previous table is relative to where you are running WebSphere Integration Developer. This table is related to where you are running remote test environment:

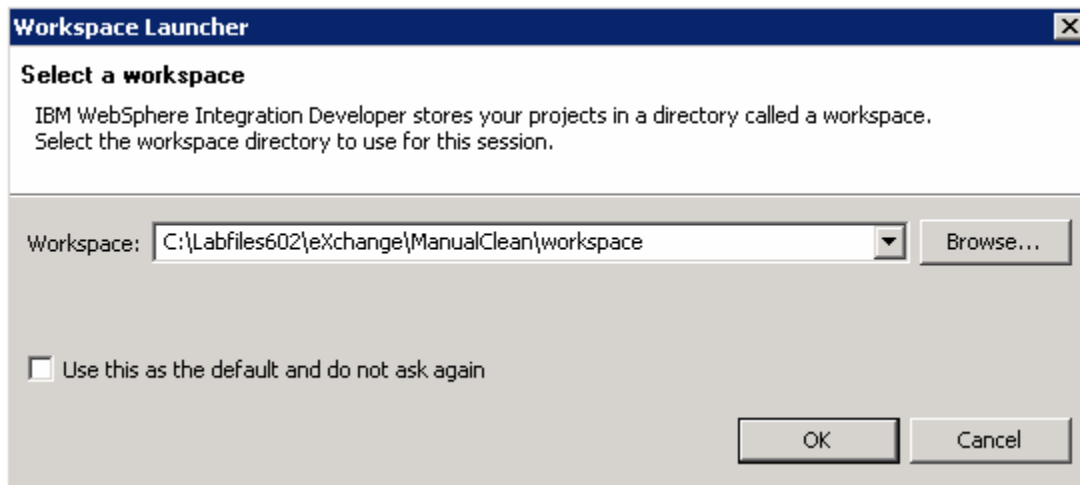
Reference Variable	Example: Remote Windows test server location	Example: Remote z/OS test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	cl1sr01	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\App Server	/etc/cl1cell/AppServerNode1	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<PORT>	9080	9080	
<BOOTSTRAP_PORT>	2809	2809	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	cl1admin	
<PASSWORD>	N/A	fr1day	

Instructions for using a remote testing environment, such as z/OS[®], AIX[®] or Solaris, can be found at the end of this document, in the section "[Task: Adding remote server to WebSphere Integration Developer test environment](#)".

Part 1: Initialize the workspace for this lab exercise and create a module

In this section of the lab, you will be importing a library project part of the CleansePublishLibrary_Pi.zip project interchange file into your workspace and eventually create a module named ManualClean.

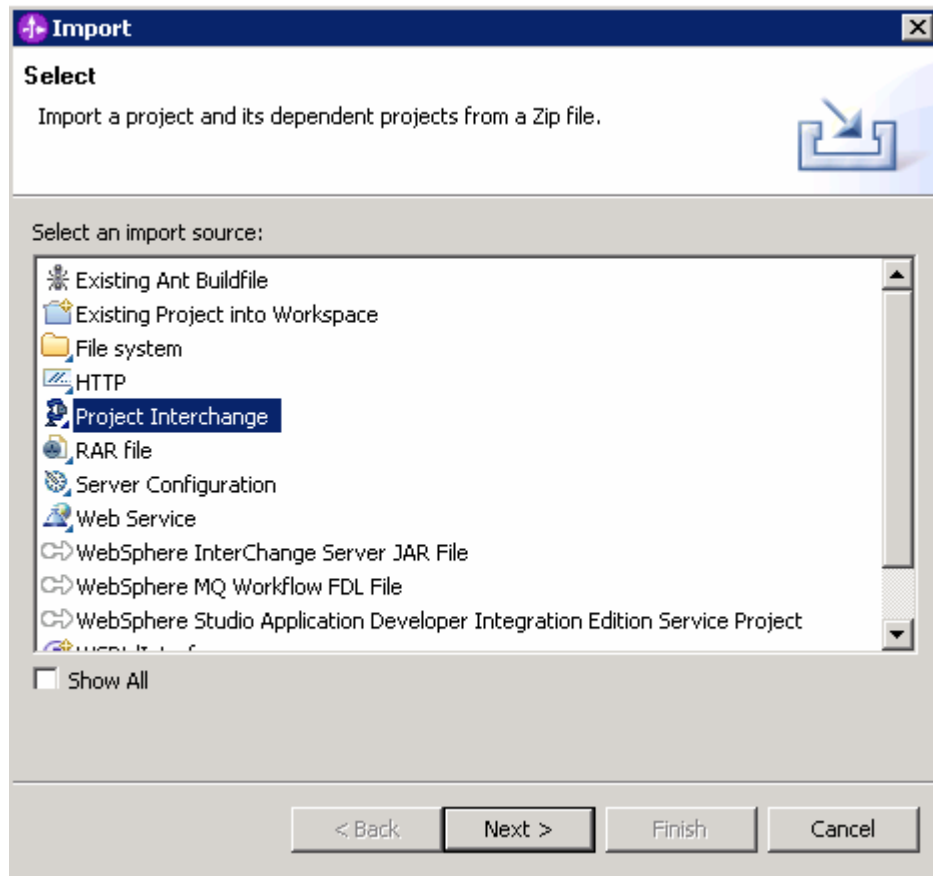
- ___ 1. Start WebSphere Integration Developer V6.0.2 with a workspace location of **<WORKSPACE>**, that is **<LAB_FILES>\eXchange\ManualClean\workspace**
 - ___ a. From Windows Explorer, navigate to the **<WID_HOME>** directory and double click on wid.exe.
 - ___ b. When prompted for workspace name, enter the value provided by the **<WORKSPACE>** variable for this lab and click **OK**.



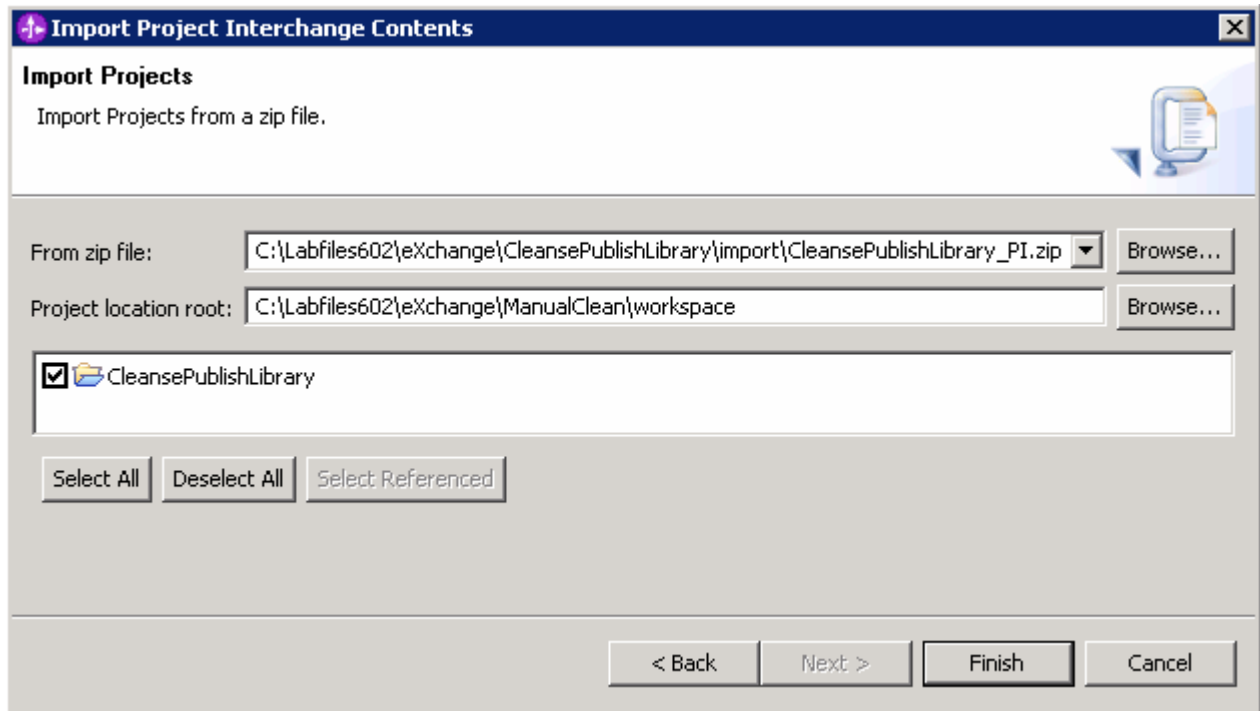
- ___ c. When WebSphere Integration Developer V6.0.2 opens, click the curved arrow at top right to **go to Business Integration perspective**.



