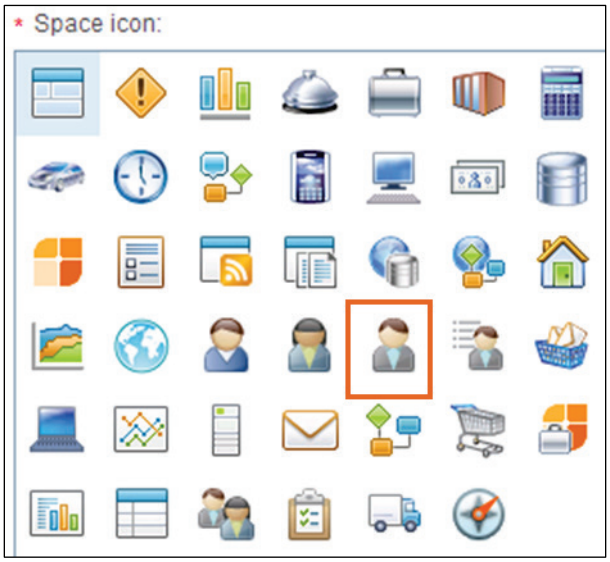
8. In the **Space style** window, select **Human** as the **Space style**.

   A style determines the color and appearance of a business space. You can use a particular style, for example, to indicate the purpose of the space. Make sure that you are using Firefox as the browser or else the Space styles are not listed.

9. You can also choose an icon to represent the business space. In the **Space icon** window, select an icon of your choice.

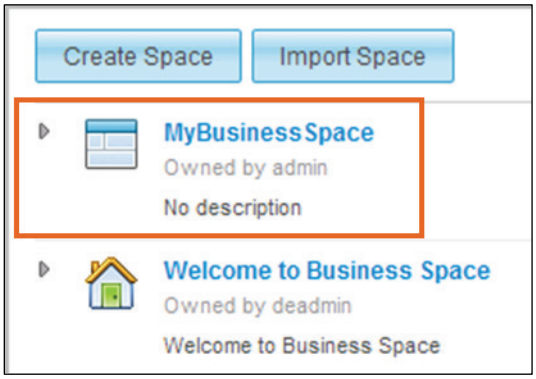


10. Click **Save** at the bottom of the window.

    The `MyBusinessSpace` workspace is added to the list of spaces.

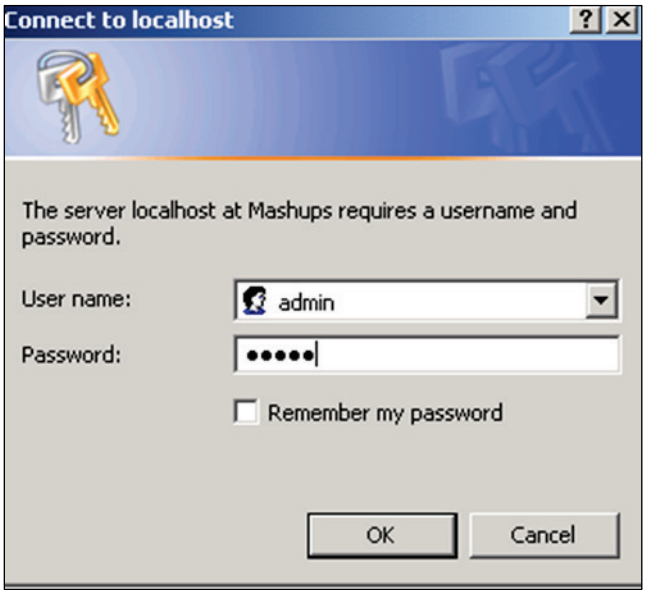
    You will explore the newly created `MyBusinessSpace` space.

11. Click **MyBusinessSpace**.



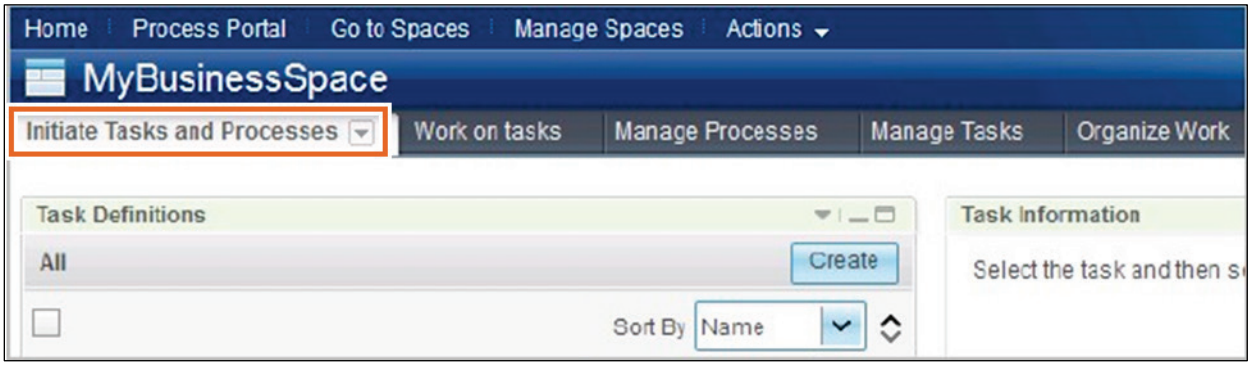
12. Verify that `MyBusinessSpace` opens in the Business Space. Five tabs or pages are created for you by default. These tabs are created because you selected the **Advanced task and Process Management** template during the space creation. This template creates a space with the following pages:
    - Initiate Tasks and Processes
    - Work on Tasks
    - Manage Processes
    - Manage Tasks
    - Organize Work

    If you receive a login window, then log back in the Business Space by entering `admin` in the User name field and `web1sphere` in the Password field and clicking OK.



13. On the **Initiate Tasks and Processes** page, you can initiate tasks, processes, and services.

    Additionally, you can also store incomplete forms as drafts, and check the status of initiated tasks and processes.

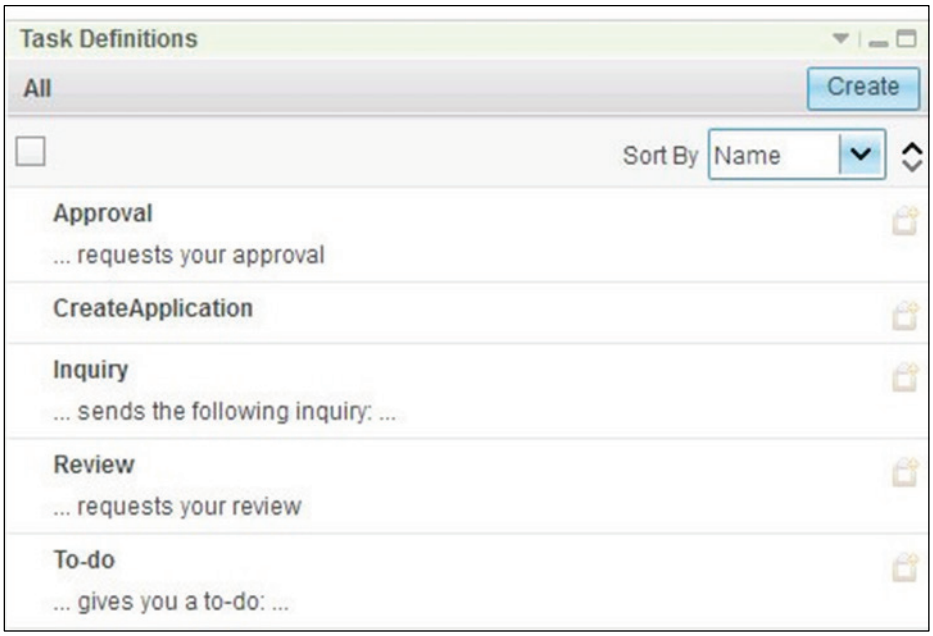


14. Note the different widgets that are already available on the page. The four widgets that are displayed are:

    - **Task Definitions**: Displays a list of task definitions that you use to create a task or to initiate services and processes.
    - **Task Information**: Displays information about a task that you selected in the Status widget or the Task Definitions widget. You can check and change the priority and due date of a task.
    - **Status**: Displays the tasks for the definitions that you selected in the Task Definitions widget. You can check and change the priority and due date of a task, or postpone work on a task and then resume it later.
    - **Process Information**: Displays information that is associated with the task or process.

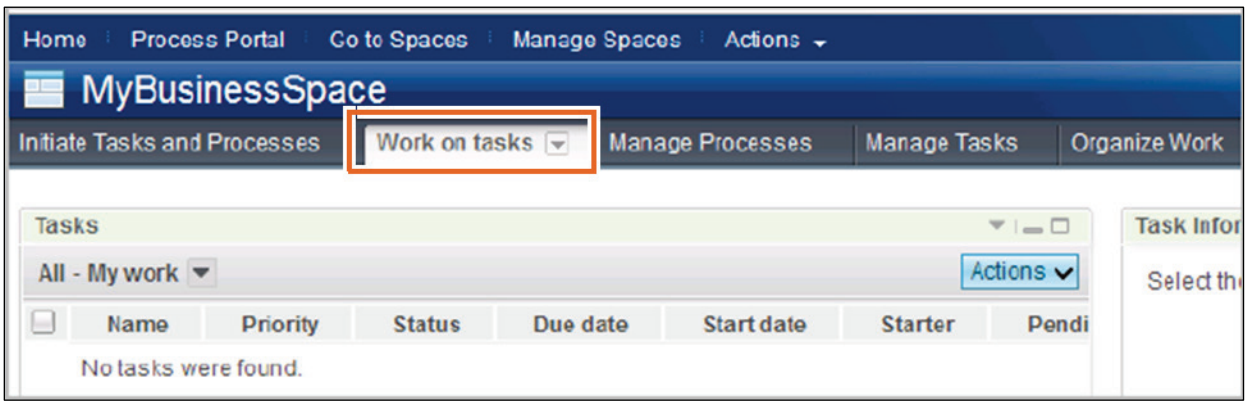
15. The **Task Definitions** widget lists several available tasks for the user.

    The number of tasks in the widgets in your lab image might be different from the tasks that are listed in your lab image.



16. Switch to the **Work on tasks** page.

    On this page you can access inbox and task queues, postpone tasks and resume them from the backlog, work on tasks, and complete them. You can also attach notes, create subtasks, set and change priorities and due dates, see and trigger escalations for tasks, and see the process status that is related to the current task.

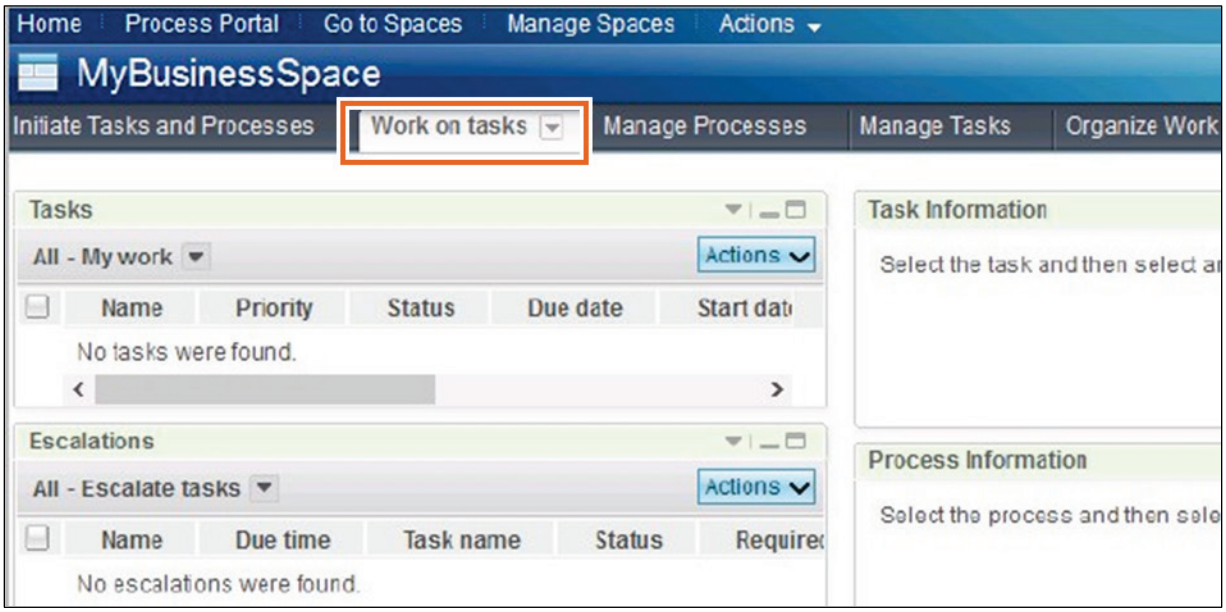


17. Note the different widgets that are available on the page.

    The four widgets that are displayed are:

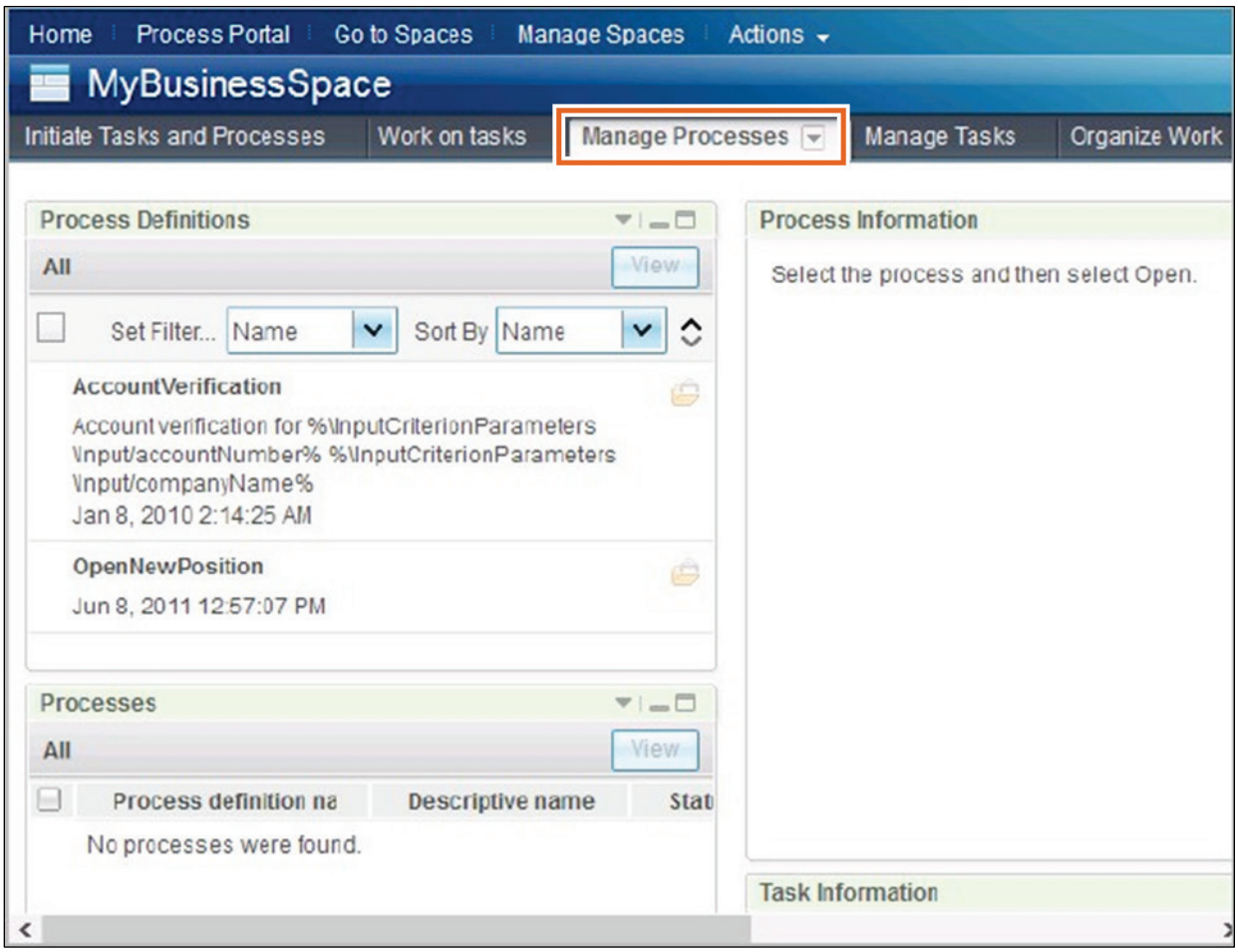
    - **Tasks**: Displays the tasks that you own.
    - **Task Information**: Displays information about a task that you selected in the Tasks widget or the Process Information widget. This widget is the place where you do the work on the tasks that you own.
    - **Escalations**: Displays the escalations for a specific task. Depending on how the Escalations widget is configured, you can also assess the escalation status of the tasks that you own.
    - **Process Information**: Displays information about the process that is associated with the task.

    All the widgets on this page are empty. Depending on the other lab exercises you completed earlier, the screen capture might not match with your lab image. Feel free to drag the border of any widget to resize the page display.



18. Click the **Manage Processes** page.

    On the **Manage Processes** page, you can drill down from process definitions to processes, filter, and sort process instances. You can also see process details and associated tasks with all task details, and act on processes and tasks.