- ___ d. Ensure you are in the **Business Integration** perspective.
- ___ 2. Import Project Interchange file, **CleansePublishLibrary_Pi.zip** located at **<LAB_FILES>\eXchange\CleansePublishLibrary\import**
 - ___ a. Right-click inside **Business Integration View** (top left view in the Business Integration Perspective) and select **Import** from the context menu
 - ___ b. Select **Project Interchange** listed in the import dialog

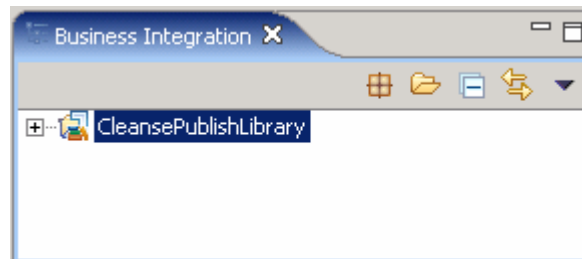


- __ c. Click **Next**
- __ d. Click the **Browse** button for “**From zip file**” and navigate to **<LAB_FILES>\eXchange\CleansePublishLibrary\import\CleansePublishLibrary_PI.zip** and hit **Open**

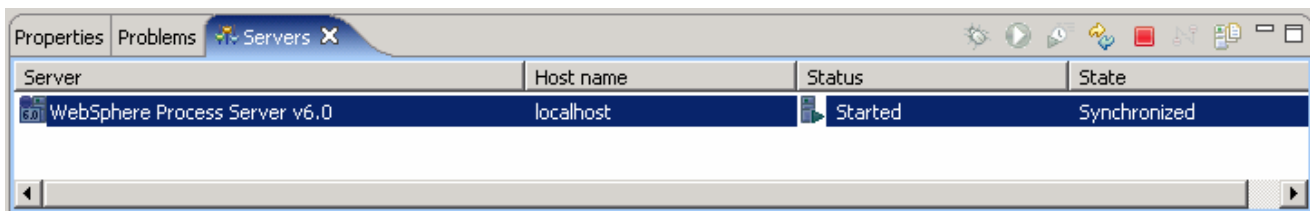


___ e. Select the check box next to **CleansePublishLibrary** and click **Finish**

___ f. Verify you have **CleansePublishLibrary** module listed in the Business Integration view



___ g. Verify you have WebSphere Process Server V6.0 listed in your Servers view

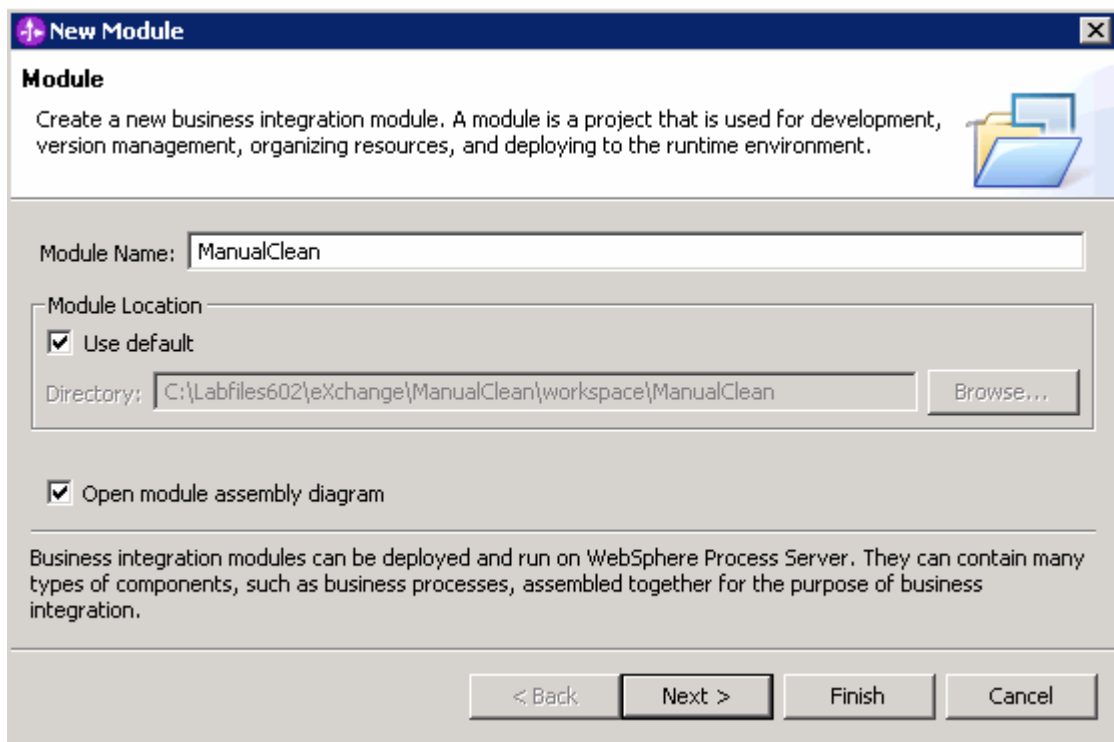


___ 3. Create a new module named, **ManualClean**

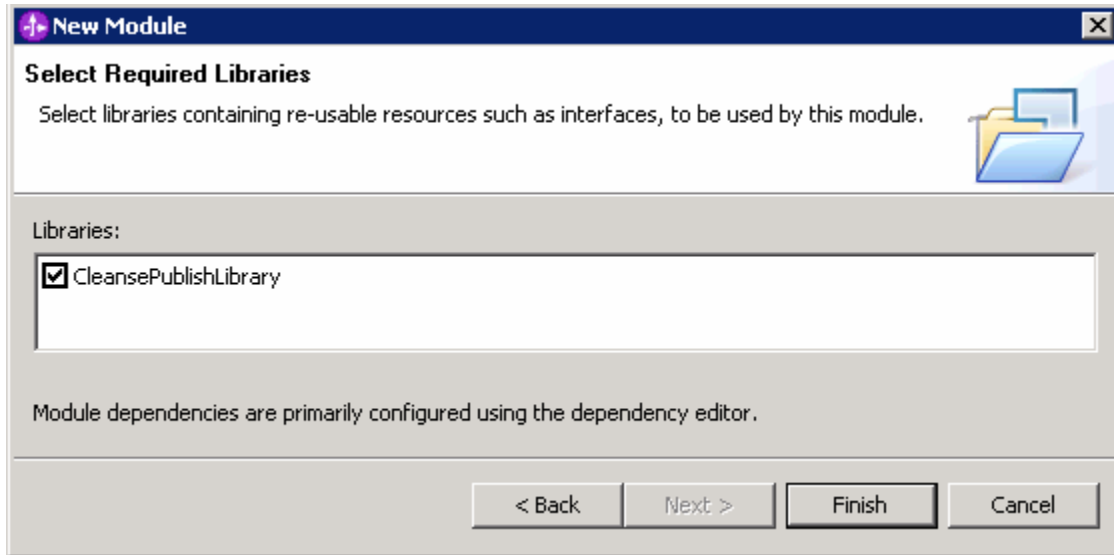
___ 4. Right-click in Business Integration view and select **New → Module**



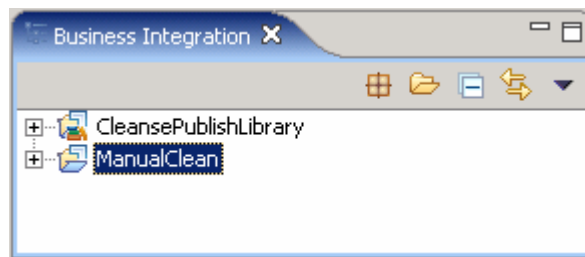
- 5. Enter the Module Name as **ManualClean**



- 6. Click **Next**
- 7. In the following panel, select the check box next to **CleansePublishLibrary**. This adds the CleansePublishLibrary as a dependant library for the ManualClean module



- ___ 8. Click **Finish**
- ___ 9. Verify you have **ManualClean** module listed in the Business Integration view



Part 2: Create manual cleanse

There are three different types of tasks you can create for integration applications.

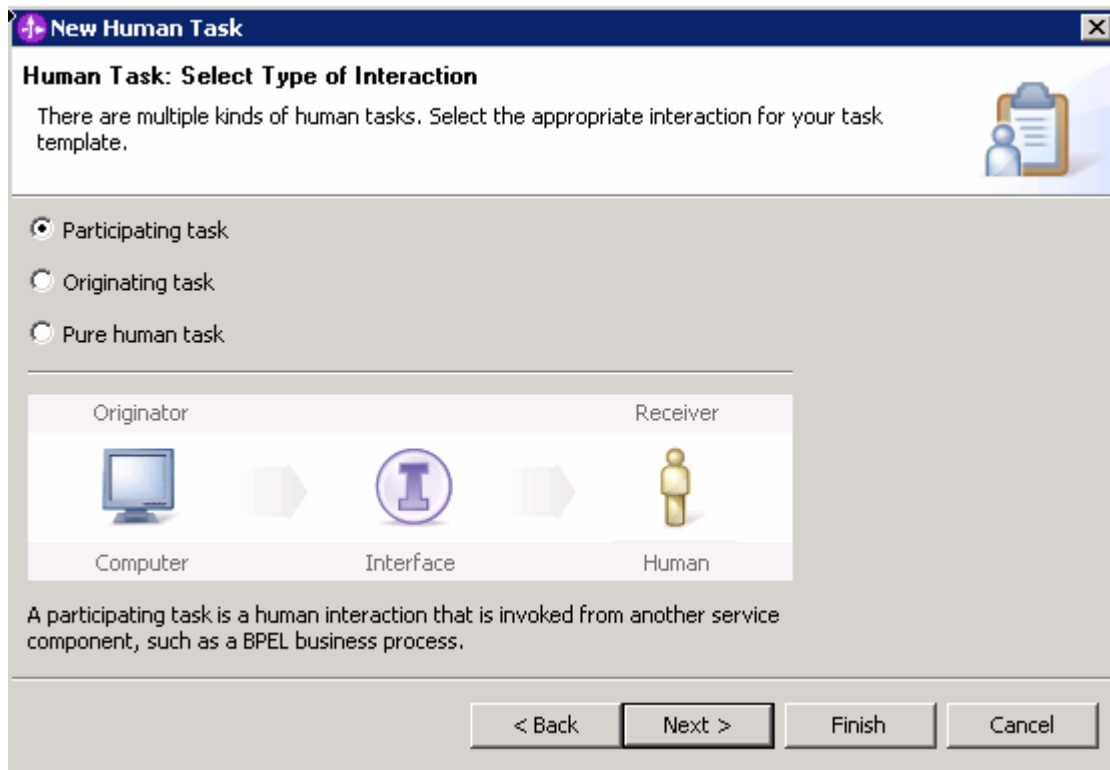
- **Participating tasks** provide a way for an integration application to create tasks for individuals.
- **Originating tasks** allow a person to invoke a service.
- **Pure human tasks** allow a person to create tasks for a specific person or a group of people.

Escalations can be set on all of these different tasks if the task is not claimed or completed in a specific time period. In this section you will create a participating task with different levels of permissions and escalation behavior.

Note: In order to correctly build and test, four user IDs (**user1**, **user2**, **user3**, and **adminuser**) are assumed to exist on the system and be part of the Administrators group. You can substitute different user IDs in the instructions if different user IDs exist on your system.

- ___ 1. Create a participating human task named ManualClean.
 - ___ a. In the Business Integration view, expand **ManualClean** → **Business Logic**. Now right click on **Human Tasks** and select **New** → **Human Task** from the context menu
 - ___ b. Enter **ManualClean** for the Name in the New Human Task panel

- ___ c. Click **Next**
- ___ d. In the following panel, ensure the radio button next to **Participating Task** is selected

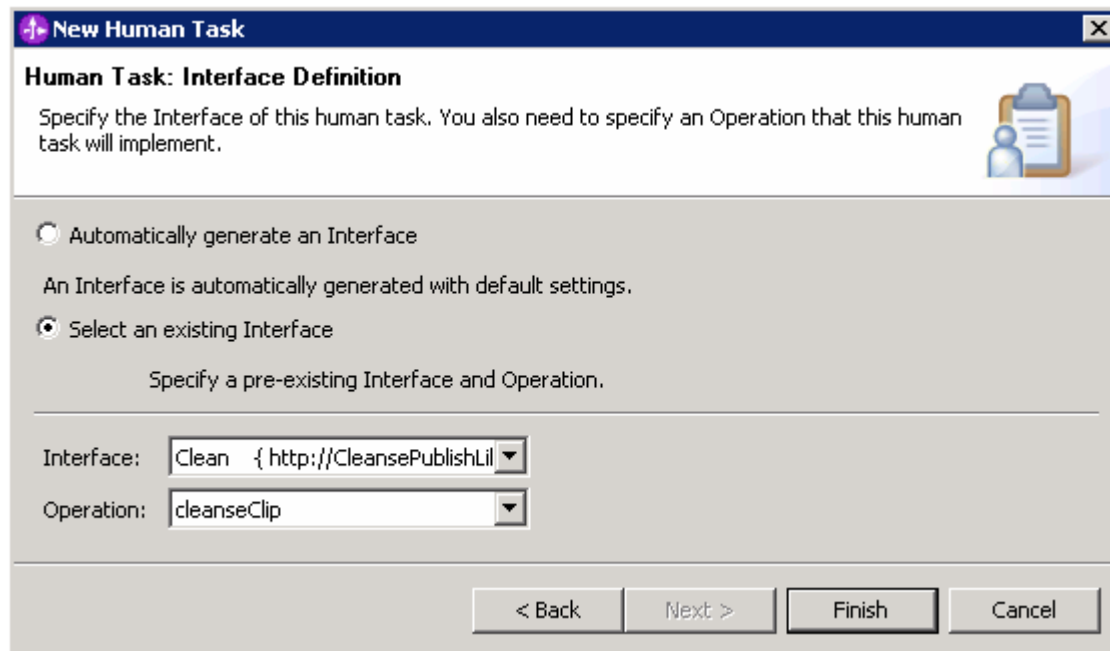


__ e. Click **Next**

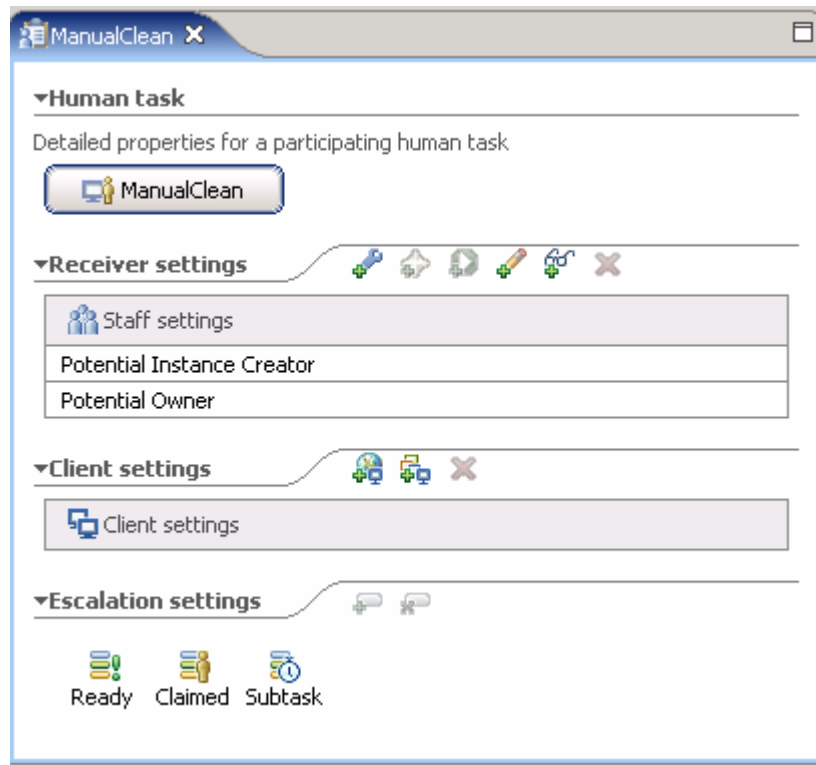
__ f. In the following panel, select the radio button next to **Select an existing interface** and specify an existing interface and an operation

1) Interface : **Clean { http://CleansePublishLibrary}**

2) Operation : **cleanseClip**



___ g. Click **Finish**. The human task editor will open



___ 2. Set the potential owner and administrator permissions for the task

There are **five** different permissions that can be set on a participating task. (Other tasks have similar permissions with slight differences).

Potential Instance Creator – The Potential Instance Creator can be set to limit who can create a participating task. With a participating task, the originator of a task is a system and this permission can require that the system is passing the correct token for authorization before the task is created.

Potential Owner - Potential Owners can claim Tasks, or make changes to tasks they claim. Potential Owners are also ultimately responsible for completing tasks which they have claimed or assigned to others as sub-tasks. Potential Owners can also transfer claimed tasks to other Potential Owners of the Staff activity.

Editor - Editors are allowed to view or update tasks that have been assigned to other owners. When a task is claimed, the owner can update the task information without completing it. That information can then be updated by an Editor. Editors cannot complete tasks. For example, at an insurance call center, an employee (the owner) has claimed a task for resolving an insurance claim. In the meantime, another employee (with editor access) finds additional information that is relevant to the claim and updates the task. The owner of the task can then view the additional information and complete the task.

Reader - Readers can view tasks that have been claimed. They can view any information that may have been set before the task is completed. Readers can not change or complete tasks.

Administrator – Administrators can perform all the same tasks as the Potential Owner. The administrator can also complete tasks claimed by another person and transfer a task to someone who is not a Potential Owner.

When setting the Potential Owner role to a user or group of users, you must specify a verb and parameters. The Verb is mapped to a specific search which is run against user registry by the Staff plug-in provider for that user registry. By default, there are a number of Verbs and mappings available. These Verbs are also available for the Editor and Reader roles. Some more information on a number of Verbs:

Users - Specify a single user to claim and then complete a task. For a Local OS user registry this can be the user ID. For LDAP user registry, the expected format for this value would be the distinguished name (cn=John Doe, ou=New York, o=Acme, c=US). Best practice is to use the Group Members verb instead of Users and specify a group which contains the user. This best practice allows for installed applications with tasks to remain unchanged when changes occur within the company hierarchy.

Users by user ID - Specify a user by their user ID. For a Local OS user registry this is the same as **Users**. For LDAP user registry, this value would be the **uid** attribute for an inetOrgPerson. Best practice with this Verb is to use the Group Members verb rather than a specific user.

Group Members - Specify a group of users who can claim and then complete a task.

Group Members without Named Users – Specify the group and a user who should be excluded from receiving a task. This verb, with the use of a process context value, can be used to implement a “4-eyes” policy where tasks must be approved by two different users from the same group.

Manager of Employee - Specify an employee by distinguished name (cn=Susan Long, ou=New York, o=Acme, c=US). The manager of the employee will then be able to claim and complete the task. Use of this verb requires a Manager-Employee relationship to exist with the user registry. For LDAP user registry, the manager attribute is used by default for resolving this query. Default support for this verb is for an LDAP user registry only.

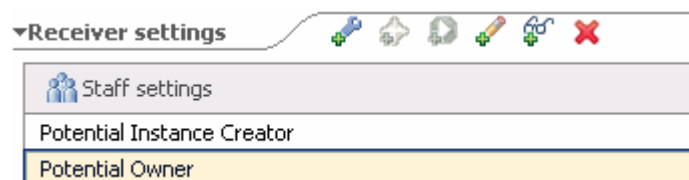
Manager of Employee by user ID - Very similar to **Manager of Employee** except the employee user ID is used instead of the full distinguished name. Support for this Verb is only available for LDAP user registries.