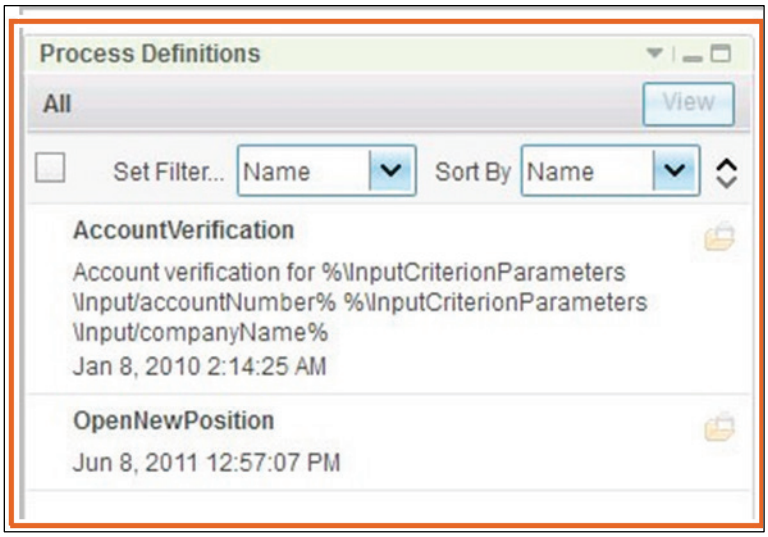
19. Note the different widgets that are available on the page.

    The five widgets that are displayed are:

    - **Process Definitions**: Displays all process definitions that are available in the runtime system
    - **Process Information**: Displays information that is associated with the process definition
    - **Processes**: Displays a list of processes
    - **Tasks**: All the available user actions for this widget are enabled
    - **Task Information**: Displays information that is associated with the task

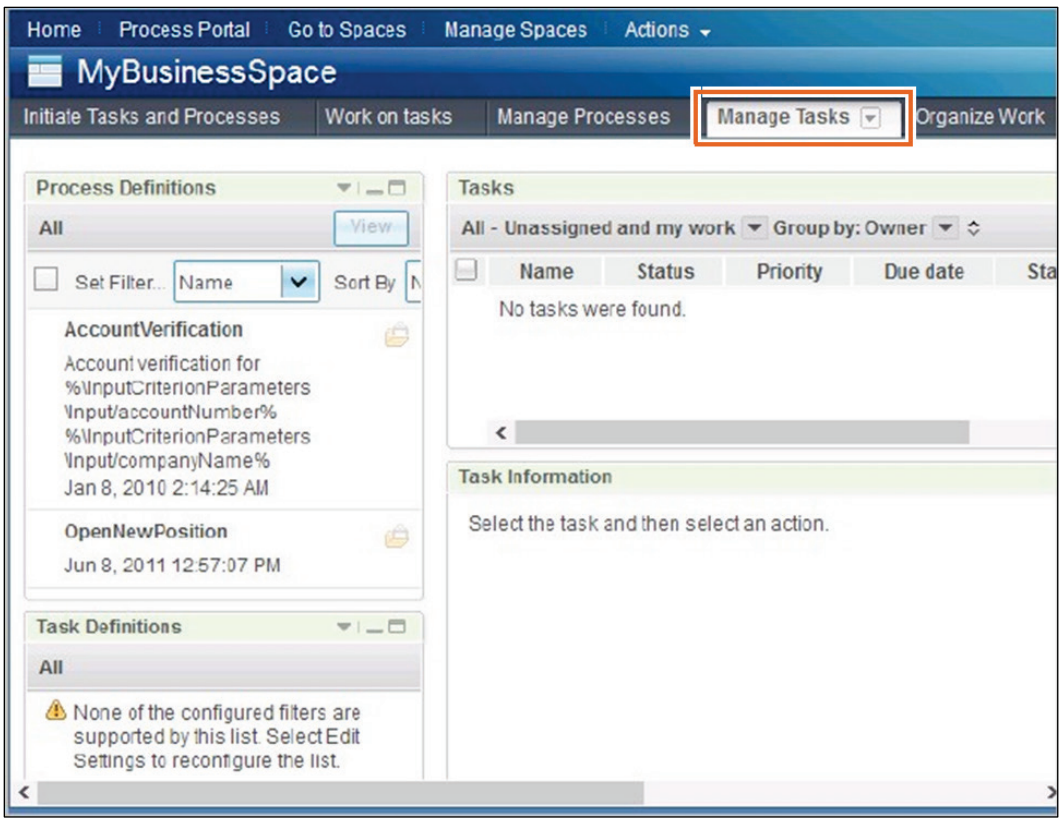
20. Verify that the **AccountVerification** process is listed in the **Process Definition** widget.

    This process is available on the server.



21. Click the **Manage Tasks** page.

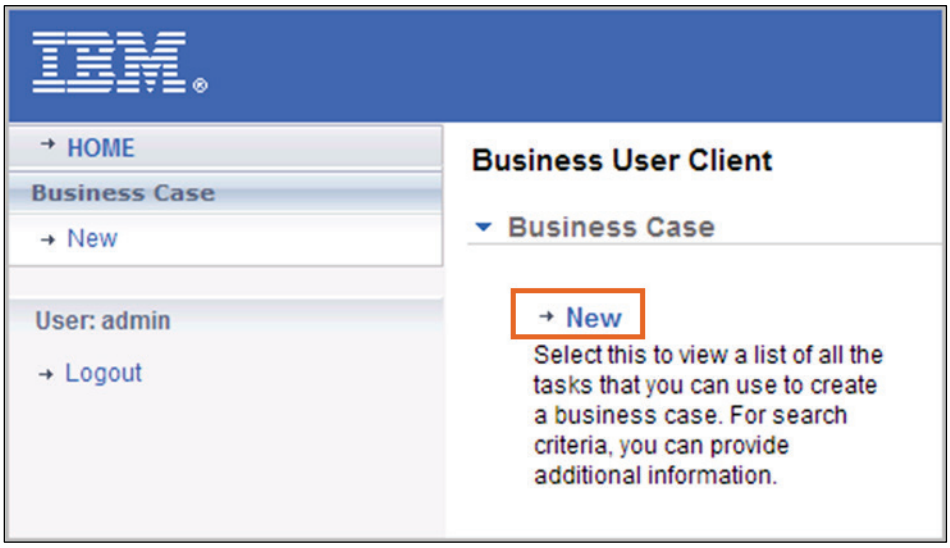
    On the **Manage Tasks** page, you can drill down from process and task definitions to filter, sort, and other tasks. You can also see task details, trigger related escalations, related process status, and team information.



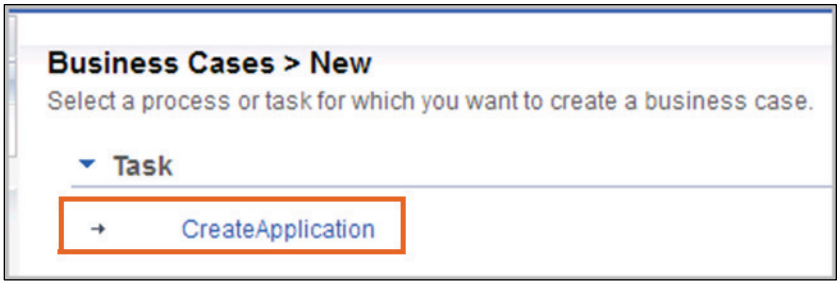
22. Note the different widgets that are available on the page.
    The seven widgets that are displayed are:
    - **Process Definitions**: Displays all process definitions that you can access.
    - **Task Definitions**: Displays a list of all task definitions that you can access.
    - **Tasks**: All the available user actions for this widget are enabled.
    - **Task Information**: This widget can display information about a task that you selected in the Tasks or the Process Information widgets. It can also use the Task Definitions widget to display the form for the task that you are creating.
    - **Process Information**: Displays information that is associated with the task or process.
    - **Team List**: Displays the potential owners for a task that you selected in the Tasks widget.
    - **Escalations**: Displays the escalations for a specific task.

    The Tasks widget on this page is empty. Later, you create a task instance and you submit it with this widget.
23. Minimize the browser.
24. Open a new instance of the browser window.
25. Type the following address in the location bar (note the case):
    `https://localhost:9443/AccountOpeningUI/Index.jsp`
26. If prompted for login, type `admin` in the **Name field** and `web1sphere` in the **Password** field, and click **Login**.
    The Business User Client page (`Workplace.jsp`) that is displayed is a JSP in the web project you generated. All pages in the project are customizable.

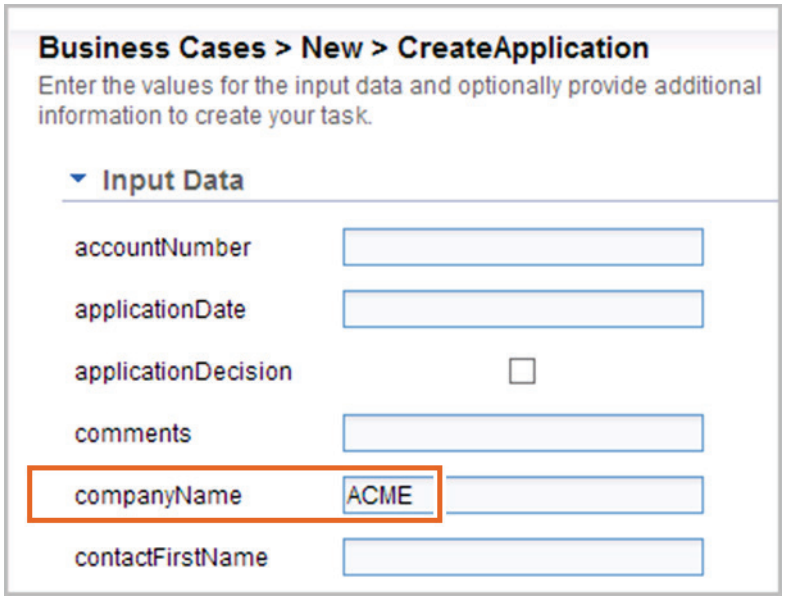
27. Under **Business Case**, click **New** to access the **CreateApplication** invocation task.



28. The only available task is **CreateApplication**. Click the link.



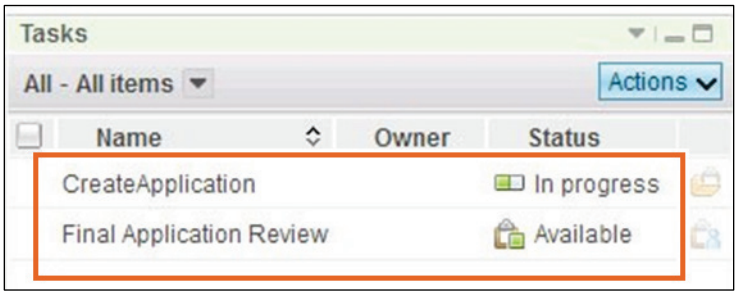
29. On the **CreateApplication** page, in the **Input Data** section, type `ACME` in the **companyName** field. You can leave the remaining fields blank. The `Determine Application Eligibility` Java snippet populates the remaining fields that are based on the `companyName`.



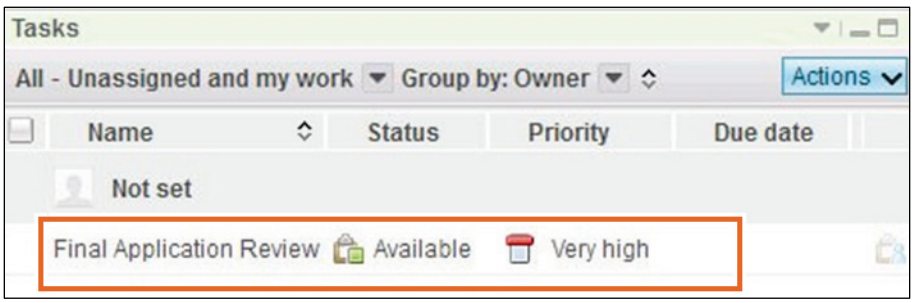
30. Click **Create** at the bottom of the page.
    The web browser returns to the **Business Cases** page.
31. Close the browser.

2. Work with the available task in the MyBusinessSpace space.

1. Switch back to Business Space. You earlier minimized that browser instance.
2. Click the **Manage Processes** page.
3. Verify that the **CreateApplication** and **Final Application Review** tasks are listed in the Tasks widget. The **CreateApplication** task has a status of **In progress** and the **Final Application Review** task has a status of **Available**.
   It is OK if more tasks are listed in the widget. It might be necessary for you to scroll down to see it.

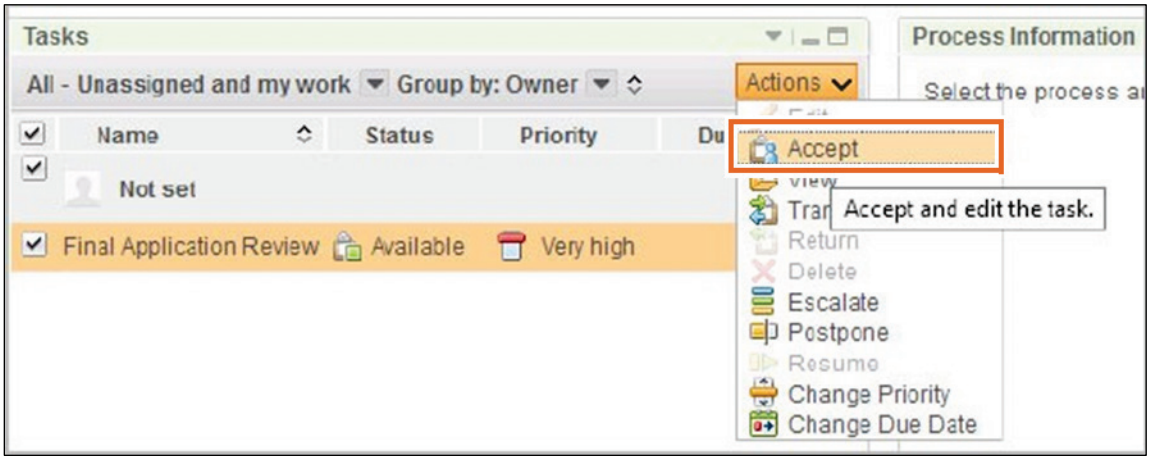


4. Switch to the **Manage Tasks** page.
5. Verify that the **Final Application Review** task is listed in the Tasks widget. Notice that the status is **Available**.

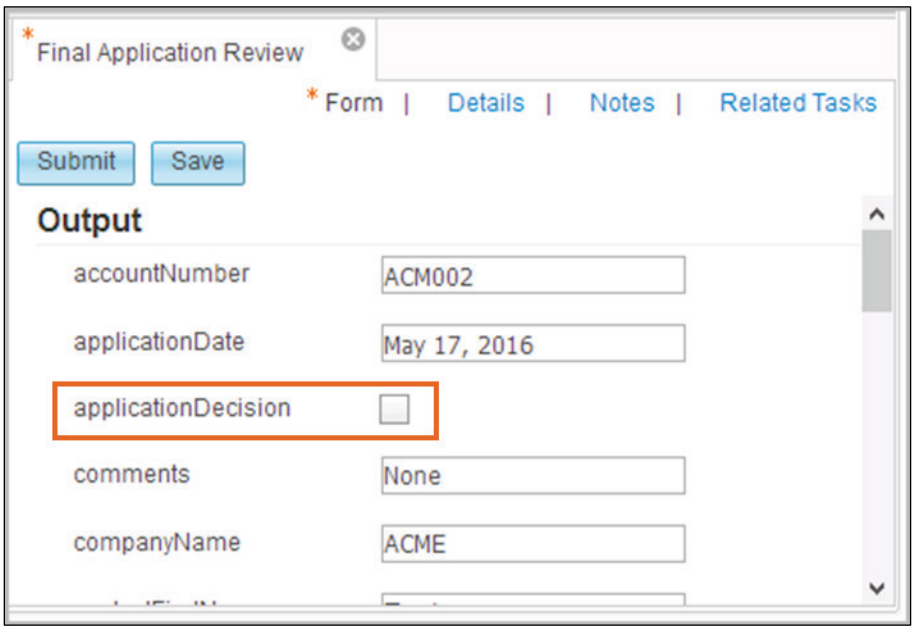


6. Hover over the **Final Application Review** task to reveal a check box to the left of it.

7. Select the check box and click **Actions > Accept** from the widget menu options.

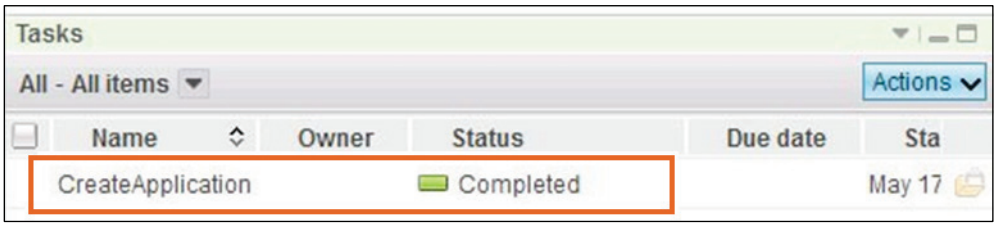


8. Scroll down to the **Task Information** widget. In the **Task Information** widget, click the **Output** form to bring it into focus, and clear the **applicationDecision** check box (to set it to `false`).



9. Click **Submit** at the top of the Task Information widget.
10. Verify that the Tasks widget is now empty because the task is complete.
11. Switch to the Manage Processes page and scroll down to the **Tasks** widget.

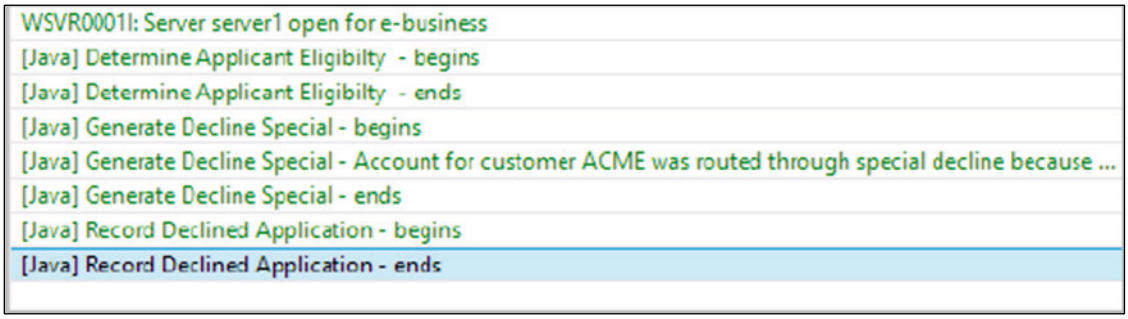
12. Verify that the Final Application Review task is no longer listed. Also, confirm that the **CreateApplication** task has a **Completed** status now.



3. Verify the path that the application took through the remainder of the business process by examining the messages in the Server Logs view.

1. In IBM Integration Designer, switch to the **Server Logs** view.
2. The following messages in the server log confirm the path through the business process.

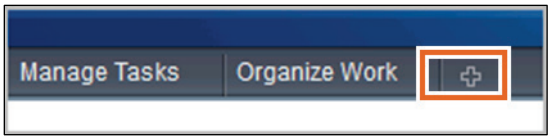
```
Generate Decline Special - begins
Generate Decline Special - Account for customer ACME was routed
through special decline because the credit risk was MED
Generate Decline Special - ends Record Declined Application - begins
Record Declined Application - ends
```



3. If time permits, you can run the test again with **applicationDecision** set to `true`. For this test case, Generate Decline is not invoked so the Generate Decline Special messages are not listed in the Server Logs view.

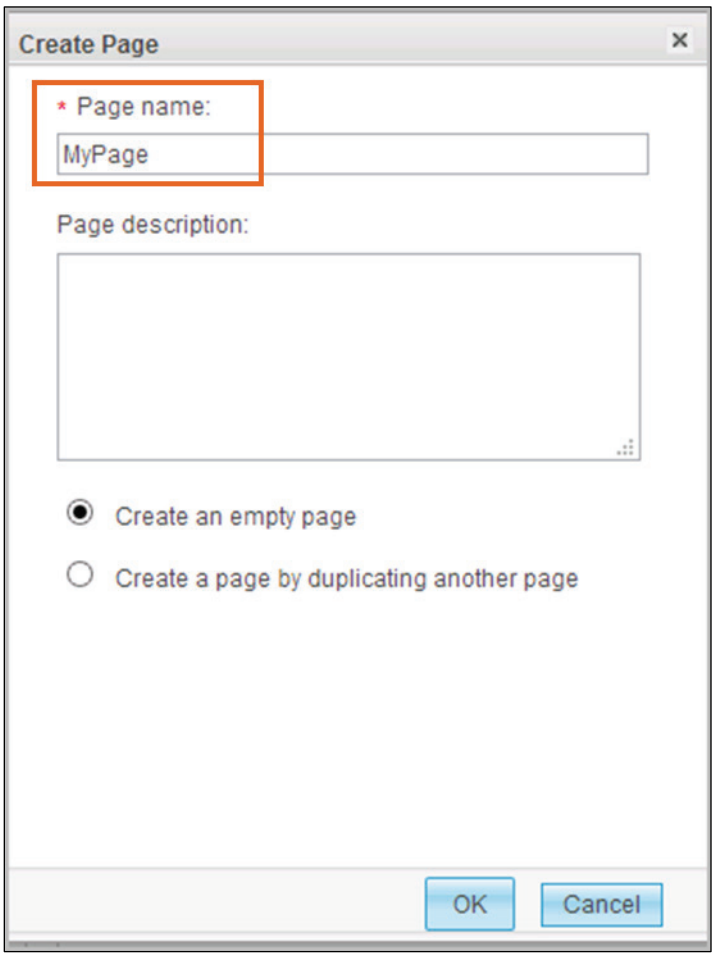
4. Create a page in MyBusinessSpace.

1. Switch back to the Business Space.
2. Create a page by clicking the **+** icon next to the **Organize Work** page.

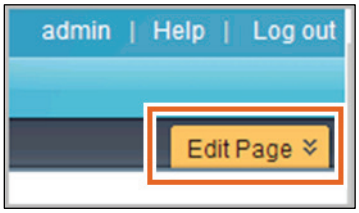


3. Enter MyPage in the **Page name** field, accept the remaining defaults, and click **OK**.

   An empty page is displayed.



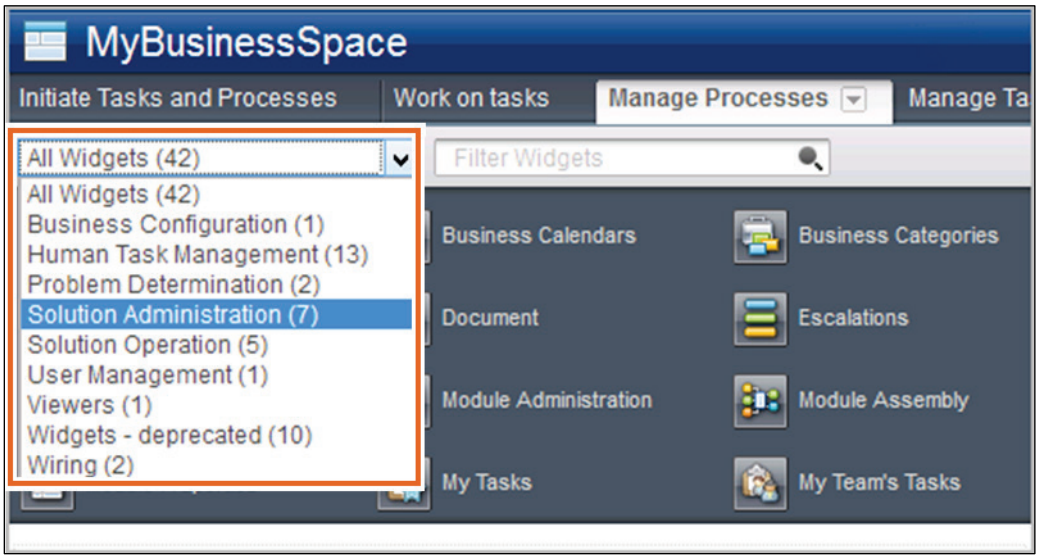
4. Click **Edit Page** at the upper-right corner of the page. After clicking, the text changes from **Edit Page** to **Finish Editing**.



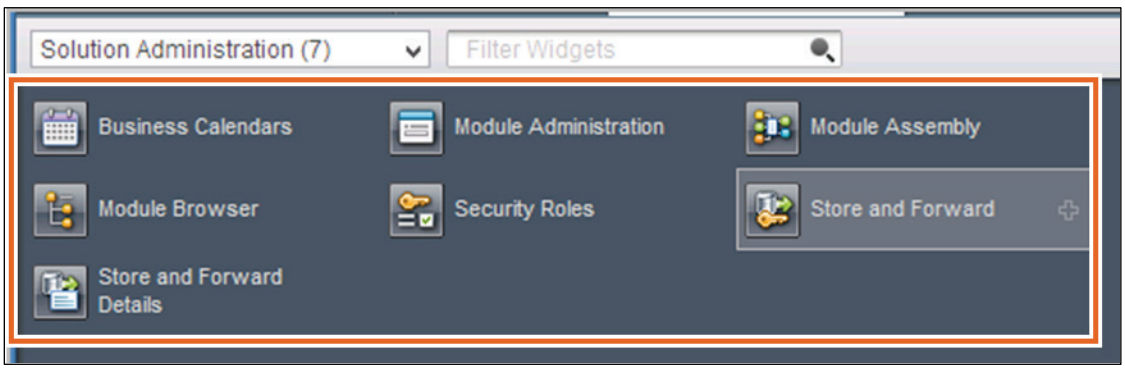
5. Confirm that a list of available widgets is displayed. Several widgets are listed. You can scroll down to view the entire list.

5. Add widgets from the Solution Administration template.

1. Select **Solution Administration** from the widgets list.

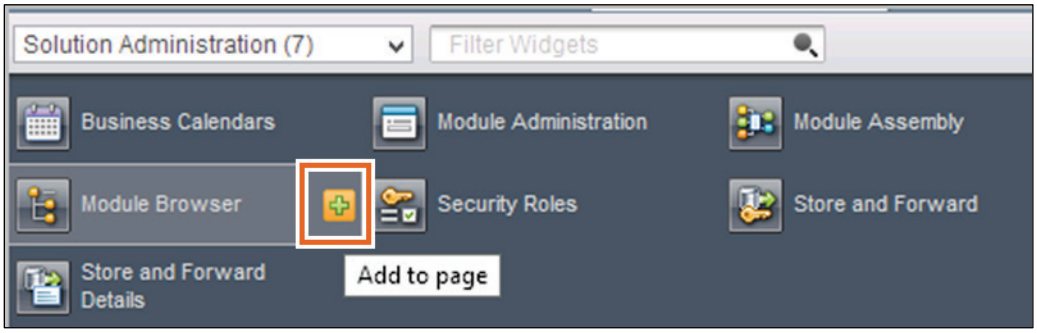


The list is filtered to display seven widgets.



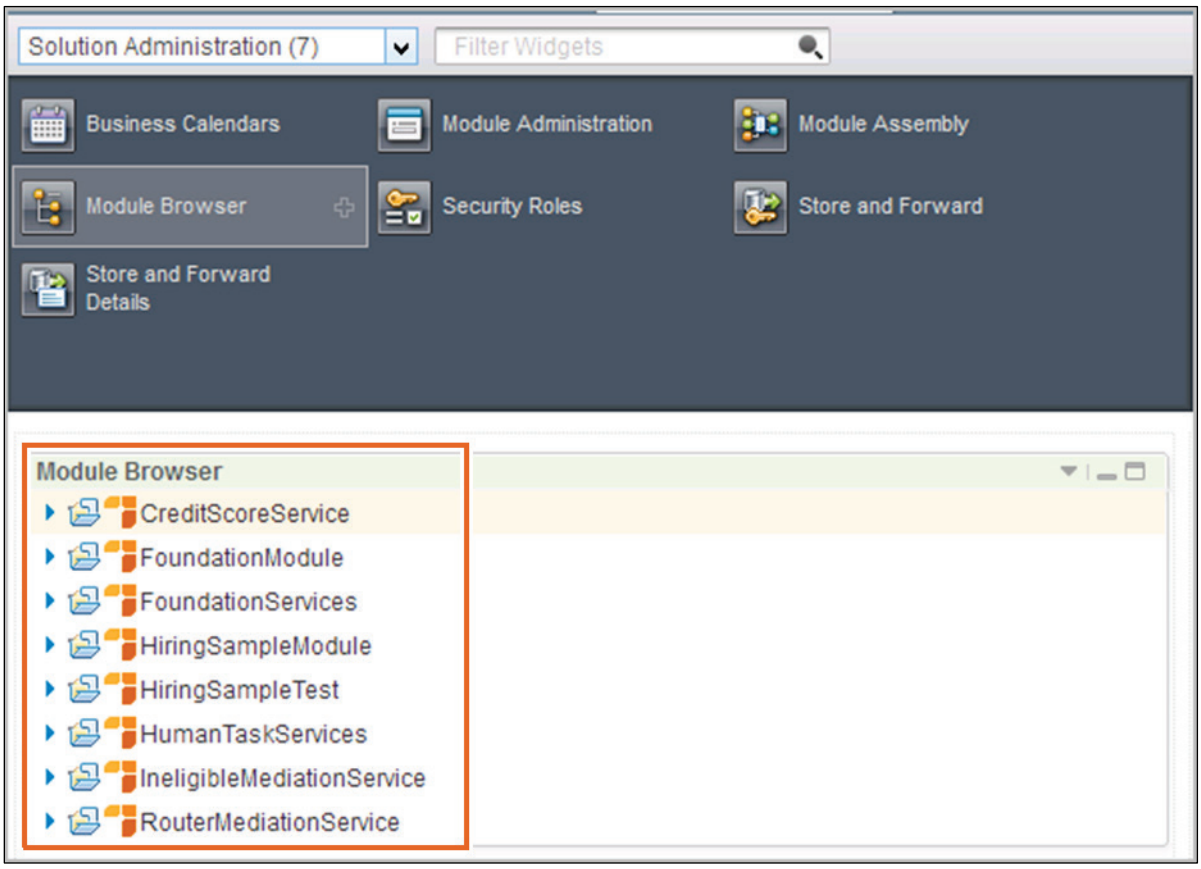
The Solution Administration widget provides access to widgets from which you can view and administer the modules and artifacts in your solution.

2. Click the **Add to page** icon for the **Module Browser** widget.

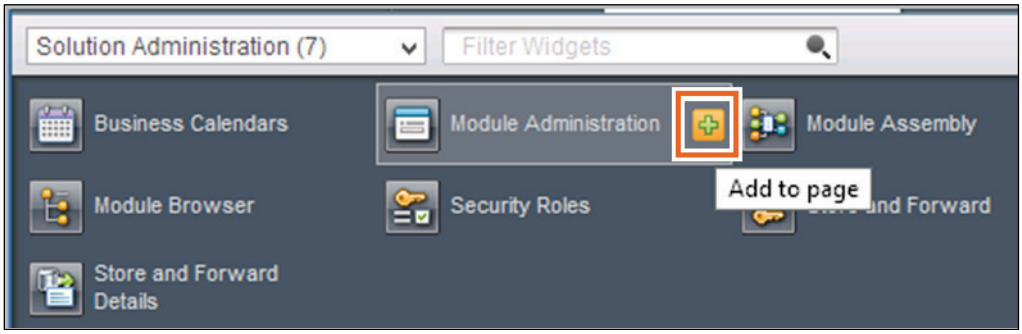


3. The newly added widget is displayed in **MyPage**.

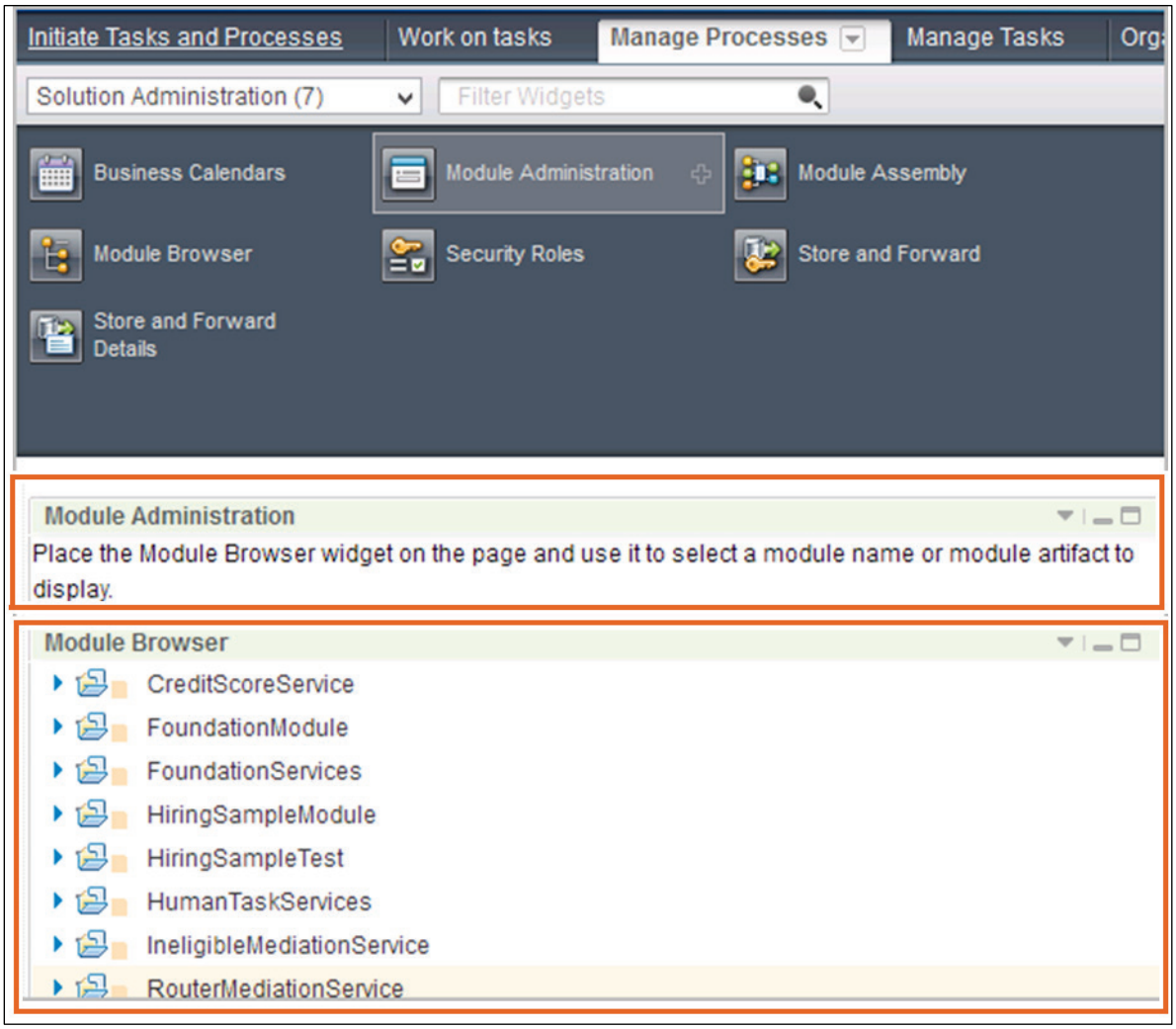
   It lists all the Service Component Architecture (SCA) modules that are deployed on the server.