Native Query - Specify a query specific to the user registry. Before using Native Query, consider creating a new Verb and appropriate translation to the actual implementation. With a Native Query, changes made to the user registry have the potential to affect the business process and force updates to be made to the process. This forces the business process to be deployed again to the Business Process engine.

Everybody - Use this Verb instead of maintaining tasks for each and every different user

Nobody - Use this Verb to restrict access to a role. Nobody can only be used in Reader roles.

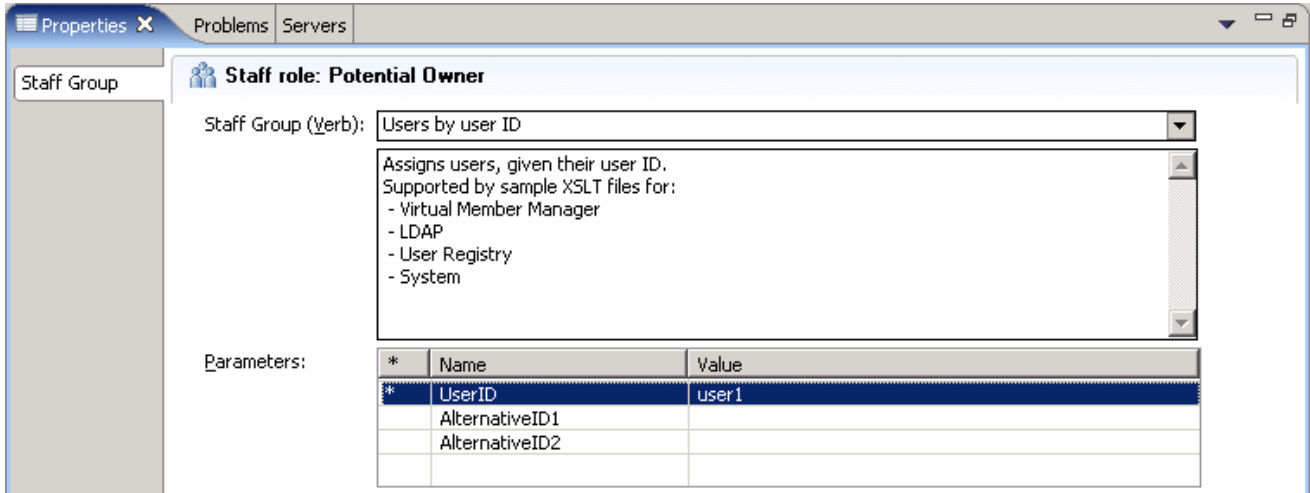
- ___ a. A Potential Owner is required for any participating task and is created by default under the Receiver Properties. Under the Receiver Settings section of the Human Task editor, select the **Potential Owner** entry and then the **Staff Group** tab under Properties view to set the Staff role for this user



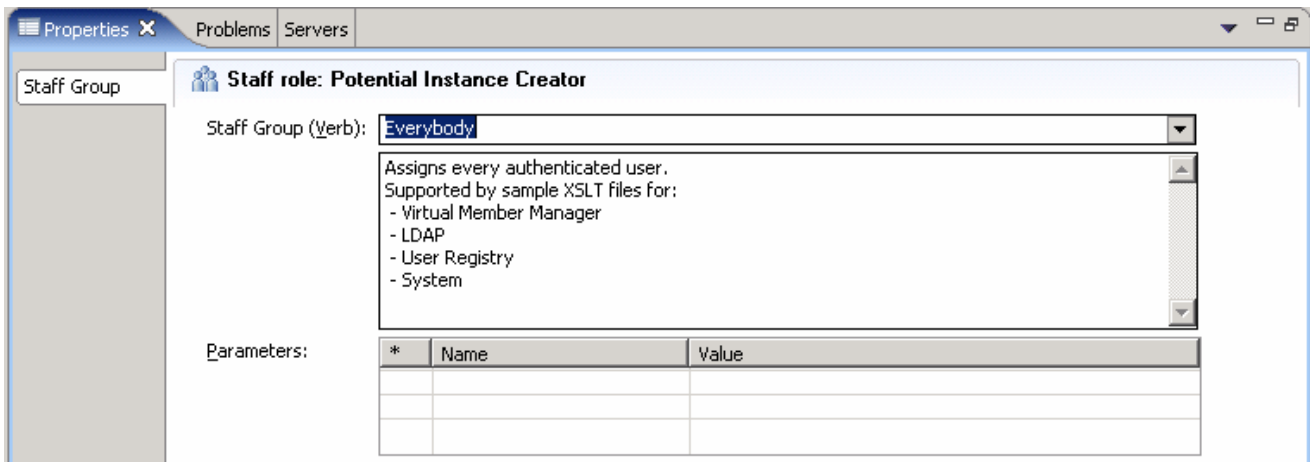
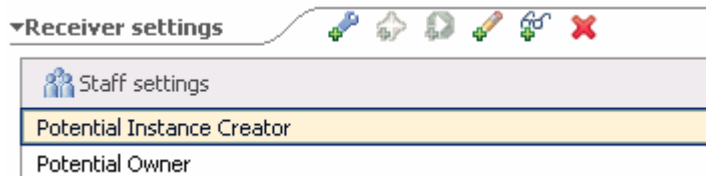
___ b. In the Staff Group tab under Properties view, provide the following information:

- 1) Staff Group (Verb) : **Users by user ID** (select from the drop down list)
- 2) User ID : **user1** (under Parameters section)

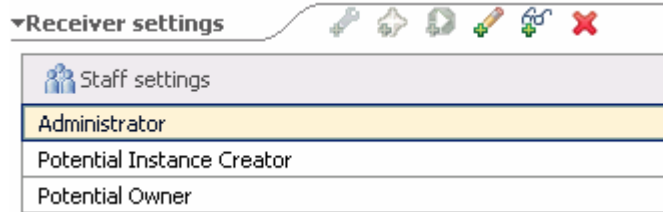
Note: For simplicity, a single user will be specified rather than a group even though using the Group Members verb is a best practice.



___ c. Now, under the Receiver Settings section of the Human Task editor, select the **Potential Instance Creator** and then the **Staff Group** tab under Properties view to review the Staff role for this user. The Default setting is **Everybody** for Staff Group (Verb) and any client will be able to create a **ManualClean** task



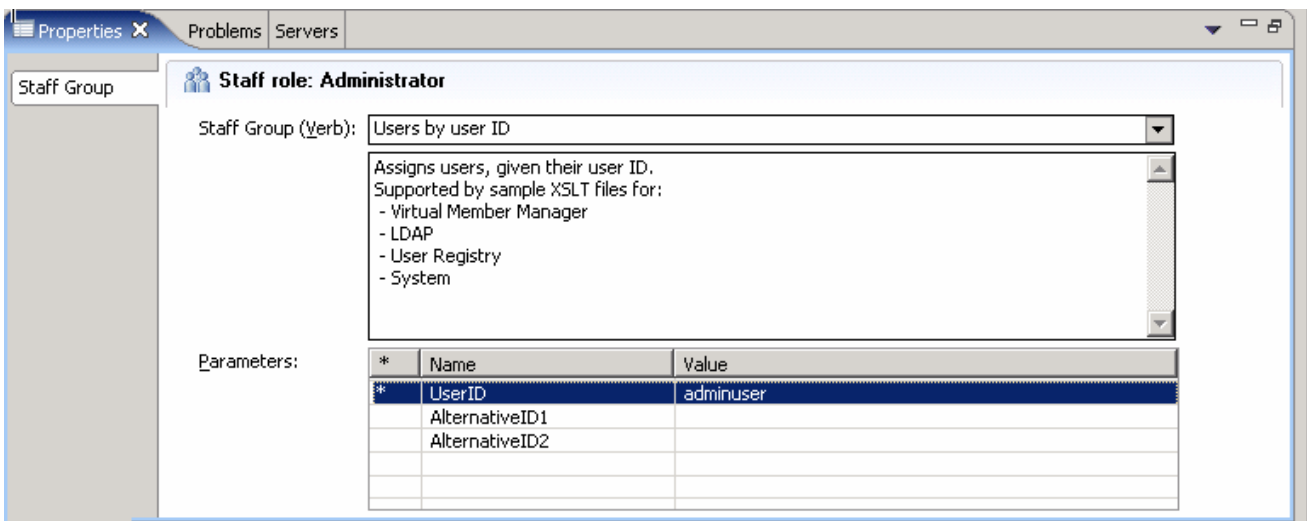
___ d. Under the Receiver Settings section, click the Administrator (🔑) icon to set an administrator for the task. A new entry named **Administrator** is created under the Receiver Settings section



___ e. Select the **Administrator** entry and then the **Staff Group** tab under Properties view to set the Staff role for this user.

___ f. In the Staff Group tab under Properties view, provide the following information:

- 1) Staff Group (Verb) : **Users by user ID** (select from the drop down list)
- 2) User ID : **adminuser** (under Parameters section)



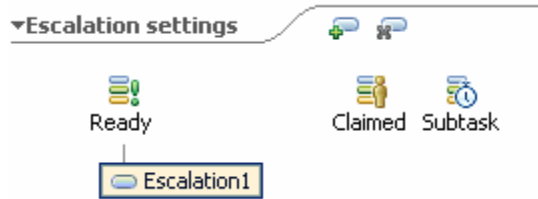
___ 3. Set an escalation on the task to assign a task if it has not been completed in a timely fashion.