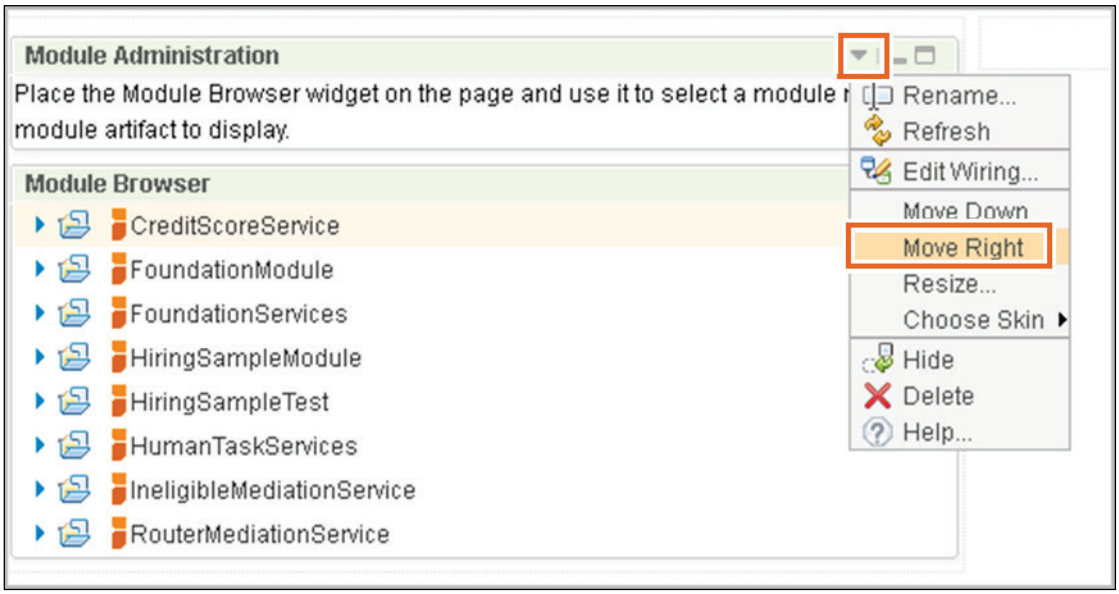
4. Click the **Add to page** icon for the **Module Administration** widget.



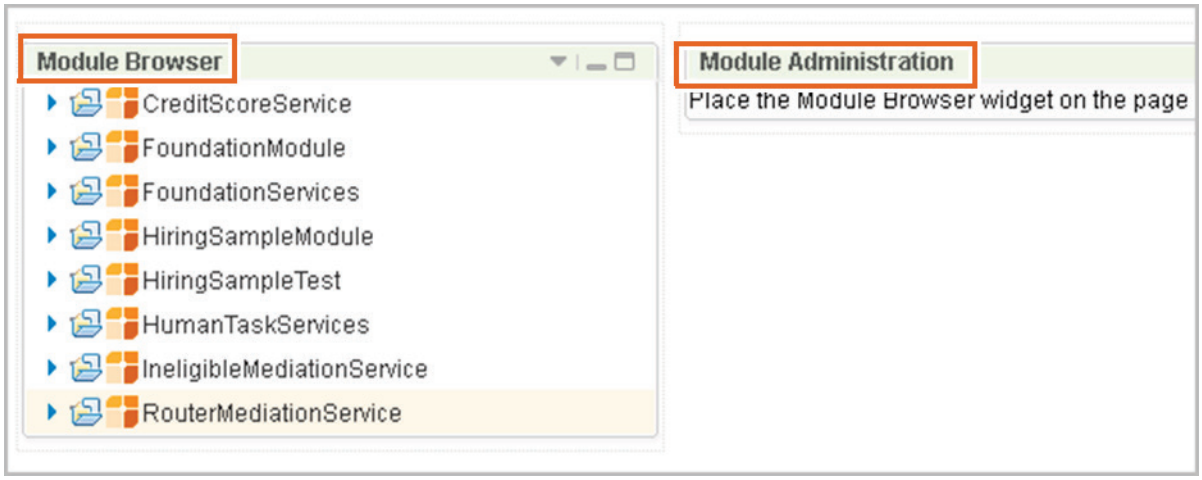
5. Verify that the **MyPage** page currently lists two widgets.
The Module Administration widget currently displays no data.



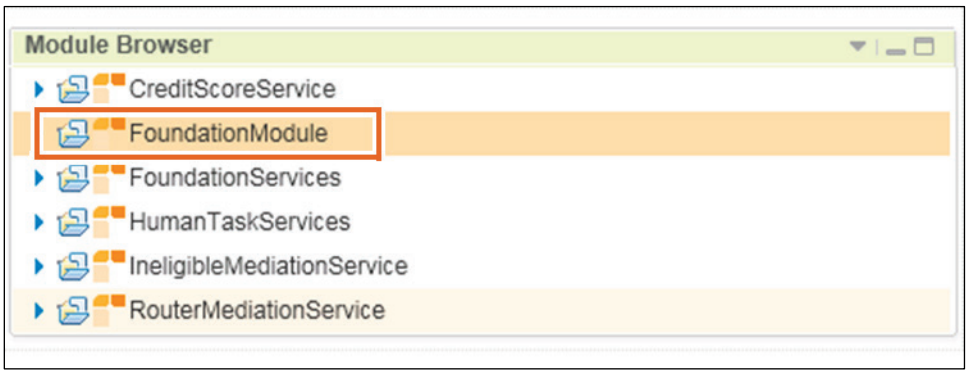
6. You can rearrange the two widgets by clicking the down arrow icon in the upper-right corner of the Module Administration widget and clicking **Move Right**.



7. The Module Administration widget is now displayed on the right of the page.



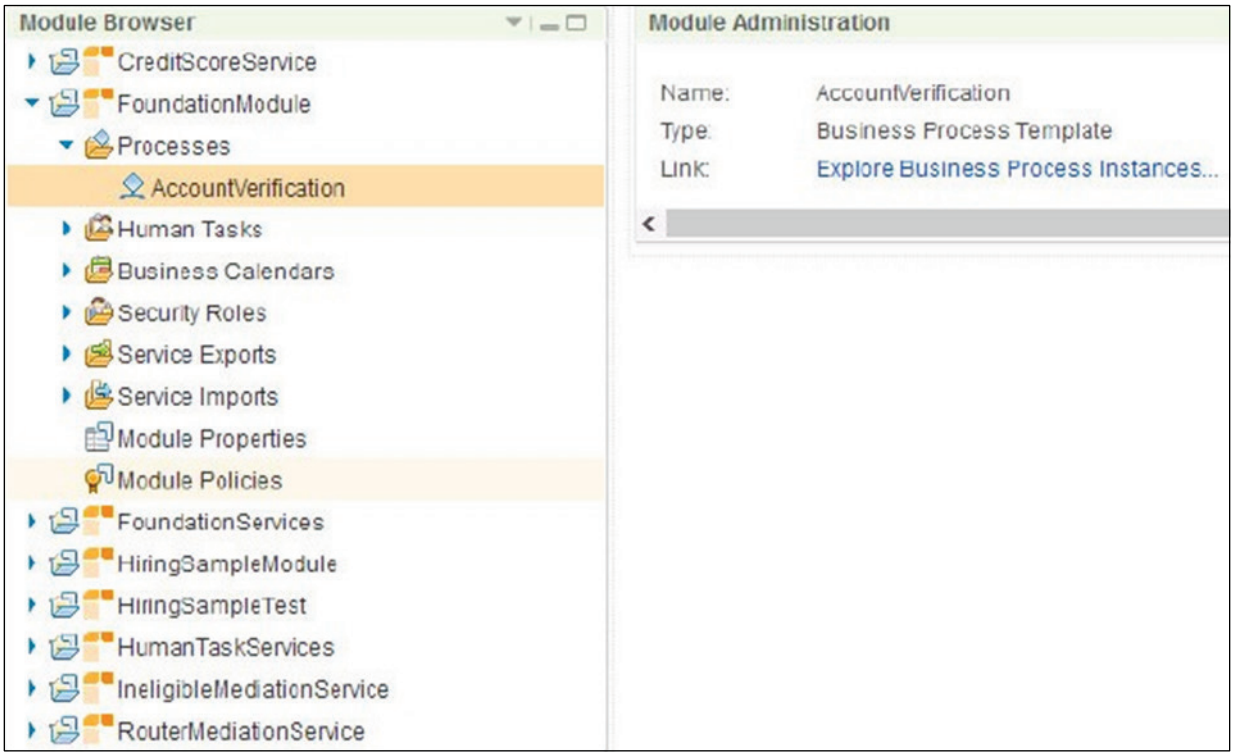
8. Click **FoundationModule** in the Module Browser widget.



The details of FoundationModule are displayed in the Module Administration widget. You can hover over the objects in the widget to see more details.

9. Expand **FoundationModule > Processes** and click **AccountVerification** in the Module Browser widget.

   The process information is displayed in the Module Administration widget. Clicking **Explore Business Process Instances** starts Business Process Explorer in a separate browser. Do not click that link.



10. Feel free to click other artifacts and modules in the Module Browser widget to view the details in the Module Administration widget.
11. Feel free to explore other available widgets by selecting different options in the widget list. When you are done exploring, click **Finish Editing**.
12. Click the **Logout** link to log out of the Business Space.
13. Close the browser window.

6. Remove the applications from the server/

1. In the Servers view, right-click IBM Process Server v8.6 at localhost and click Add and Remove Projects from the menu.
2. Click **Remove All** and click **Finish**.
3. Close IBM Integration Designer.

Results:
In this exercise, you created a business space and worked with several widgets to view displayed data.

IBM Training

Advanced testing

IBM Business Process Manager V8.6

IBM Training

Unit objectives

- Describe the advanced testing facilities that are available in IBM Integration Designer, including the Component Test Explorer and cross-component trace
- Describe the integration debugger
- Define the purpose and function of the serviceDeploy tool
- Describe how to use serviceDeploy in single-developer and multiple-developer environments