A robust set of escalations may be established for a task. Escalations are run based on an exceeded duration and the current state of task. When a state is reached (Ready, Claimed, or Subtasks created), the escalation chain begins at that state. The first escalation in the chain will run after the duration has exceeded and will check to see if the required state for the escalation has been reached. If the state has been reached, the escalation will not fire and the escalation chain will continue. If there is another escalation in the chain, it will fire when its duration has been reached, checking to see if the required state has been reached for the task. Escalation times are relative to the preceding escalation durations. For example if escalation is 2 days, and the second escalation is 3 days. The first escalation would occur after 2 days and the second escalation would occur 3 days later or 5 days from the start of the escalation chain.

___ a. Under the **Escalation settings** section of the Human task editor, select the **Ready** icon and then click the **Escalation (+)** icon to add a new Escalation.

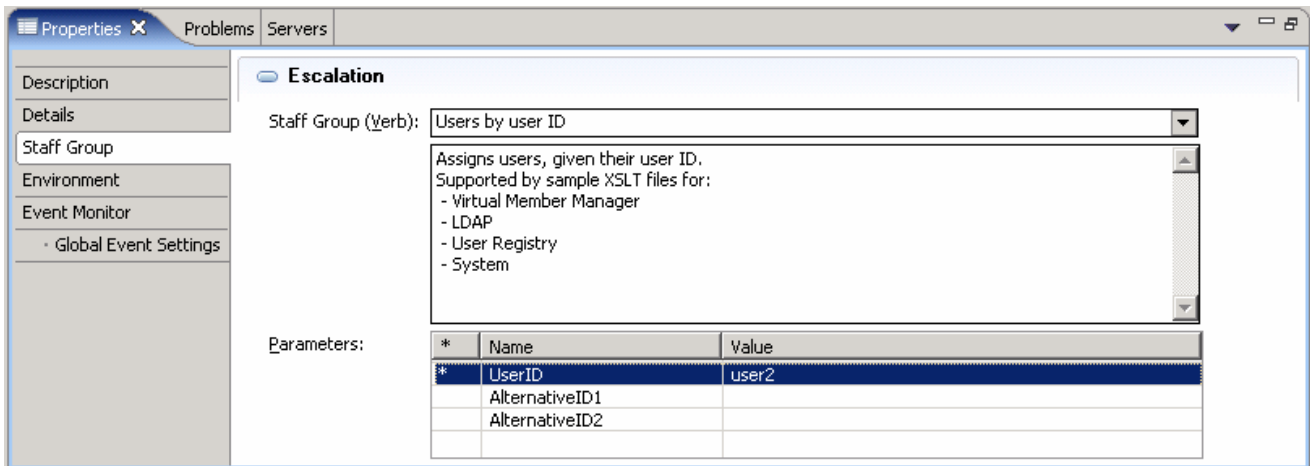


___ b. A new Escalation named **Escalation1** is added as shown below:



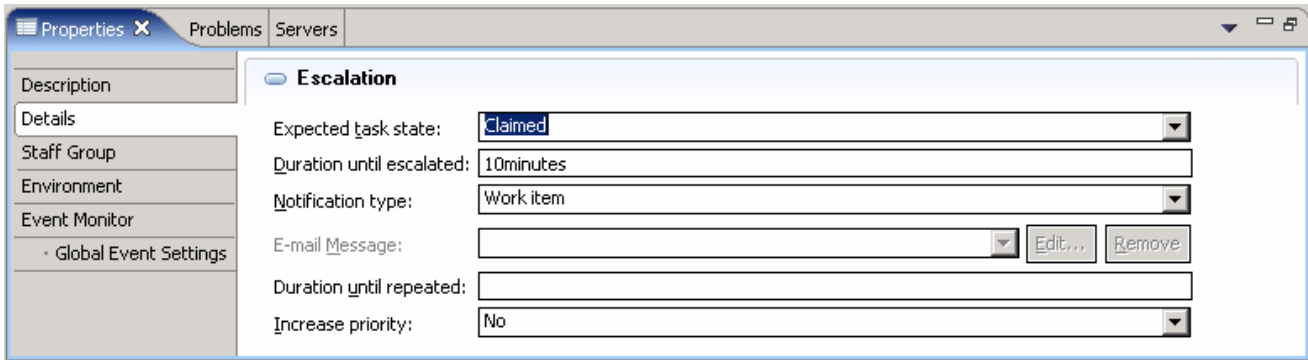
___ c. In the Properties view for **Escalation1**, there are a number of different settings. Escalation can result in an action being performed such as creation of a work item for a task, generation of an e-mail, or an event generation, which can be monitored at a central location. For assigning tasks, a Staff Group (Verb) and Parameter must be specified and will be used to search a user registry for the appropriate person to assign the task. While Escalation1 is selected, select the **Staff Group** tab under the Properties view and enter the following information:

- 1) Staff Group (Verb) : **Users by user ID** (select from the drop down list)
- 2) User ID : **user2** (under Parameters section)



___ d. While you are in the Properties view for **Escalation1**, select the **Details** tab. The combination of the state and the duration are checked at the time the escalation occurs and can be specified on this tab. Enter the following information:

- 1) Expected task state: **Claimed** (select from the drop down list)
- 2) Duration until escalated: **10minutes** (This escalation will occur after 10 minutes if the task is not claimed)
- 3) Notification type: **Work item**



Note: The format for the duration is based on the calendar, which is set on the Details tab for the task. The calendar supports different time formats, such as 10minutes, 5days, or 2months. Custom calendars may also be created to support additional formats.

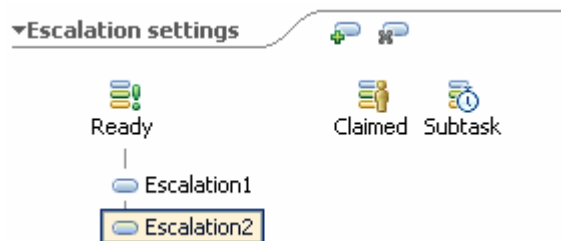
Note: Escalation occurs when the duration is exceeded. A built-in scheduler checks for escalation approximately every 10 minutes and is not meant for precision firing. Therefore, if escalation duration is reached 1 minute after the scheduler last ran, it could be over 9 minutes before the escalation occurs. If escalations must fire exactly after the duration has expired, you might want to consider a solution other than the built in escalation capabilities.

Note: For this example a small duration was used in order to easily simulate escalations. Typically, a larger value would be used.

In summary, for this escalation, after the task is created, if it is not claimed in 10 minutes, a task will be created for user2.

___ e. A second escalation can be created after the first escalation. Under the Escalation settings of the Human Task editor, select **Escalation1** that you had created before and click the **Add Escalation** (+) icon again.

___ f. A new Escalation named **Escalation2** is added as shown in the picture below:



___ g. For the second escalation, that is **Escalation2**, follow similar steps to set the properties as was done for the first escalation, **Escalation1** and set the correct values to create a task for **user3** after **10minutes** if the task is not **Claimed**. For the second escalation of 10 minutes, this will fire 10 minutes after the first escalation has fired or a total of 20 minutes after the task was created. You could also create an escalation chain which would start when the task was claimed. To create these escalations, follow the same steps as above.

___ h. Save all work by **File → Save All** or **Ctrl + Shift + S**

Part 3: Prepare for unit testing and assemble manual cleanse

In this part, you will prepare the task for testing by specifying the Staff plug-in provider and assembling it as a SCA component. In order for a participating task to be called by other SCA components, you must create an SCA binding.

1. Specify the Staff plug-in provider for the task.

In order for tasks to be assigned, a query must be run against a user registry to locate the correct user or users based on the Verb and Parameters specified. The query is run by a resource running in WebSphere Process Server known as a Staff plug-in provider. Since each user registry is potentially different in schema and connection type, there are different Staff plug-in providers for each user registry. WebSphere Process Server includes 3 Staff plug-in providers that can be used for locating users for task assignment:

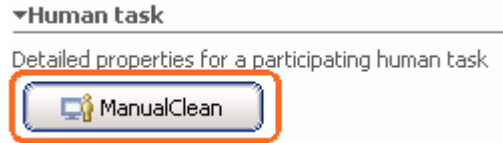
LDAP Staff plug-in provider – LDAP servers can be used for locating users for task assignment with this Staff plug-in provider. This provider supports the largest number of verbs and provides the highest level of customization because many attributes can be defined in the LDAP server and these can be used in searches from the LDAP Staff plug-in provider.

User Registry Staff plug-in provider – The User Registry Staff plug-in provider directs all searches for users and groups to the security manager running in WebSphere Process Server. When Global Security is enabled, the security manager communicates with a particular user registry (local OS, LDAP, or custom registry) to perform authorization of requests coming in to the server. The User Registry Staff plug-in provider can pass searches to the security manager and utilize any of the different registries. Since the security manager only needs to search for users and groups, the number of Verbs available to the User Registry Staff plug-in provide is limited to searches for users and groups too. Verbs such as Manager of Employee, Department Members, and Role members are not available for this Staff plug-in provider.

System Staff plug-in provider – For test environments, the System Staff plug-in provider supports a small number of verbs and is ideal for verifying the logic of task assignment without dealing with the complexities of actual user registries. With this Staff plug-in provider, no queries are run against a user registry to locate users or groups. Instead, hard-coded values, specified as Parameters are used. The System Staff plug-in provider also contains a special configuration that can assign all tasks to anyone (everybody). With this setting, everyone will be able to view, claim, or complete tasks without configuring security on the server. Because of the lack of capabilities and potential security risks, the System Staff plug-in provider is only supported for test environments and is not intended for production servers.

You can prepare the ManualClean task for easy testing by setting it to use the Everybody capability on the System Staff plug-in provider.