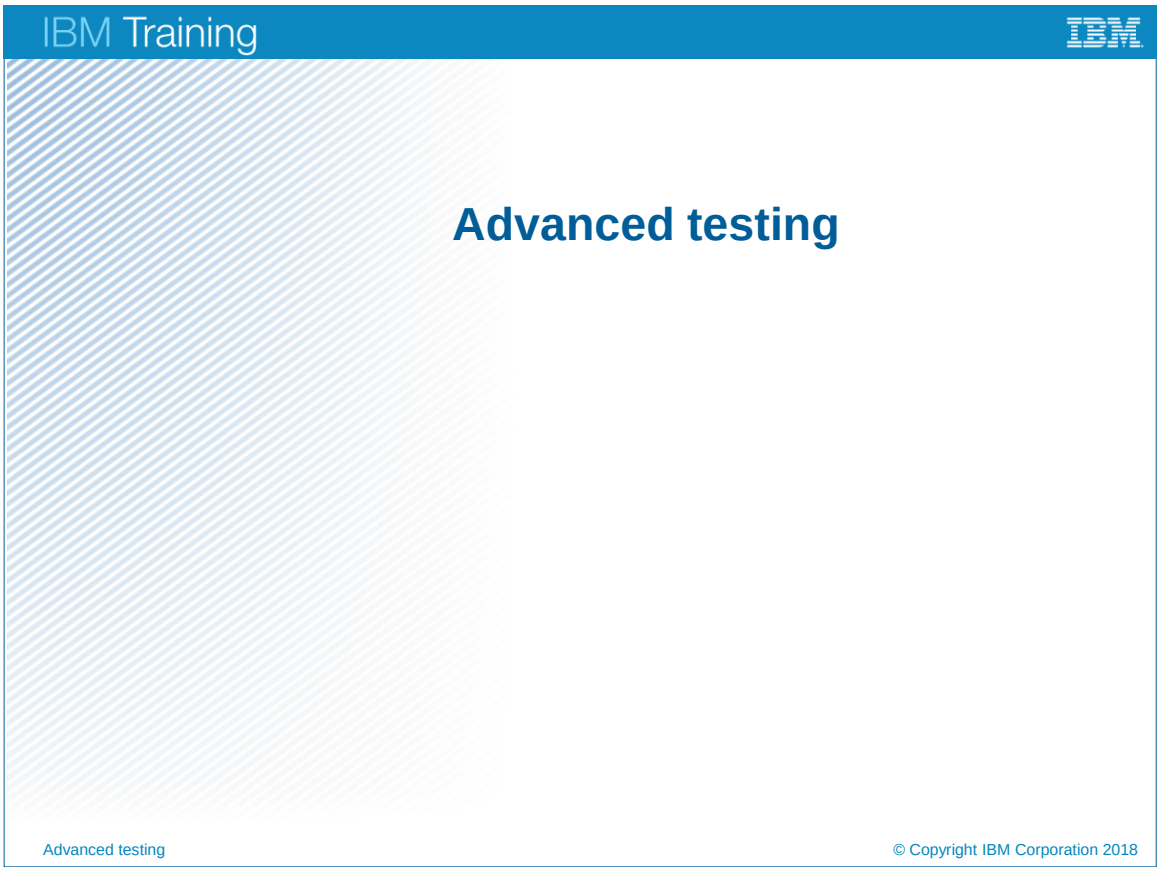
Unit objectives

IBM Training

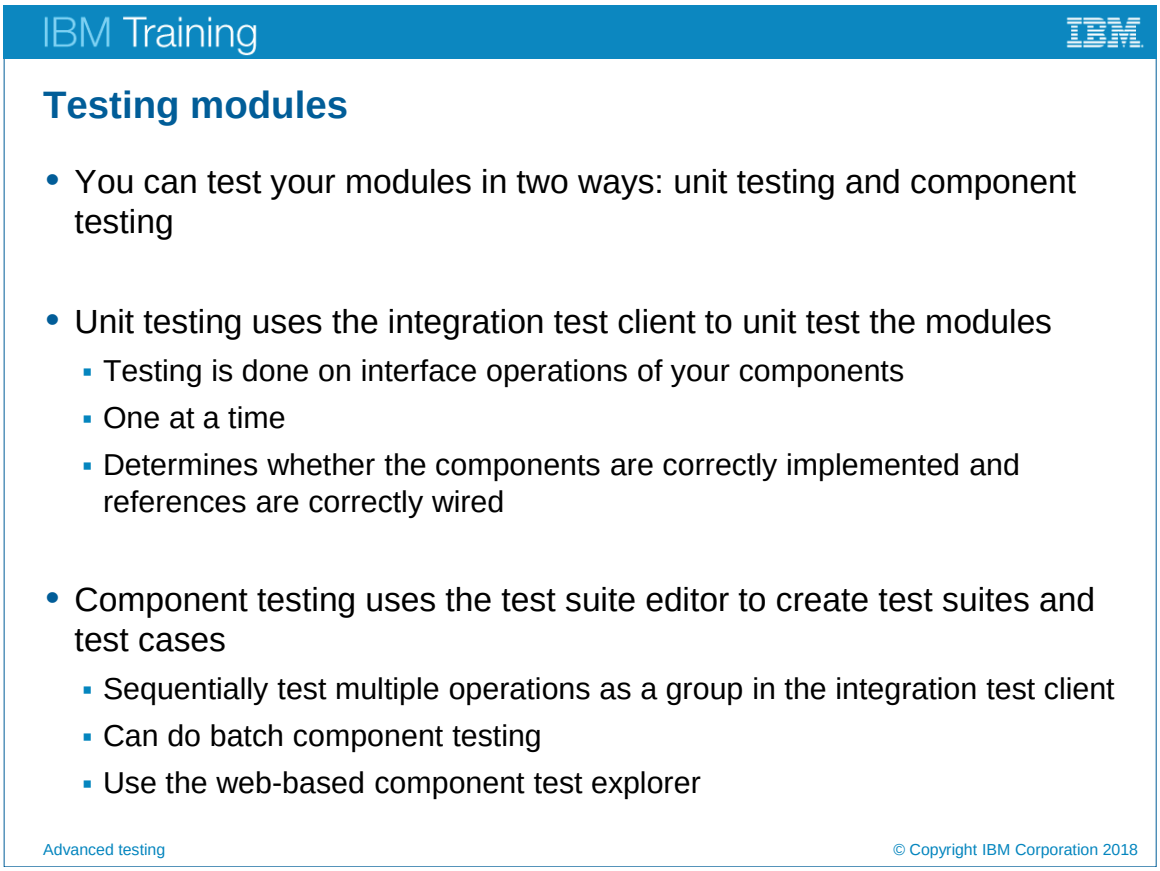
Topics

- Advanced testing
- Deploying integration applications with serviceDeploy

Topics



Advanced testing

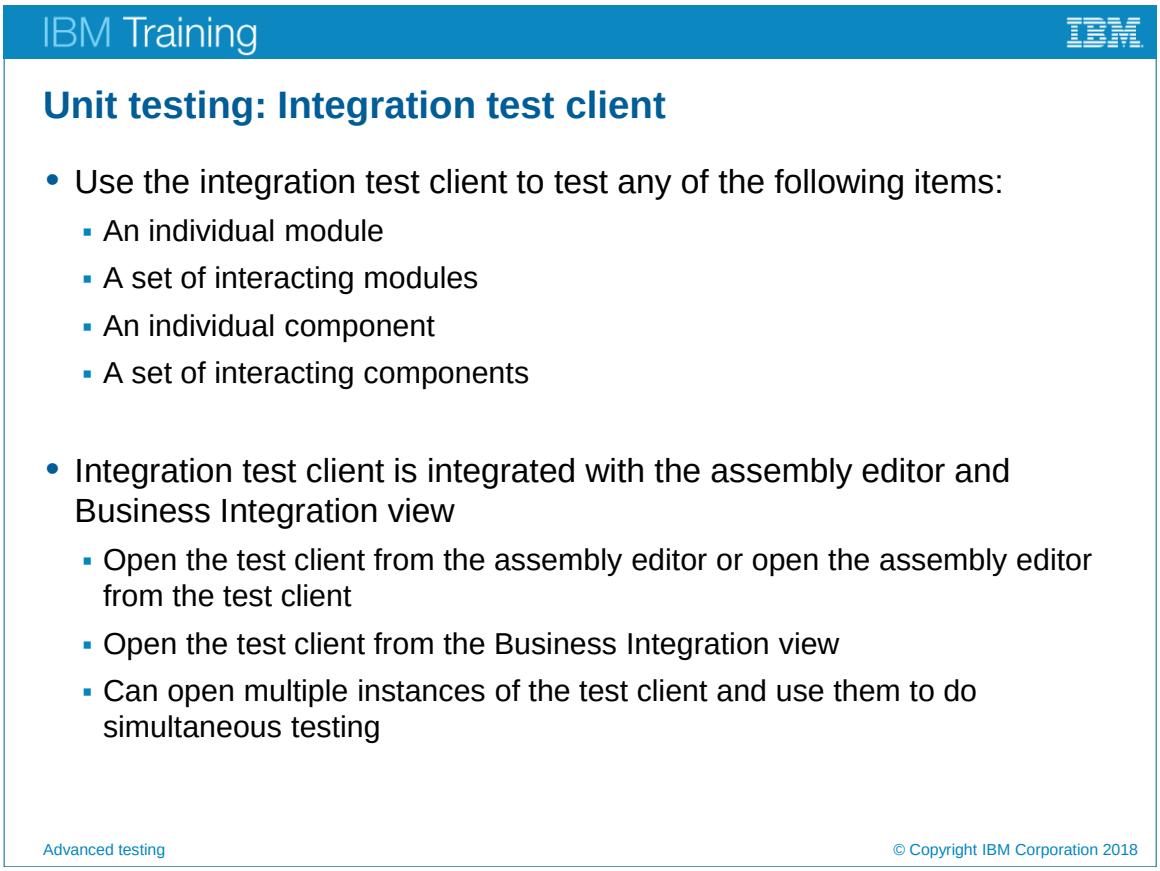


Testing modules

In IBM Integration Designer, you can test your modules in two ways. In unit testing, you choose components and interfaces, and then you test the operations one at a time in the integration test client. In component testing, you use the new test suite editor to create test suites and test cases.

In IBM Integration Designer, you can use the integration test client to unit test your modules. The testing is generally done on the interface operations of your components, which you can use to determine whether the components are correctly implemented and the references are correctly wired.

By using the test suite editor and associated wizards, you can create and define test cases that consist of one or more operations. You can sequentially test multiple operations as a group in the integration test client. You can also do batch component testing on either a test environment server or a stand-alone server by using test scripts or the user interface of the web-based Component Test Explorer.



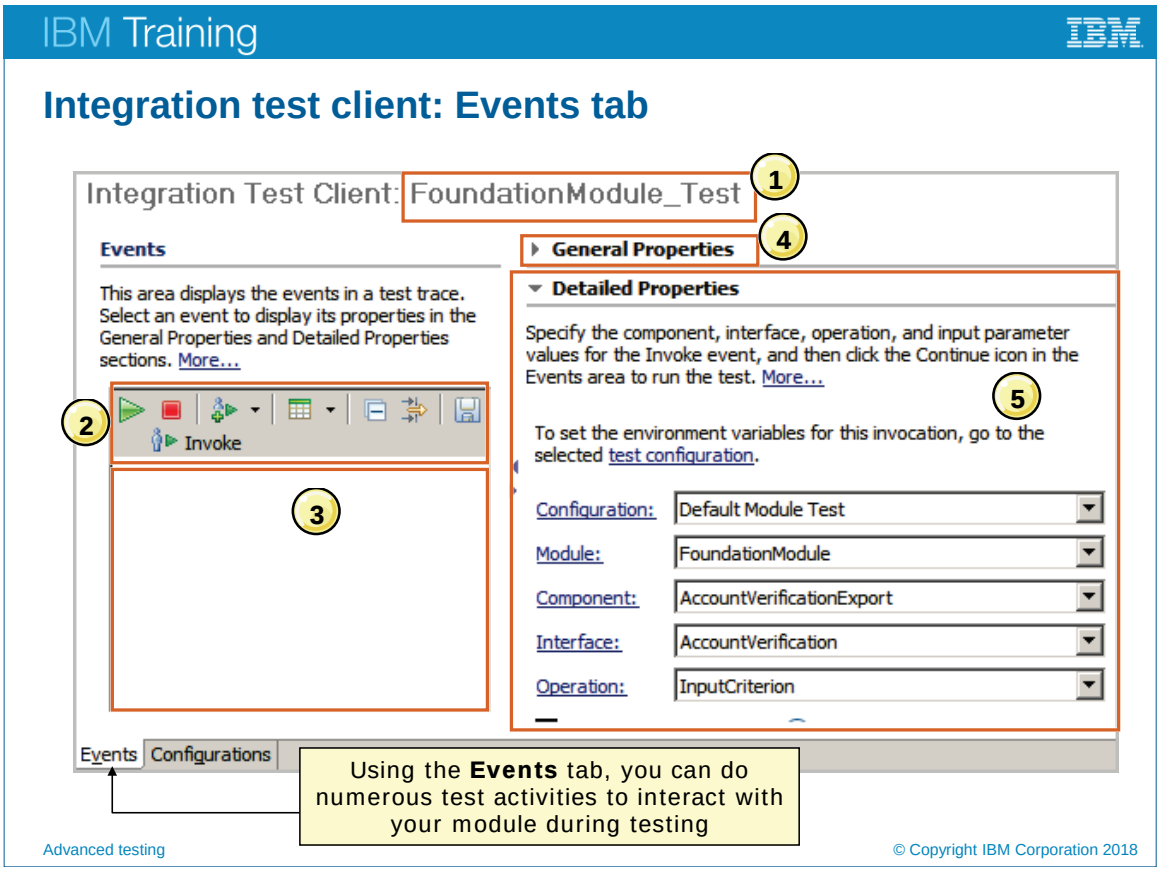
Unit testing: Integration test client

In IBM Integration Designer, you can use the integration test client to unit test your modules. The testing is generally done on the interface operations of your components, which you can use to determine whether the components are correctly implemented and the references are correctly wired.

Use the integration test client to test any of the following items:

- An individual module
- A set of interacting modules, including modules that are part of a process application or toolkit
- An individual component
- A set of interacting components

The integration test client is fully integrated into the workbench so that you can go through the Business Integration view and other views while you are using the test client. It is also closely integrated with the assembly editor. You can open the test client from the assembly editor, and you can open the assembly editor from the test client. Although the assembly editor is considered the primary starting point for the integration test client, you can also open the test client from the Business Integration view. Regardless of whether you open the integration test client from the assembly editor or the Business Integration view, you can open multiple instances of the test client and use them to do simultaneous testing.



Integration test client: Events tab

In IBM Integration Designer, the integration test client is the designated tool for testing modules and components. In the test client, you can manage and precisely control your tests.

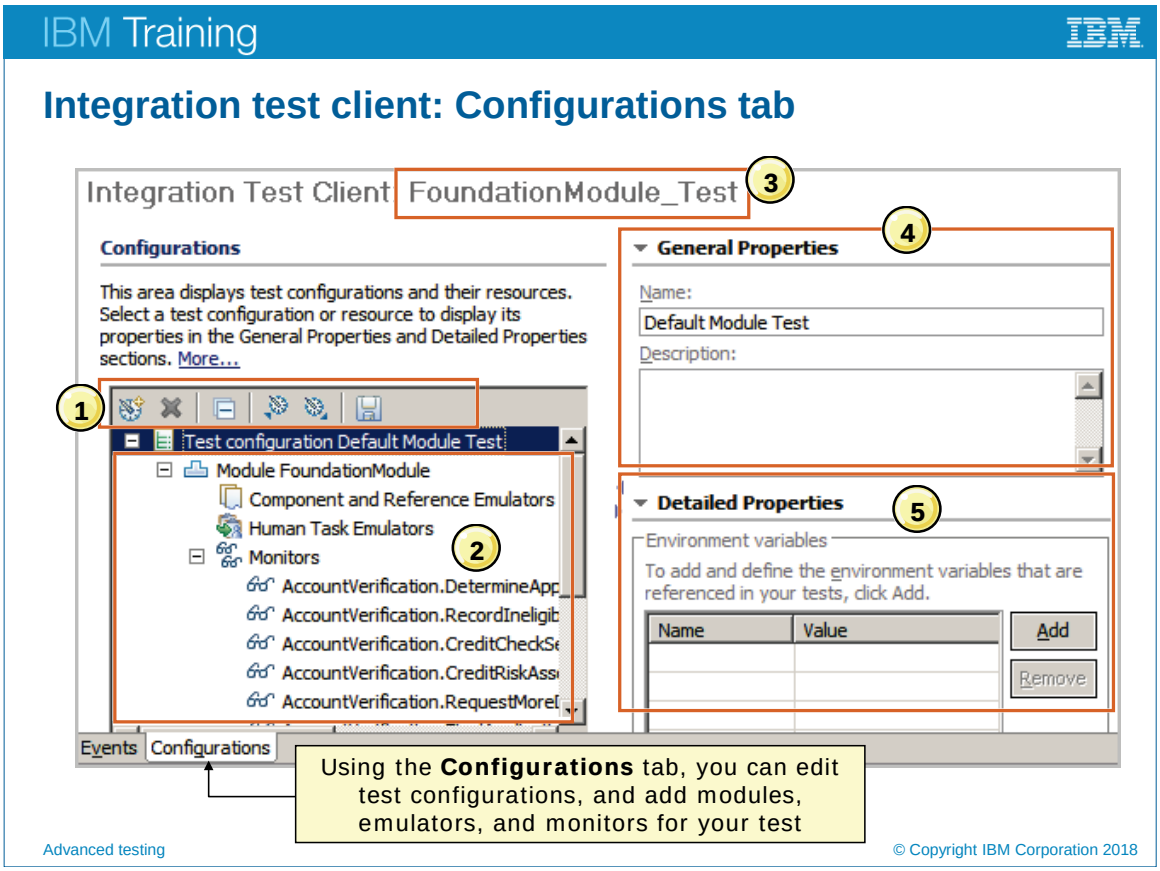
The test client interface has two main features:

- Events page
- Configurations page

Almost all of the tasks that you can do in the integration test client are either initiated or completed in one of the two pages. The Events and Configurations pages are described in the topics “Events page” and “Configurations page.”

In the Events page of the integration test client, you can do numerous test activities to interact with your module during testing. You can select an operation to test, specify values for the operation, and invoke the operation. The image depicts the following items:

1. Module name that runs the test client
2. Control area that has the following icons:
   - Continue: Initiates the invocation. Depending on the current deployment state of your module, the Deployment Location wizard might open so that you can select a test server.
   - Stop: Detaches the integration test client from the server. All running operations that are waiting on user input from the test client are terminated. All other running applications continue until they terminate, but no status is reported in the test client.
   - Invoke: Generates an Invoke event in the Events area, which you can use to select an operation, specify values for the operation, and invoke and test the operation.
   - Data pool: In the Detailed Properties area of the Events page, any values that you specify for an operation, manual emulation, or event definition in the value editor can be saved to the data pool. The data pool icon opens the data pool editor, which you can use to view, edit, select, and use the saved data pool values.
   - Filters: Opens the Event Filter window, from which you can choose an event filter that suppresses the display of selected event types.
   - Save: Opens the Save Test Trace window, with which you can save the test trace.
3. The Events area displays a hierarchical test trace for the events that are generated during a test. These events are typically grouped under a top-level event. For example, if you are testing an operation, the events are nested together under the Invoke event that was used in the invocation of the operation.
4. The General Properties area shows the time that an event that is selected in the Events area was generated.
5. The Detailed Properties area displays the specific properties of any event that is selected in the Events area. For some events, the specific properties can include information about the resources that are associated with the selected event, such as the name of the test configuration and module that is used in the test. If you are testing a module that is contained in a process application or a toolkit, the name of the process application or toolkit is added to the name of the module as a prefix.

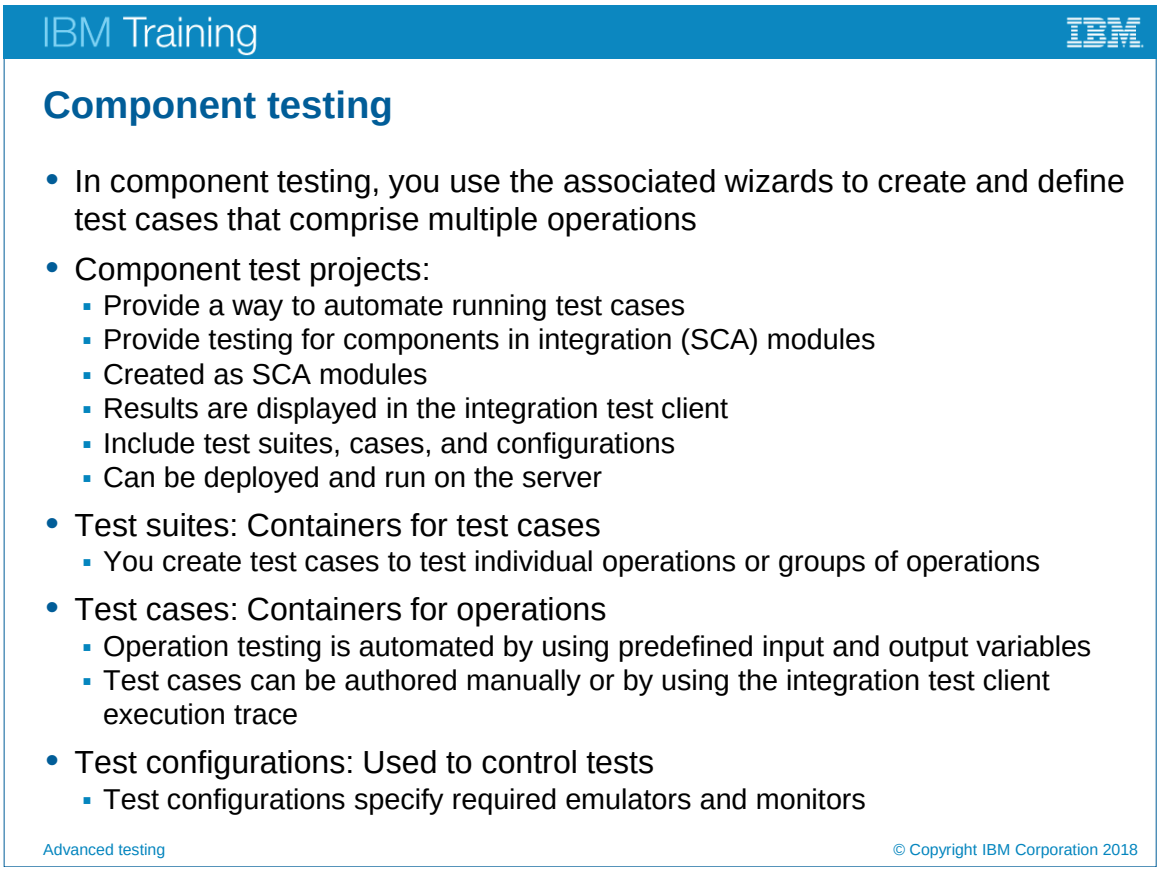


Integration test client: Configurations tab

In the Configurations page of the integration test client, you can edit the default test configuration or you can create and edit new test configurations. You can add modules to your test configurations, or add emulators and monitors to your test configuration modules, to more precisely control your tests.

1. The control area has the following icons:
   - Add: Opens the New Configuration wizard, with which you can add one of the following test configuration elements to the Configurations page:
     - Test bucket configuration
     - Test configuration
     - Test case
     - Process application or toolkit
     - Module
     - Emulator
     - Monitor

2. The Configurations area provides a tree view of your test configurations and test bucket configurations.
    - For any specified test configuration, the following information is shown:
        - The name of the test configuration
        - The name and version of the process application or toolkit
        - The names of the modules in the test configuration
        - The names of any emulators in each test configuration module
        - The names of any monitors in each test configuration module
    - For any specified test bucket configuration, one or more of the following items are shown:
        - The name of the test bucket configuration
        - The name of the test suite in the test bucket configuration
        - The names of the test cases in the test suite
        - The name of the test configuration in the test suite
        - The name and version of the process application or toolkit
        - The names of the modules in the test configuration
        - The names of any emulators in each test configuration module
        - The names of any monitors in each test configuration module
3. The module name that is running the test client.
4. The General Properties area provides the name and description (if any) of any test configuration element selected in the Configurations area.
5. The Detailed Properties area displays the specific properties of any test configuration element that is selected in the Configurations area. For example, if a test configuration is selected in the Configurations area, the Detailed Properties area displays the names of any modules that are contained in the test configuration.