___ a. Select the **ManualClean** button under the Human Task section of the Human task editor

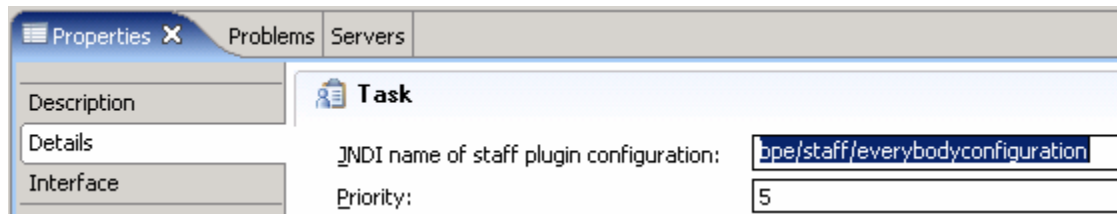


___ b. Now select the **Details** tab under the Properties view and enter the following information:

- 1) JNDI name of staff plug-in configuration : **bpe/staff/everybodyconfiguration** (select from the dropdown list)
- 2) Accept the defaults for the remaining properties

Note: This is the configuration which provides the ability to make tasks available to everyone and is part of the System Staff plug-in provider which is preconfigured in WebSphere Process Server.

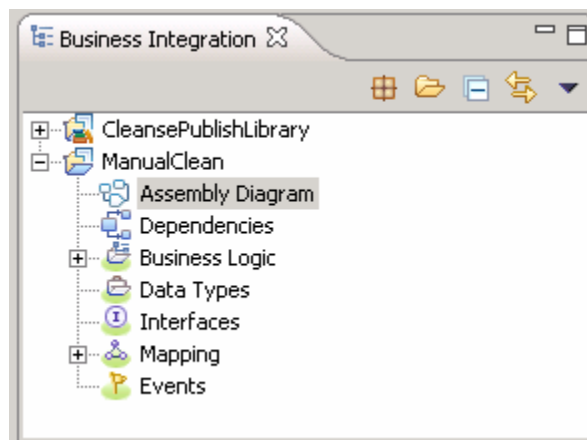
Note: The Staff plug-in providers can be viewed in the WebSphere Process Server Administrative console under Resources → Staff plug-in provider.



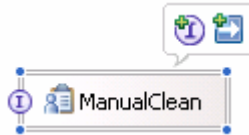
___ c. Save all work by **File → Save All** or **Ctrl + Shift + S**

___ 2. Assemble the human task as a SCA component

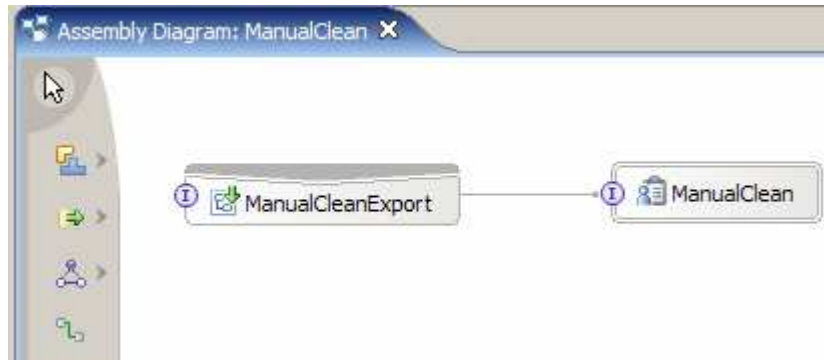
___ a. Expand **ManualClean** in the Business Integration view and double click on **Assembly Diagram** (Assembly Diagram) to open it in an Assembly Diagram editor



___ b. In the Business Integration view, expand **ManualClean → Business Logic → Human Tasks** and drag the **ManualClean** task on to the Assembly Diagram.



- ___ c. In the Assembly editor, right-click on **ManualClean** and select **Generate Export →SCA Binding** from the context menu. The resultant Assembly diagram is as shown below:



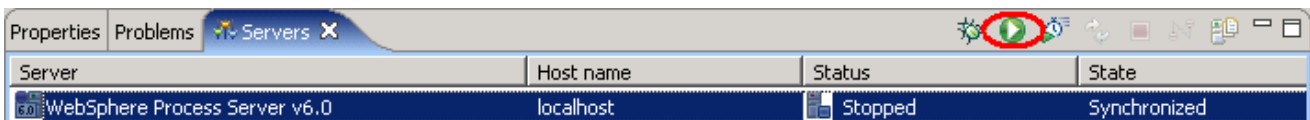
- ___ 3. Save the Assembly editor (**File → Save All** or **Ctrl + Shift + S**). The task is now a SCA component and the appropriate code will be generated to expose the ManualClean human task as a SCA component

Part 4: Test manual cleanse

In this part, you will use the End-to-End Test Framework in order to test the ManualClean human task.

- ___ 4. Start the WebSphere Process Server, that the test environment server.
- ___ a. If using a remote testing environment, follow the directions provided in [Task: Adding remote server to WebSphere Integration Developer test environment](#) at the end of this document to add a server to the WebSphere Integration Developer test environment and start it. This is especially true for z/OS, AIX, Solaris remote test environment, where the WebSphere Integration Developer will be remote to the test environment.

If using a local testing environment, right click on WebSphere Process Server V6.0 from the Server view and select **Start the server** button on the right hand side:



- ___ 5. Add the ManualClean application to the server
- ___ a. Right-click on the server in the Servers view and select **Add and remove projects** from the context menu
- ___ b. Select **ManualCleanApp** and then click **Add** to Move the ManualCleanApp to the **Configured projects** list
- ___ c. Click **Finish**
- ___ 6. Prepare for the test
- ___ a. In the Assembly diagram, right-click on **ManualClean** module after the server has started and select **Test Component** from the context menu
- ___ b. A test instance will open with a table displaying the initial request parameter. Enter the following values or of your choice under **Initial request parameters** section:
- verb : **Create** (from drop down list)
 - clipID : **1000**
 - GLN : **1000**
 - clip : **testClip**
 - size : **500**
 - color : **red**
 - brand : **dull**

▼ Detailed Properties

Configuration: Default Module Test

Module: ManualClean

Component: ManualClean

Interface: Clean

Operation: cleanseClip

Initial request parameters

Name	Type	Value
<input type="checkbox"/> inClipBG	ClipBG	
verb	string	Create
<input type="checkbox"/> Clip	Clip	
clipID	string	1000
GLN	string	1000
clip	string	testClip
size	string	500
color	string	red
brand	string	dull
retailItems	ClipItem []	<null>

Data Pool Continue

___ c. Right-click on the value for **retailItems** entry and select **Add Element**

Initial request parameters

Name	Type
<input type="checkbox"/> inClipBG	ClipBG
verb	String
<input type="checkbox"/> Clip	Clip
clipID	string
GLN	string
clip	string
size	string
color	string
brand	string
retailItems	ClipItem []

Data Pool

Set Value...
Add Element
 Remove Element

___ d. Enter **1** in the Add Element Dialog and click **OK**

___ e. Now expand **retailItems** and enter the following values or of your choice:

- itemID : **123**
- GTIN : **456**
- package : **small**

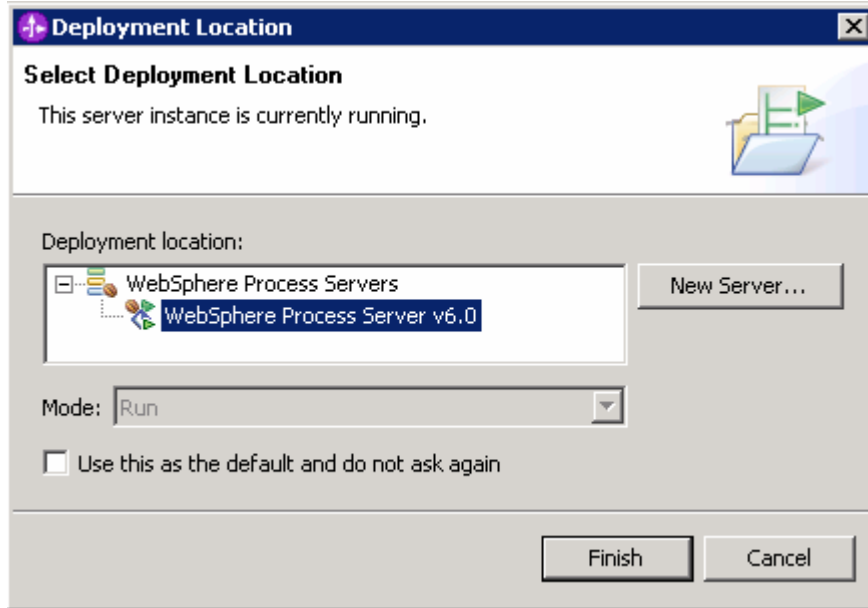
- quantity : 1
- fullDescription : **dull clips**
- price : **10.0**
- startDate : **2007-03-21**
- endDate : **2007-04-09**
- contactFirstName : **John**
- contactLastName : **Doe**

Initial request parameters

Name	Type	Value
clip	string	testClip
size	string	500
color	string	red
brand	string	dull
<input type="checkbox"/> retailItems	ClipItem []	
<input type="checkbox"/> retailItems[0]	ClipItem	
itemID	string	123
GTIN	string	456
package	string	small
quantity	string	1
fullDescription	string	dull clips
price	string	10.0
startDate	string	2007-03-21
endDate	string	2007-04-09
contactFirstN...	string	John
contactLastN...	string	Doe