Component testing

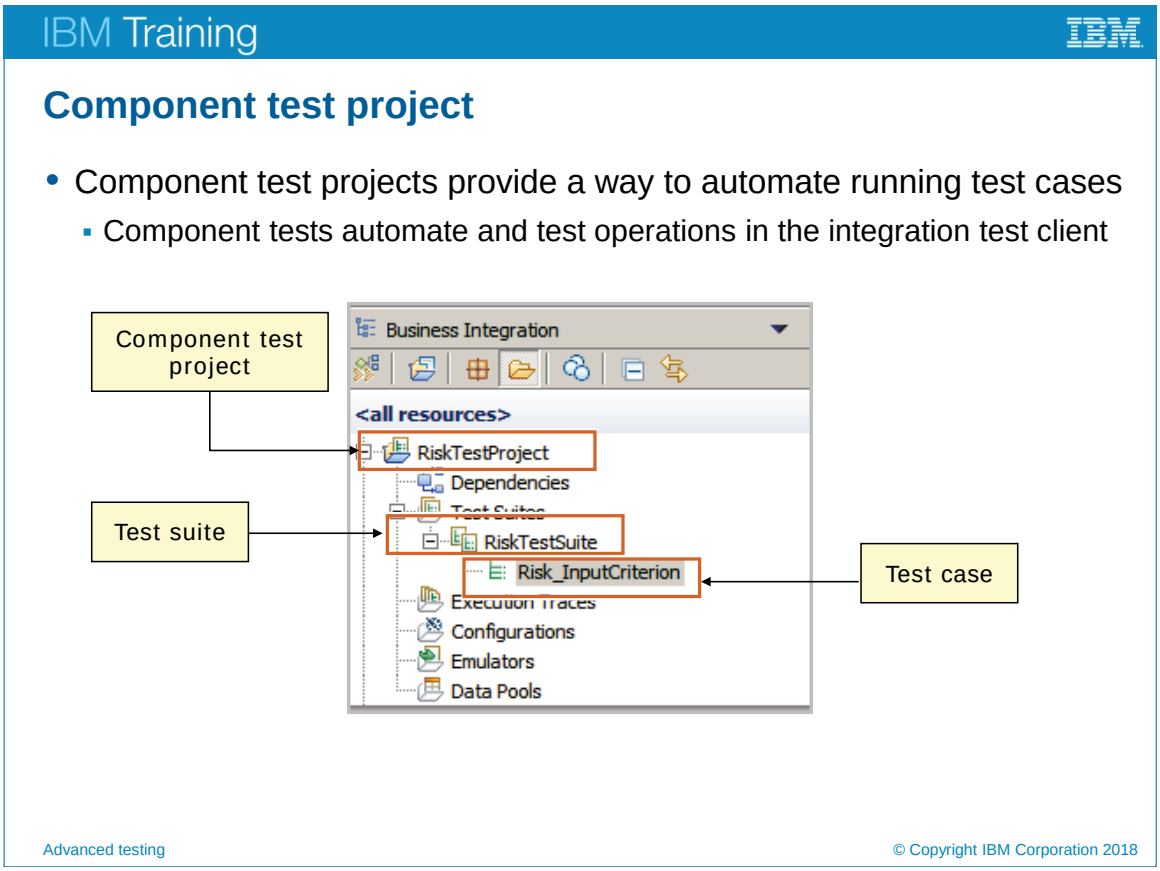
In component testing, you use the test suite editor and associated wizards to create and define test cases that comprise multiple operations. By using the test suite editor, you can automate and simultaneously test the operations in the integration test client. The component test project that you create is a module that is deployed to the IBM Process Server test environment along with the module that is being tested. Component test projects can be invoked from an Ant script.

Component test projects are containers for test suites. Test suites are containers for test cases. Test cases are containers for operations, and test configurations are used to control your tests.

- **Component test projects**: Generally, creating a component test project to contain your test suites is the first step in preparing to do component testing.
- **Test suites**: After you create a component test project, you generally create a test suite to contain your test cases. When you create a test suite, you can choose from one of the following test patterns:
  - **Operation-level testing**: In operation-level testing, a separate test case is created for each operation that you select for component testing.
  - **Scenario-based testing**: In scenario-based testing, a single test case is created for all of the operations that you select for component testing.

- **Test cases**: You can think of a test case as a container for multiple operations that you select for testing. You use test cases to automate and simultaneously test the operations in the integration test client. Test data for test cases is defined in the test data table, which contains a set of named variables that can be used for either input or output in the test case invocations.
- **Test configurations**: A test configuration specifies one or more modules to test. Each of these modules can include zero or more emulators for components or references in the module and zero or more monitors for the wires in the module. When you open the test suite editor, a default test configuration is automatically created that you can use for testing. The default test configuration is often all that you need for testing your test cases. However, you can choose to add modules to your test configuration, or add emulators and monitors to your test configuration modules.
- **Test suite editor**: The test suite editor is the designated tool for editing test suites. It features a rich user interface that you can use to easily manage your test suites and test cases. The test suite editor is designed to closely resemble the test client, which helps you easily transition from using one tool to the other.
- **Top-down testing of test cases**: In component testing, you use the test suite editor and associated wizards to create and define test cases that comprise multiple operations. You can use the test suite editor to automate and simultaneously test the operations in the integration test client. In top-down testing, you define test cases by selecting components and operations for testing, and then you run the test cases in the integration test client. (In bottom-up testing, you use integration test client invocations to define test cases.)

Component test projects are deployable SCA modules that simplify build automation where a component test project can be built like any other SCA module. In addition, you can add dependent Java libraries easily since component test projects have a dependency editor. By adding Java libraries, you can add your own custom utility functions.



Component test project

Test suites

A test suite is a collection of one or more test cases. When you create a test suite, you can choose from one of the following test patterns:

- Operation-level testing
- Scenario-based testing

In *operation-level testing*, a separate test case is created for each operation that you select for component testing. In *scenario-based testing*, a single test case is created for all of the operations that you select for component testing.

Test cases

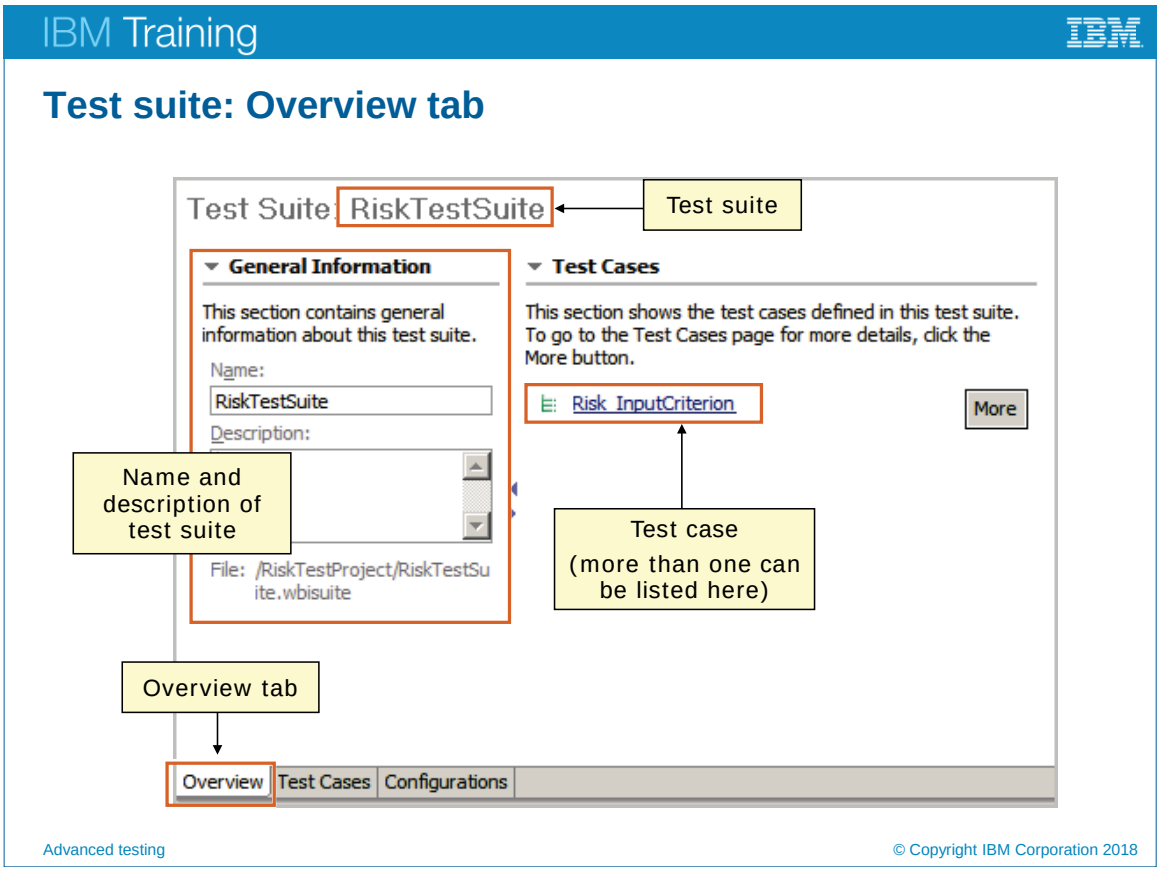
You can think of a test case as being a container for multiple operations that you select for testing. Test cases enable automation and testing of the operations in the integration test client. Test data for test cases is defined in the test data table, which contains a set of named variables that can be used for either input or output in the test case invocations.

Test variations

A test variation is a specific set of variable values for a test case. Although each test case is automatically assigned a default test variation, you can create multiple test variations for a test case that contain a different set of variable values. When a test case is run, all of the test variations for the test case are run unless one of the test variations fails.

Test bucket configurations

A test bucket configuration is a set of specific test suites and test cases that are run together in the same test session.



Test suite: Overview tab

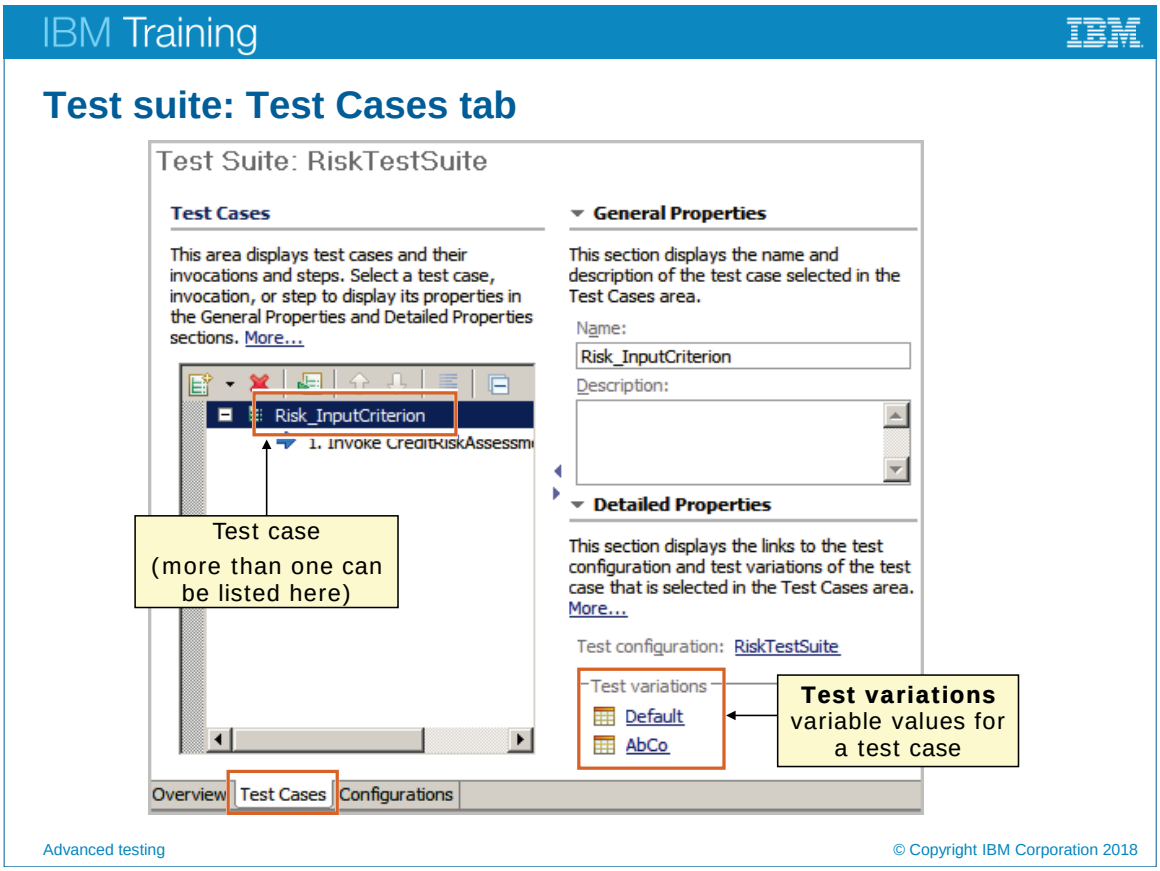
In the Overview page, you can quickly view the essential information for the test suite that is open in the test suite editor.

General Information area

The General Information area displays the name of the test suite, which you can rename by refactoring. It also contains an editable description for the test suite and the path and name of the test suite file.

Test Cases area

The Test Cases area displays the test cases that are defined in the test suite. It also contains a More button that opens the Test Cases page.



Test suite: Test Cases tab

In IBM Integration Designer, the test suite editor is the designated tool for editing test suites. It features a rich user interface that enables easy management of your test suites and test cases. The test suite editor is designed to closely resemble the test client, which helps you easily transition from using one tool to the other.

In the Test Cases tab of the test suite editor, you can do numerous activities to prepare your test suite and test cases for testing. For example, you can select an operation invocation for which you want to specify values in the test data table.

Test Cases area

The Test Cases area displays the test cases, the associated invocation steps, and Wait On steps for the operations that are defined in the test suite.

Control area

The control area provides the following icons to help you manage test cases and operation invocations:

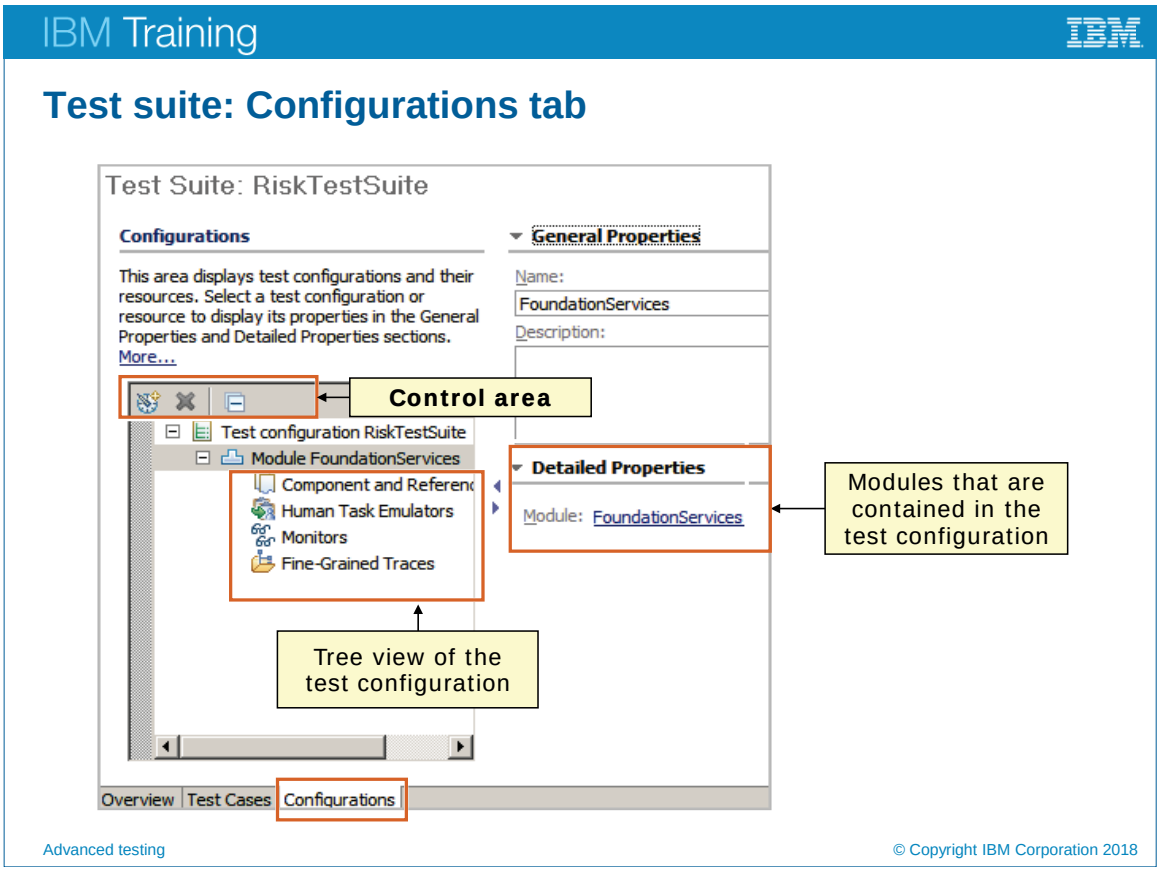
- New test case: Adds an operation invocation for the test case that is selected in the Test Cases area
- Remove: Removes selected test cases or operation invocations from the Test Cases area
- Run test case: Runs the test in the integration test client
- Move Up: Moves up selected test cases or operation invocations in the Test Cases area
- Move Down: Moves selected test cases or operation invocations down in the Test Cases area
- Show Descriptions: Toggles selected test cases or operation invocations between the default description and the user-defined description

General Information

The General Information area shows name and description for a selected test case or the description for a selected operation invocation.