Data Pool Continue

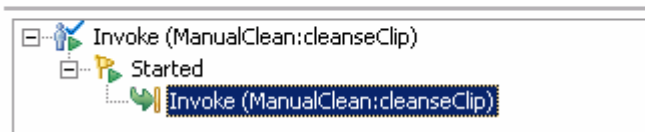
- ___ 7. Run the test
- ___ a. Click the **Continue** button under the list of initial request parameters section
 - ___ b. Select WebSphere Process Server 6.0 as the Deployment Location



- ___ c. Click **Finish**. The ManualCleanApp application and the Test Connector application which drives the testing on the server will be published to the server. The SCA component will also be invoked

Events

Events



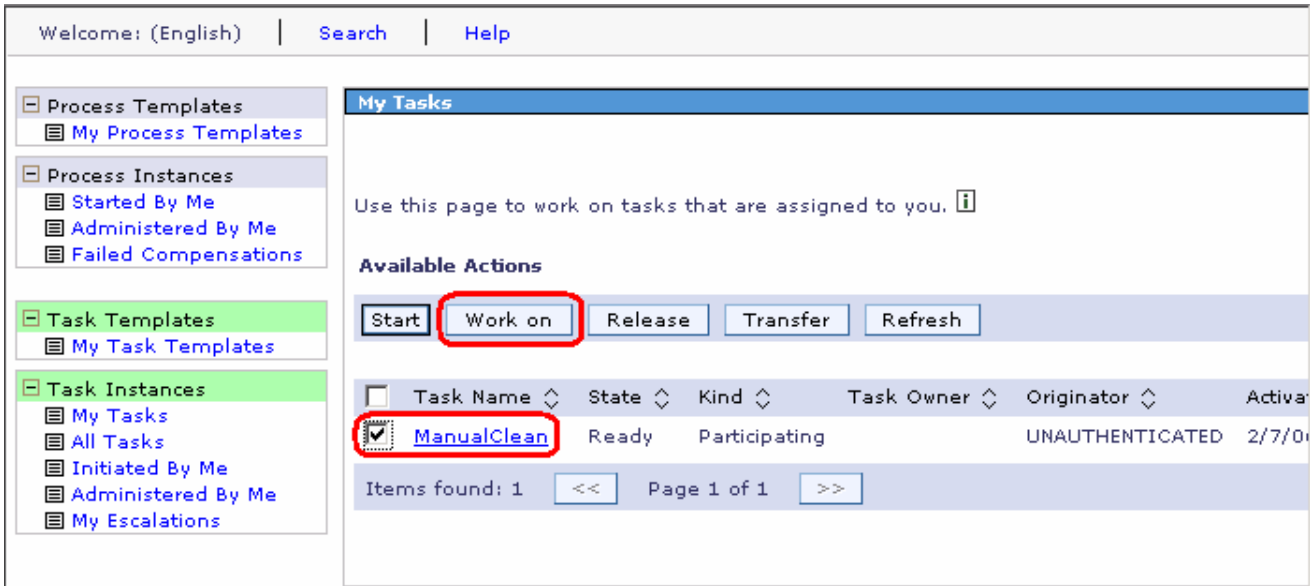
- ___ 8. Complete the Task implementation.

The SCA interface on the task is called and a task will be created and available to anyone as the Everybody configuration of the System Staff plug-in provider was used. WebSphere Process Server includes the BPC (Business Process Choreography) Explorer, which provides a Web interface for interacting with business process instances and tasks generated from human tasks. Users can view, claim, or complete tasks, depending on their permission levels for the tasks.

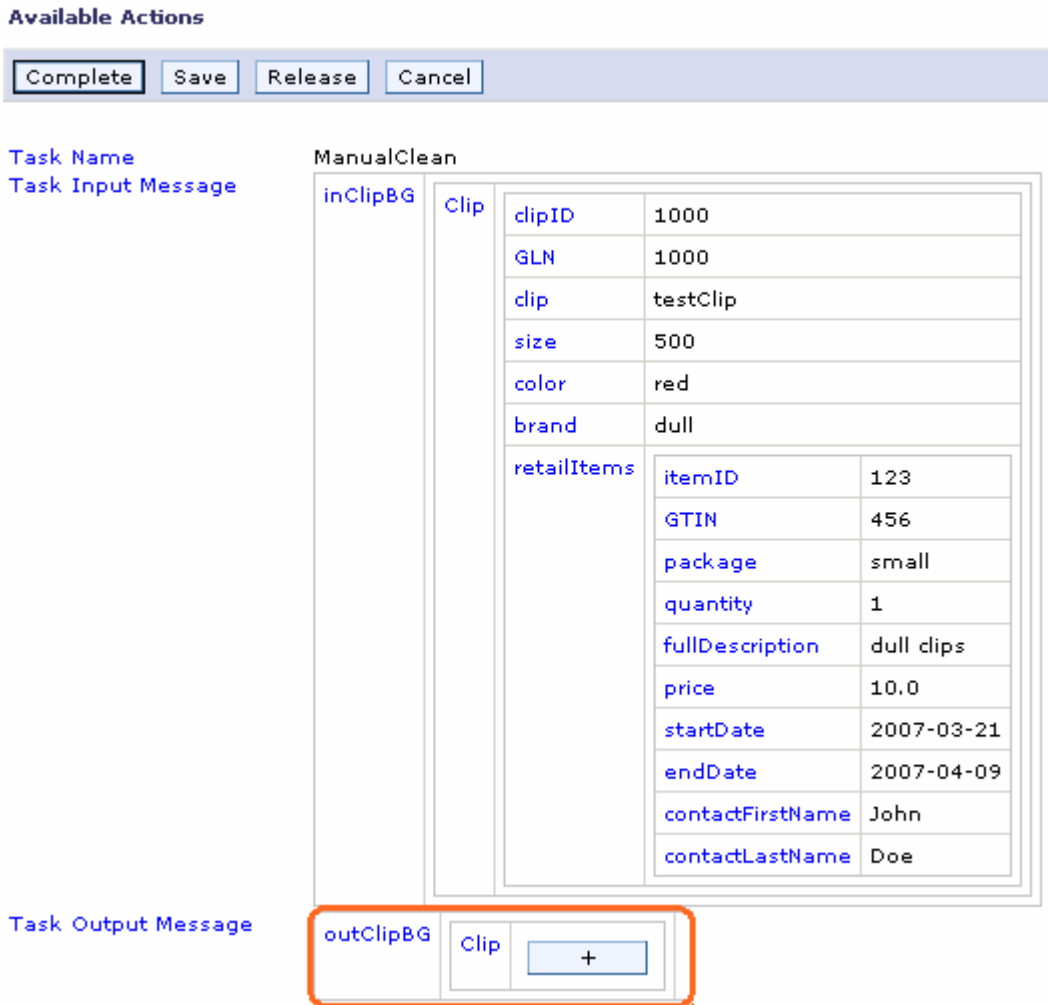
- ___ a. Right-click on WebSphere Process Server V6.0 in the Servers view and select **Launch → Business Process Choreographer Explorer** from the context menu. You can also open a Web browser and navigate to **http://<HOSTNAME>:9080/bpc**. The My Tasks view will be shown with the task ready to be completed.

Note: The task for ManualClean will be listed in the view. As security is not enabled and the System Staff plug-in provider with the Everybody configuration is being used, the tasks are listed for all users. If security was enabled and the Everybody configuration was not used, only those tasks which you have permission to work with will be listed.

- ___ b. Select the check box next to **ManualClean** and click the **Work on** button. This will claim the task on behalf of the current user. Since security is not enabled and you are unauthenticated, the task is claimed, but not for a specific person.



___ c. The ManualClean task with the inclipBG is displayed as shown below:



___ d. Now click the “+” button to expand and enter the following Task Output Message information:

- clipID : **1001**
- GLN : **1001**
- clip : **testClip2**
- size : **500**
- color : **very red**
- brand : **very dull**

Task Output Message

outClipBG	Clip	clipID	<input type="text" value="1001"/>
		GLN	<input type="text" value="1001"/>
		clip	<input type="text" value="testClip2"/>
		size	<input type="text" value="500"/>
		color	<input type="text" value="very red"/>
		brand	<input type="text" value="very dull"/>
		retailItems	<input type="button" value="+"/>
			<input type="button" value="-"/>

___ e. Now click the “+” button to expand and enter the following **retailItems** information or of your choice:

- itemID : **456**
- GTIN : **789**
- package : **medium**
- quantity : **2**
- fullDescription : **dull clips**
- price : **20.0**
- startDate : **2007-04-20**
- endDate : **2007-12-31**
- contactFirstName : **Jane**
- contactLastName : **Doe**

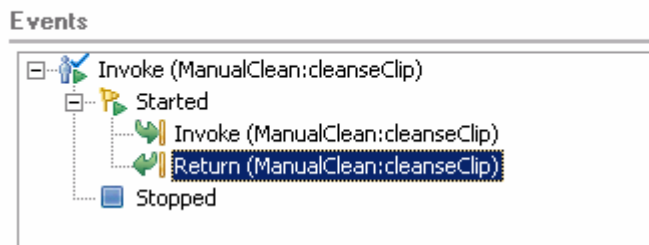
retailItems	itemID	456	
	GTIN	789	
	package	medium	
	quantity	2	
	fullDescription	dull clips	
	price	20.0	
	startDate	2007-04-20	
	endDate	2007-12-31	
	contactFirstName	Jane	
	contactLastName	Doe	
			-

___ f. Scroll to the top and click the **Complete** button. The list of tasks will be displayed again and will show the task as finished

Available Actions

___ g. Return to the component test window and view the return response captured as shown below:

Events



___ h. Notice in the Detailed Properties section, the response, that is the Output task Message values that you provided in when completing the task are reflected. This indicates the test was successful

► **General Properties**

▼ **Detailed Properties**

Module: [ManualClean](#)

Component: [ManualClean](#)

Interface: [Clean](#)

Operation: [cleanseClip](#)

Return parameters:

Name	Type	Value
[-] outClipBG	ClipBG	
verb	string	<unset>
[-] Clip	Clip	
clipID	string	1001
GLN	string	1001
clip	string	testClip2
size	string	500
color	string	very red
brand	string	very dull
[-] retailItems	ClipItem	
itemID	string	456
GTIN	string	789
package	string	medium
quantity	string	2
fullDescription	string	dull clips
price	string	20.0
startDate	string	2007-04-20
endDate	string	2007-12-31
contactFirstName	string	Jane
contactLastName	string	Doe

Note: If the task is not claimed within 10 minutes, the first escalation will be run. The 'Everybody' configuration is used with the Staff plug-in provider and when the escalation occurs, a task will be assigned to everybody. Since the task was originally assigned to everybody, no additional work items are created for the task. For this reason, the actual escalation behavior is not exhibited in this exercise. If a different configuration was used with the Staff plug-in provider and global security was enabled, a new and different task would be created for the escalation receiver.

___ i. The test is complete. Close the **Business Process Choreographer Explorer** and the Test Component editor. Click **No** when prompted for Save Changes to the Test Component Editor

___ 9. Close the Test Component

___ 10. Clean the test server environment

___ a. In the Servers view, right-click on WebSphere Process Server V6.0 and select **Add and remove projects**

___ b. Select AutoCleanApp and click **< Remove**. Click **Finish**

___ c. Right-click on **WebSphere Process Server V6.0** and select **Stop** from the context menu

What you did in this exercise

In this exercise, you built a human task with a potential owner and an administrator. You added escalations to create tasks after a particular duration is exceeded if the task is not claimed. With the End-to-End Test Framework, you tested the task as a SCA component, verifying task creation and completion.

Solution instructions

- ____ 1. Follow the directions in the task [Initialize the workspace for a lab exercise](#), using the following values:

<WORKSPACE>

C:\Labfiles602\eXchange\ManualClean\workspace

<PROJECT_INTERCHANGE>

C:\Labfiles602\eXchange\ManualClean\Solution\ManuelClean_Pi.zip

<MODULE>

n/a

<DEPENDENT_LIBRARIES>

n/a

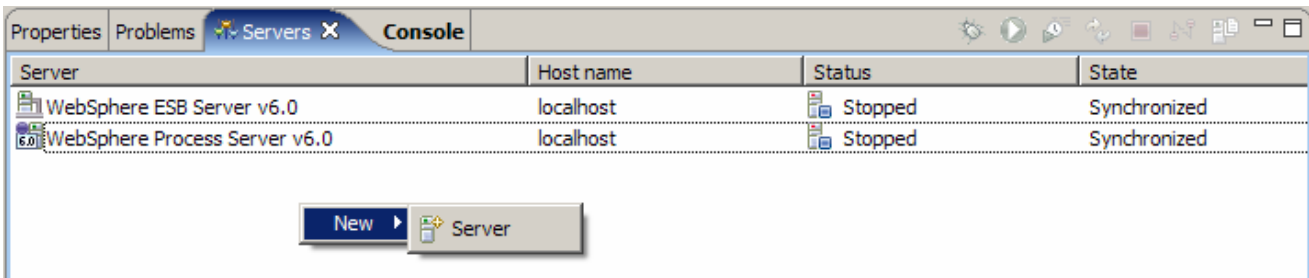
- ____ 2. Continue from **Part 4: Test Manual Cleanse**

Task: Adding remote server to WebSphere Integration Developer test environment

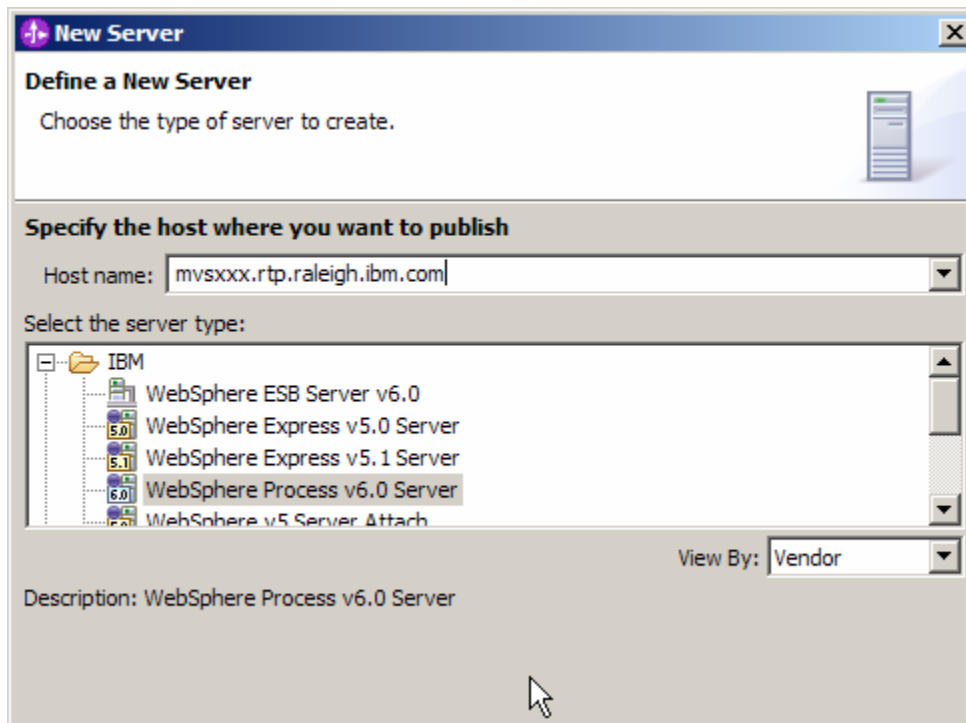
This task describes how to add a remote server to the WebSphere Integration Developer Test environment. The sample will use a z/OS machine.

Create a new remote server.

- ___ 1. Right click on the background of the Servers view to access the pop-up menu
- ___ 2. Select **New → Server**



- ___ 3. Specify host name to the remote server, **<HOSTNAME>**
- ___ 4. Ensure that 'WebSphere Process V6.0 Server' is highlighted in the server type list.



- ___ 5. Click **Next**.

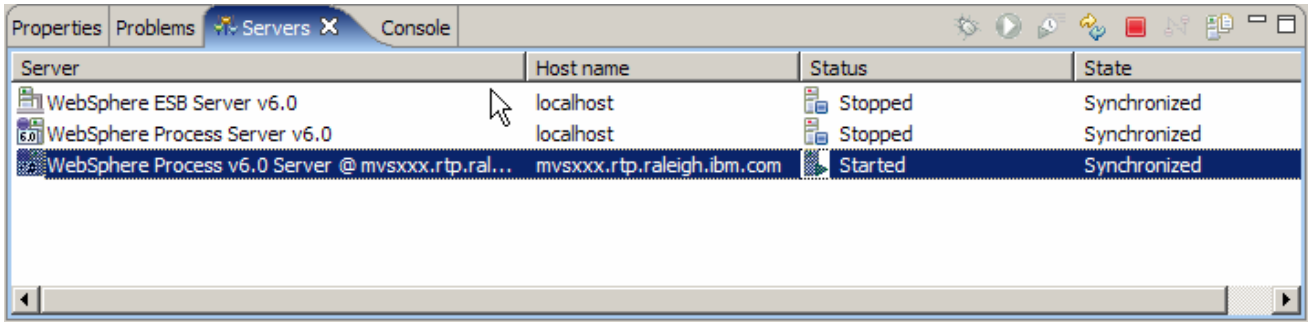
- ___ 6. On the WebSphere Server Settings page, select the radio button for **RMI** and change the ORB bootstrap port to the correct setting (<BOOTSTRAP_PORT>).

The screenshot shows the 'New Server' dialog box with the following settings:

- WebSphere profile name: [Empty]
- Server connection type and admin port:
 - RMI (Better performance)
 - ORB bootstrap port: 9131
 - SOAP (More firewall compatible)
 - SOAP connector port: 8880
- Run server with resources within the workspace
- Security is enabled on this server
- Current active authentication settings:
 - User ID: [Empty]
 - Password: [Empty]
- Server name: server1
- Server type:
 - BASE, Express or unmanaged Network Deployment server
 - Network Deployment server
 - Network Deployment server name: [Empty]
 - The server name is in the form of: <cell name>/<node name>/<server name>
 - For example, localhost/localhost/server1.
 - Click this button to detect the server type.

Buttons at the bottom: < Back, Next >, Finish, Cancel

- ___ 7. Click **Finish**.
- ___ 8. The new server should be seen in the Server view.



- ___ 9. Start the remote server if it is not already started. WebSphere Integration Developer does not support starting remote servers from the Server View.
- ___ 10. From a command prompt, telnet to the remote system if needed:
- ```
'telnet <HOSTNAME> <TELNET_PORT>'
 userid : <USERID>
 pw : <PASSWORD>
```
- \_\_\_ 11. Navigate to the bin directory for the profile being used:
- ```
cd <WAS_HOME>/profiles/<PROFILE_NAME>/bin
```
- ___ 12. Run the command file to start the server: `./startServer.sh <SERVER_NAME>`
- ___ 13. Wait for status message indicating server has started:

```
ADMU3200I: Server launched. Waiting for initialization status
ADMU3000I: Server c11sr01 open for e-business; process id is 0000012000000002
```