Detailed Properties

The Detailed Properties area displays the specific properties of any test case or operation invocation that is selected in the Events area.



Test suite: Configurations tab

In the Configurations tab of the test suite editor, you can edit the default test configuration. Using this Configuration tab, you can add modules to your test configuration, or add emulators and monitors to your test configuration modules, to more precisely control your tests.

The Configurations area provides a tree view of your default test configuration and displays the following information:

- The name of the test configuration
- The names of the modules in the test configuration
- The names of any emulators in each test configuration module
- The names of any monitors in each test configuration module

The control area provides the following icons to help you manage your default test configuration:

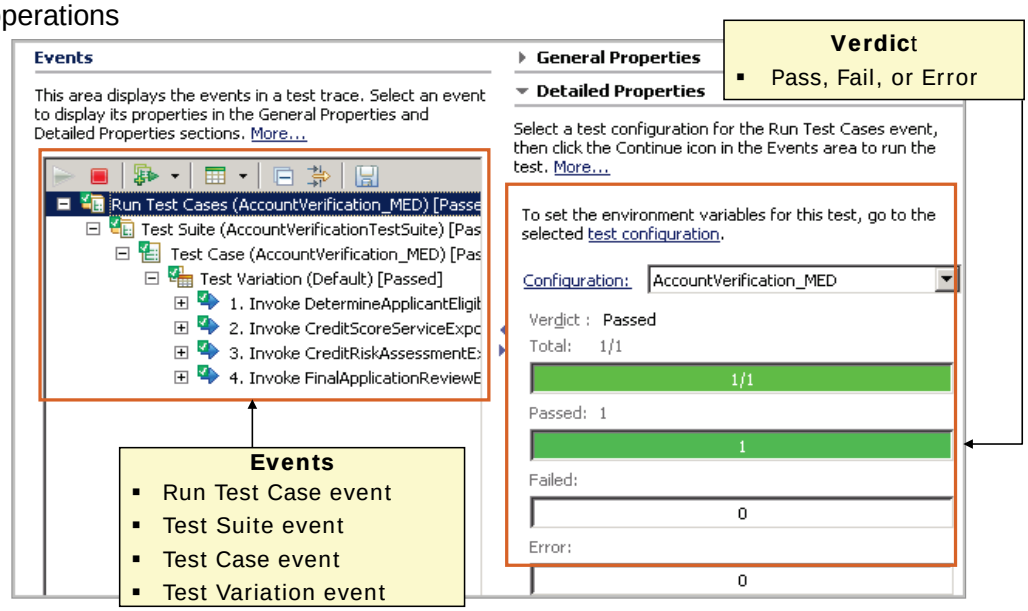
- Add: Opens the New Configuration wizard to add one of the following test configuration elements to the Configurations page:
  - Module
  - Emulator
  - Monitor
- Remove: Removes selected test configuration elements from the Configurations page

The Detailed Properties area displays the specific properties of any test configuration element that is selected in the Configurations area.

IBM Training

Running a test case

- Component test projects are created in the test suite editor
- Associated wizards are used to create and define test cases that comprise multiple operations



Advanced testing © Copyright IBM Corporation 2018

Running a test case

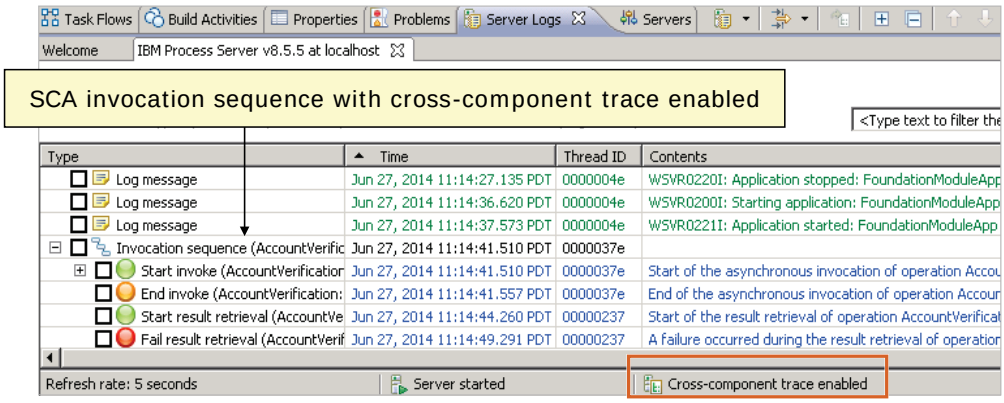
In addition to the standard events that are generated in the integration test client when you are unit testing, such as Invoke and Return events, component testing adds the following events in the integration test client:

- **Run Test** event: An informational event that is generated when you select the Run Test menu item from either the integration test client or the Business Integration view. The event informs you whether the component test passed or failed. It also presents statistics on the total number of test cases that were run and the number that passed, failed, or were flagged with an error.
- **Test Suite** event: An informational event that informs you whether the test suite passed or failed the test run. It also presents statistics on the total number of test cases that were run and the number that passed, failed, or were flagged with an error.
- **Test Case** event: An informational event that informs you whether the test case passed or failed the test run.
- **Test Variation** event: An informational event that informs you whether the test variation passed or failed the test run.

IBM Training

Cross-component trace (1 of 2)

- Cross-component trace provides server logs and SCA trace in one view
- Log information is correlated to the SCA flow
  - Across multiple synchronous or asynchronous instances and processes
  - Multiple log files and multiple threads
- Cross-component trace shows the execution path of all SCA components even if the source code is not in the current workspace
  - Invocations, the error or exception and which SCA component they are related to, and a snapshot of the input/output data that is passed and returned



Advanced testing

Cross-component trace

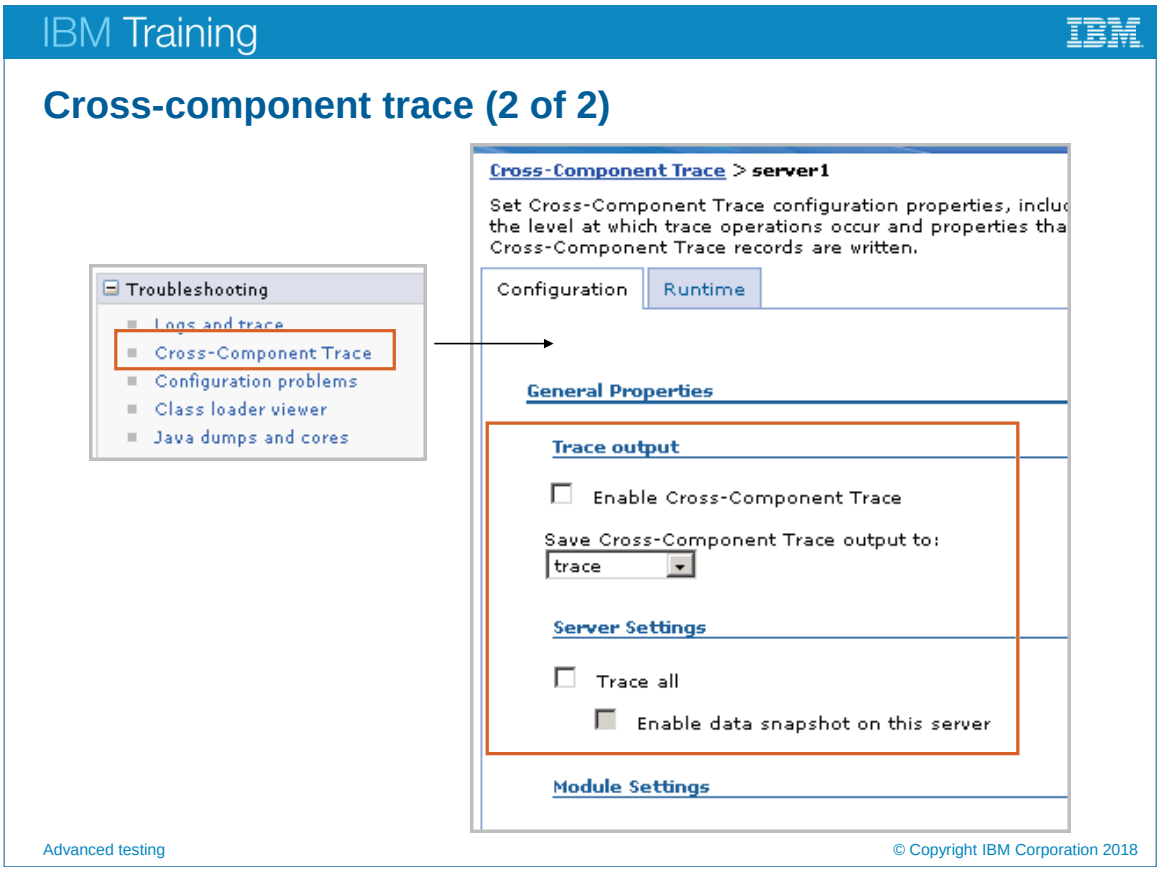
By default, the Server Log view displays standard server console and log records. However, if you enable cross-component tracing, the Server Log view also displays invocation records that can contain the invocation data that is passed between the components in your application. The invocation records are displayed in hierarchical format in the Server Log view. You can more easily understand the relationships that exist between the records. When you enable cross-component tracing, the Server Log view becomes an even more powerful tool for problem determination.

When you enable cross-component tracing on a server, invocation records are generated during SCA processing of modules and components. The invocation records include information about any errors or events that occurred during processing, such as runtime exceptions. If you choose to enable cross-component tracing with the data snapshot feature, the generated invocation records also contain the invocation input and output data that is passed between the components during processing.

You can enable or disable cross-component tracing for a server from either the Server Log view or the server administrative console. If you enable cross-component tracing from the Server Log view, the tracing is enabled only during the server session. When you next stop or restart the server, the cross-component trace state is automatically disabled by default. By comparison, if you enable cross-component tracing for a server from the server administration console, the cross-component tracing remains enabled for all sessions of the server until you choose to disable it again.

When you enable or disable cross-component tracing, you can choose from one of the following options:

- **Disabled**: This option disables cross-component tracing. No invocation records are generated in the server console and logs.
- **Enabled**: This option enables cross-component tracing. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, but the record properties do not include any invocation input and output data. The `SystemOut.log` and `trace.log` files are in the Server Log directory.
- **Enabled with data snapshot**: This option enables cross-component tracing with the data snapshot feature. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, and the record properties include invocation input and output data. This data is captured in input and output files under the `logs\xfa` folder.



Cross-component tracing allows identification of the `trace.log` data that is associated with IBM Business Process Manager modules and components. You can enable cross-component tracing on a server or on specific modules. The input and output data that is passed between Business Process Manager components can also be captured and can be used for problem determination by using IBM Integration Designer.

Configuration parameters are used for cross-component trace display on the Configuration tab.

When the server is restarted, the system applies the parameters that you set on the Configuration tab.

The console panel contains the following sections:

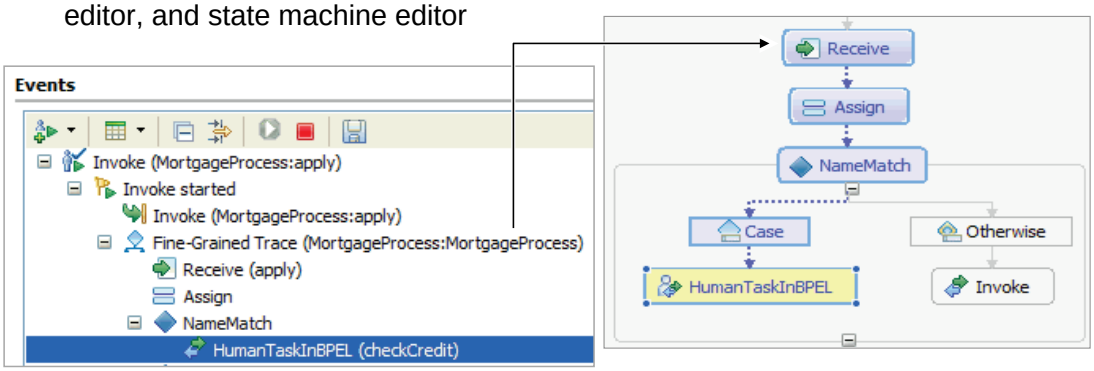
- **Trace output**
  - **Enable Cross-Component Trace**: Selecting **Enable Cross-Component Trace** prepares the server for the following options:
    - Cross-component trace for inbound application-specific call chains
    - Enabling cross-component trace on any module that is selected under Enable tracing for the selected Service Component Architecture (SCA) modules

    - **Save Cross-Component Trace output to**: Choose which file holds the data that the cross-component trace operations gather.
- **Server settings**
    - **Trace all**: Select this option to turn on cross-component trace for the creation of call chain information for all SCA modules in the server. Even with Trace all selected, you can add more SCA modules to the table of modules under Enable tracing for the selected Service Component Architecture (SCA) modules.
    - **Enable data snapshot on this server**: Select this option to enable the data snapshot feature of cross-component trace. When data snapshot is enabled, the system captures data that is sent in and passed between SCA components. This extra data (about what is passed between SCA components) can be large and is kept in separate files and not in the `trace.log` or `systemout.log` file.
- Module settings
    - Enable tracing for the selected Service Component Architecture (SCA) modules. This table provides a list of modules for which cross-component trace is enabled.

IBM Training

Fine-grained trace

- Fine-grained trace combines the test client and debugger to overlay the editors with a visualization to track steps in the path of events
  - Quickly visualize the path that is taken for a single execution of a component
  - Split screen feature while testing highlights activities as you pass through
- Fine-grained trace can be used in BPEL, state machines, and mediations and is enabled by default
  - Appropriate editors are auto-opened by clicking the event
  - The path of events is highlighted in the business process editor, mediation flow editor, and state machine editor



Advanced testing

Fine-grained trace

When you are testing business processes, state machines, or mediation flows in the integration test client, you can choose to run your tests either with or without fine-grained trace. If you run your tests with fine-grained trace, the events area of the test client is populated with more events. These events correspond to the elements encountered in the execution path of the component that is being tested. If the associated component editor is open, the execution path is traced and highlighted in the component editor so you can easily see the specific path that was taken and tested.

Although fine-grained trace is enabled by default, you can set a preference to disable it, as described in the topic “Enabling or disabling fine-grained trace” in the product documentation. If you disable fine-grained trace and you later encounter an exception during your testing, you can enable fine-grained trace again to help you determine the source of the problem. When you test with fine-grained trace and an exception is encountered, the events area of the test client displays an event. This event corresponds to the last element encountered in the component execution path before the exception was thrown.

IBM Training

Command-line test invocation

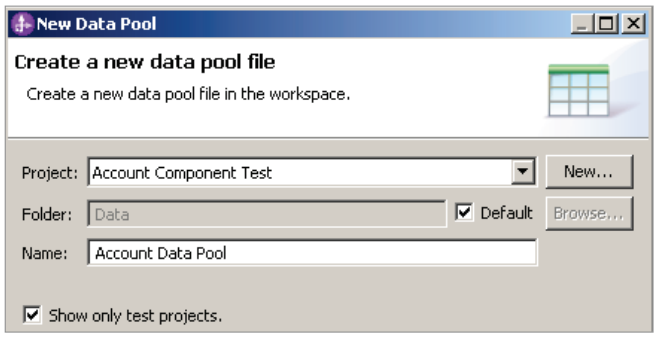
- Integration test client includes command-line invocation
- Automates tests by using Ant scripts
  - Schedule the tests during a low usage time
  - Result is an XML file that describes the results
  - Batch files (for Windows O/S) and shell scripts (for Linux O/S) are provided to run the Ant scripts
- Servlet that the test client calls to run test cases remotely in test project
  - Generated in a side Java EE project along with modules
  - URL for the servlet is in the form: `http://<host>:<port>/<testproject>Web/TestServlet`
- Ant has core support for CVS
  - Test projects and modules can be extracted from CVS, built, deployed, tested, and removed
  - User must know CVS commands

Command-line test invocation

IBM Training

Stand-alone data pool

- The integration test client has a stand-alone data pool wizard
- The data pool:
  - Is saved in your workspace
  - Can be exported with a project
  - Can be checked into source control
- By using the data pool editor:
  - You can open multiple data pools
  - You can copy values between pools

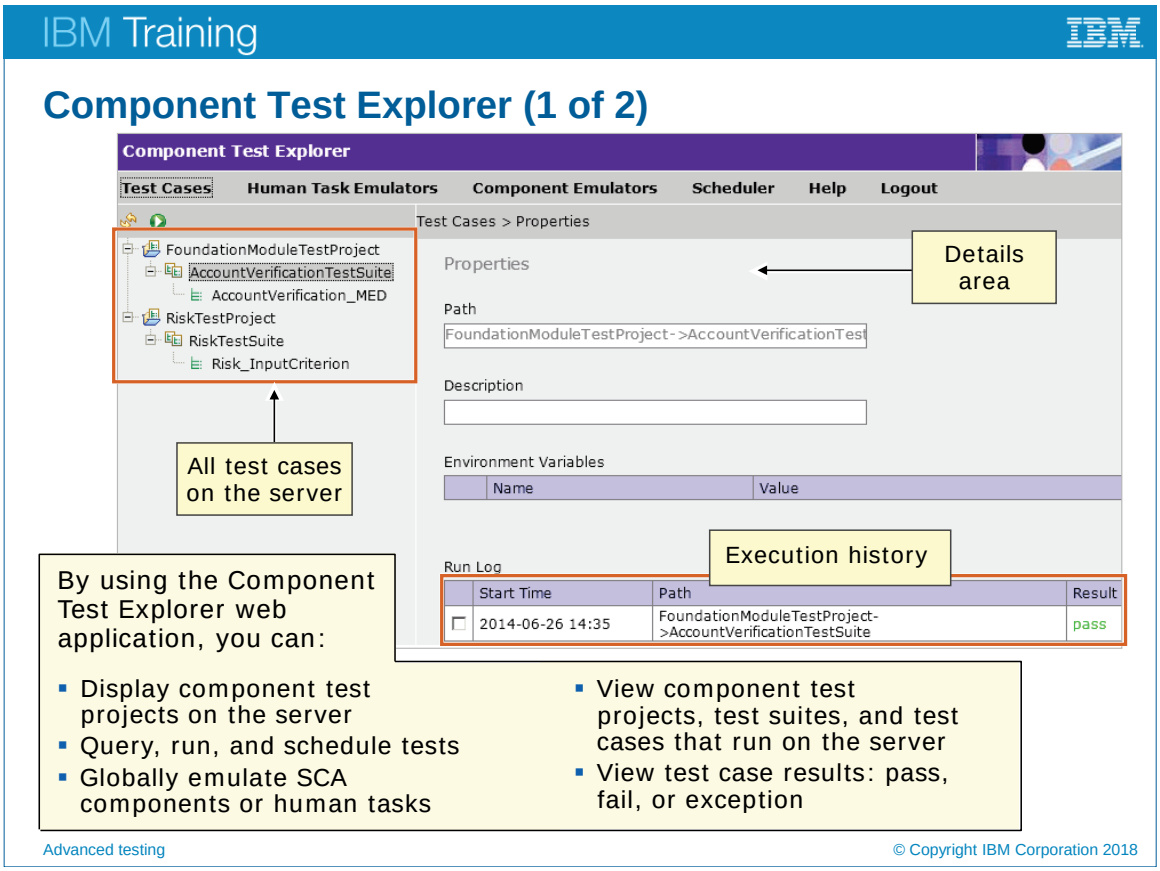


Advanced testing

Stand-alone data pool

On the events page of the integration test client, a value editor is provided. By using this editor, you can specify, view, edit, and pass values for operations, manual emulations, and event definitions. You can also save values to a data pool, where you can use the data pool editor to view and edit the values, and later reuse them in the value editor.

In the integration test client, you can use one or more data pools. You can save values to the data pools, edit values in the data pool editor, and reuse values from the data pools. You can more easily manage the input values for your operations and output values for your manual emulations. When you create a data pool, it is added to a test project.



Component Test Explorer

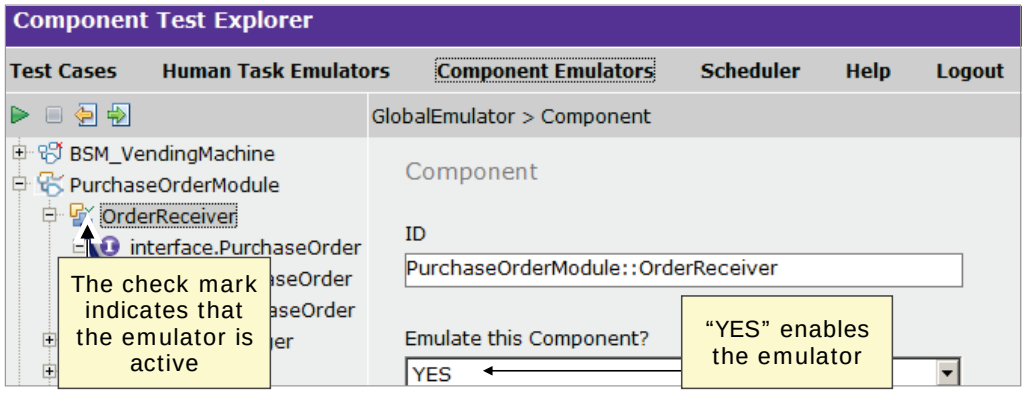
With the Component Test Explorer, you can manage and run test cases that are deployed to a test server or a stand-alone server. Like the Business Process Choreographer Explorer, the Component Test Explorer is a web client that you can invoke and run from inside or outside IBM Integration Designer. Using the Component Test Explorer, you can:

- Query and display component test projects, test suites, and test cases that are deployed on the server
- Select and run one or more component test projects, test suites, or test cases and then display and automatically save the results
- Schedule specific times to automatically run one or more component test projects, test suites, or test cases
- Define global server-wide emulators for components and import interfaces that are deployed on the server
- Specify global server-wide rules to automatically process and claim human tasks that are deployed on the server

IBM Training

Component Test Explorer (2 of 2)

- Component Test Explorer emulators:
  - Use if the component is not available at test time or if the execution can produce errors
  - Define a global emulator for any component in any module on server
  - Groovy assistance to emulate interface information for request and response parameters
  - Define and manage global emulators for specified human tasks



Advanced testing

Using the component emulators page, you can define a global emulator for any component in any module on the system. You can globally emulate a component that is not yet available but is needed for testing. You can also globally emulate a component that is available for testing but is not yet implemented.

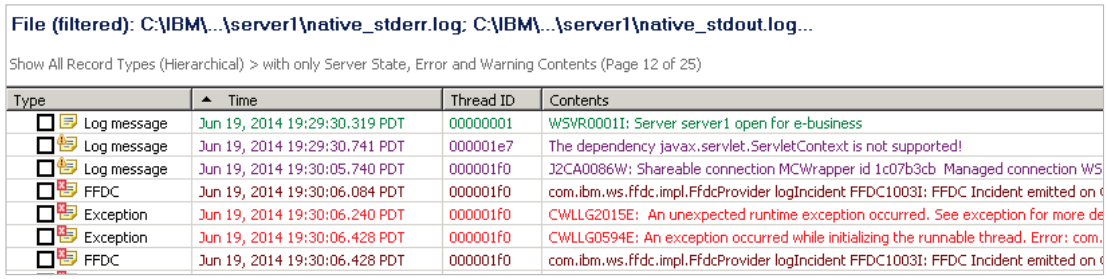
If you want to define an emulator for an operation, expand the component and interface, then select the operation name, and click **Create**. The details area displays the operation name and a table that contains the emulator rules as you create them. Each rule has a condition that describes the conditions under which the rule must be applied and a response that describes the output from the operation.

The condition section includes the Groovy statements that are constructed when you select the options in the request and operations sections. You can also enter a Groovy statement directly. An evaluated Groovy statement determines whether the response must be returned for the operation invocation. The condition section can contain many Groovy statements, but the last statement must evaluate to a Boolean string to determine whether the rule is used. Only the final statement is evaluated as the condition.

IBM Training

Server Logs view

- Console view returns a sizable volume of messages
- Server Logs view is filterable view of server logs in one view
  - `SystemOut.log and SystemErr.log`
- Able to view SCA errors along with server errors
- Highlight events with different colors
- Displays only server logs
- Potential performance effect
  - Full logs are loaded in memory



Advanced testing

Server Logs view

The Server Logs view is used to display server console and log file records. Although the Server Logs view automatically displays console records for each server that is started, you can also manually load and display the server console and log file records for any server. When cross-component tracing is enabled, the Server Logs view also displays invocation records that can contain the invocation data that passes between components. The Server Logs view is the tool for working with server console and log records. It provides several advantages over the traditional console view, such as the ability to filter records, display invocation records in hierarchical format, and load invocation records directly into the integration test client.

In the Business Integration perspective, you can open the Server Logs view by clicking the Server Logs tab. (If the Server Logs tab is not visible, you can open it by clicking **Window > Show View > Server Logs**.)

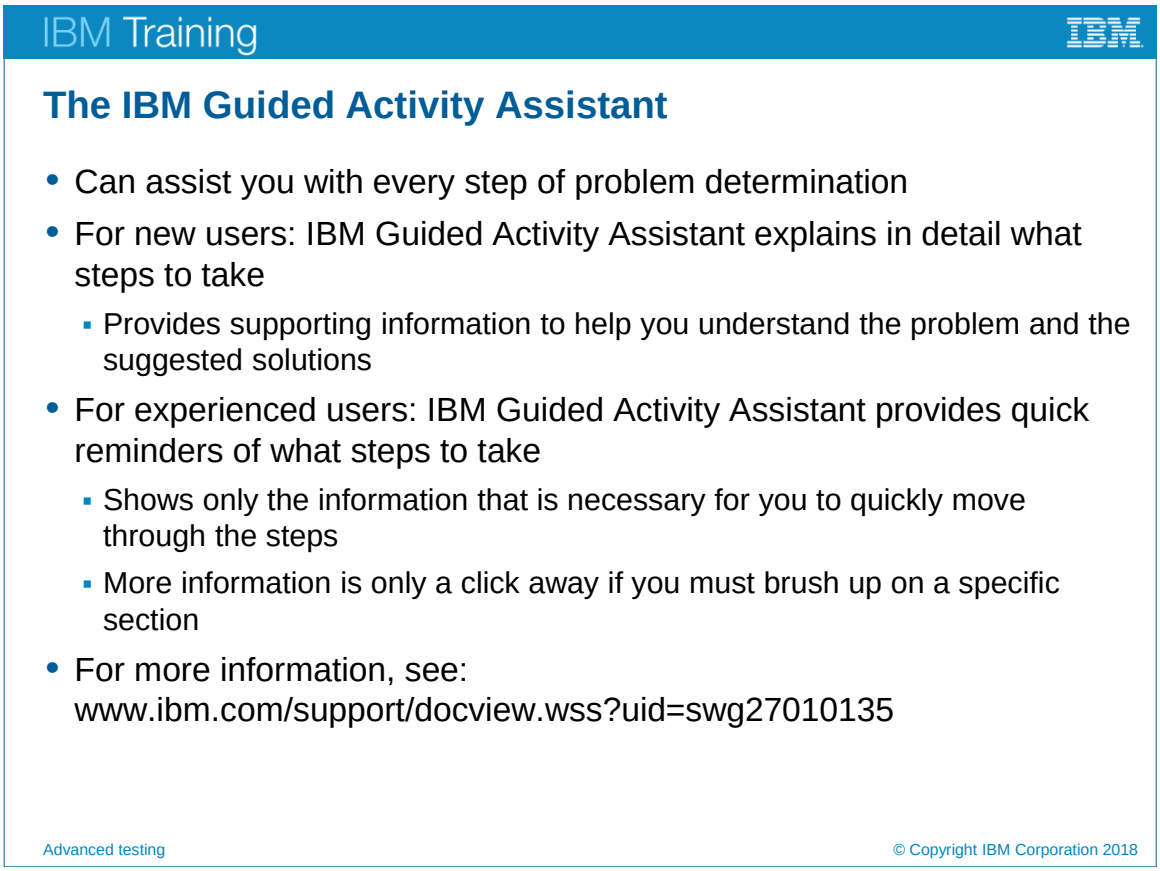
The Server Logs view has three main areas:

- **Welcome tab**: The **Welcome** tab is the home page of the Server Logs view. It provides an overview of the Server Logs view and introduces you to some of the key tasks that you can do in the view, such as:
  - Loading the contents of server consoles or server logs in to the Server Logs view
  - Filtering server console or server log records in the Server Logs view
  - Enabling or disabling cross-component tracing for servers
  - Loading invocation records into the integration test client
- **Server console and log tabs**: The **Server console** and **Log** tabs display the records of server consoles and server logs that you load into the Server Logs view. The following four columns in the Server console and Log tabs provide detailed information about the records:
  - **Type**: Displays the types of the records, such as log messages, FFDC records, exception records, and invocation records
  - **Time**: Displays the date and time that the records were generated
  - **Thread ID**: Displays the thread IDs of the records
  - **Contents**: Displays the first line of the contents for the records

  You can also right-click any record and click **Properties** to open a properties dialog box that displays the time, thread ID, and contents for the record. You can choose to view the contents in the default translated format (for easier reading and assimilation) or you can view the contents in the raw native format in which they were originally generated. Also, if you open the properties dialog box for an FFDC record or an invocation record, such as a “start,” “fail,” or “end” invocation record, the properties dialog box contains another data field. This field displays the invocation data that is passed between components.
- **Toolbar**: The toolbar in the Server Logs view comprises several icons that are used to initiate numerous tasks.

At the bottom of the Server Logs view, a status area returns messages that are related to the server console and log records.

You can click the **Select records to display** icon to open and select menu items that filter the records in the Server console or Log tabs. By default, any invocation records are displayed in hierarchical format and have check boxes that you can select for direct loading into the integration test client. Although you can choose to display records in a flattened format rather than a hierarchical format, the invocation records do not have check boxes. You cannot select them for direct loading into the integration test client.



The IBM Guided Activity Assistant

The IBM Guided Activity Assistant assists you with every aspect of debugging, from basic problem solving to gathering the diagnostic information needed to open a problem management report with IBM support.

Deploying integration applications with serviceDeploy

IBM Training

Overview of serviceDeploy

- Automate deployment (assemble, generate, and compile) of Service Component Architecture applications
- Helpful in situations where not all SCA component resources are in a single workspace
- Can be invoked by using either a command line or Ant tasks
- SCA resources (modules) and other application components (WAR, EJB JAR, RAR, and utility JAR) are passed to **`serviceDeploy`** as a single input archive
- Provides basic assembly for SCA components and generates Java EE application artifacts and packages into a single EAR ready for installation

```
serviceDeploy archive [<-workingDirectory temppath>
 <-outputApplication outputpathname.ear>
 <-nojeeDeploy true|false> <-freeform true|false >
 <-cleanStagingModules true|false> <-keep true|false>
 <-ignoreErrors true|false>
 <-classpath jarpathname;rarpathname;warpathname;...  -help]
```

Advanced testing

Overview of serviceDeploy

The `serviceDeploy` tool is designed to help automate deployment of SCA modules. With `serviceDeploy`, you can take multiple components that multiple developers build, and bring them together in an application that can be installed on IBM Process Server. The `serviceDeploy` tool is available as a separate executable script that can be called from the command line. An Ant task is also available, which calls `serviceDeploy` for larger build and deploy processes.

You can specify various parameters for the `serviceDeploy` utility. The main parameters are the input archive (including the name of the module that contains the components that you want to generate into an installable `.ear` file) and the output name of the `.ear` file. Other options for troubleshooting or providing more resources can be used at build and deploy time.

Parameters: With the `serviceDeploy` tool, you can specify the following parameters:

- `-inputarchive`

  A required positional parameter that specifies the `.jar`, `.zip`, or `.ear` file that contains the application to deploy. If the command is not run from the path in which the file exists, it must contain the full path for the file. The `.zip` file can be either a nested archive or a project interchange file.

- `-classpath`

  An optional parameter that specifies the locations of required resource files (`.jar` and `.rar`). The path to each file must be a fully qualified path that is separated with semicolons (;) with no spaces.

- `-cleanStagingModules`

  An optional parameter that specifies whether to delete staging modules within an input `.ear` file before deployment. By default, the `serviceDeploy` command imports the existing staging modules and their contents.

- `-freeform`

  An optional parameter that specifies that the Java EE subdirectory in the `service.jar` file must be treated as a free-form project.

- `-help`

  An optional parameter that is used to display the parameters for this command.

- `-ignoreErrors`

  An optional parameter that specifies that the `serviceDeploy` command builds a `.ear` file regardless of errors while building or validating the application. By default, the `serviceDeploy` command does not generate a `.ear` file if an application has errors.

- `-keep`

  An optional parameter that specifies whether to save any temporary files that are generated after deployment. By default, the `serviceDeploy` command deletes the temporary workspace.

- `-novalidate`

  An optional parameter that specifies whether to disable validation of all artifacts when using `serviceDeploy`. By default, the `serviceDeploy` validates all artifacts for the application.

- `-outputApplication`

  An optional parameter that specifies the name of the `.ear` file that the `serviceDeploy` command creates. The default is `inputarchiveApp.ear`, where `inputarchive` is the file name minus the extension that is specified for the input `.jar` file.

- `-uniqueCellID`

  An optional parameter that specifies a string identifier that is used to create a unique instance of the application. The identifier must be unique in the context of the cell.

- `-workingDirectory`

  An optional parameter that specifies a directory that the `serviceDeploy` command uses to write temporary files.

Input: The following file types can be used as input to the `serviceDeploy` command:

- `.jar`

  This file type is the most useful type for simple applications. The resulting `.ear` file contains a single `.jar` file and the required staging modules. The `.jar` file must contain the `service.module` file.

- `.zip` (project interchange)

  You can use IBM Integration Designer to export an archive file in project interchange format. This format is unique to Eclipse development. The exported `.zip` file must contain exactly one project with the `service.module` file. The resulting `.ear` file contains a number of modules, depending upon exactly what is in the project interchange.

- `.zip`

  You can create a `.zip` file that contains `.jar` files, `.war` files, and `.rar` files. Exactly one `.jar` file must contain the `service.module` file. All contained archives are included in the final exported `.ear` file.

- `.ear`

  You can always run the `serviceDeploy` command against a `.ear` file while exactly one `.jar` file in the `.ear` file contains a `service.module` file.

Output: When `serviceDeploy` completes processing, it creates a `.ear` file in the directory from which the command is run unless the `-outputApplication` parameter is specified.

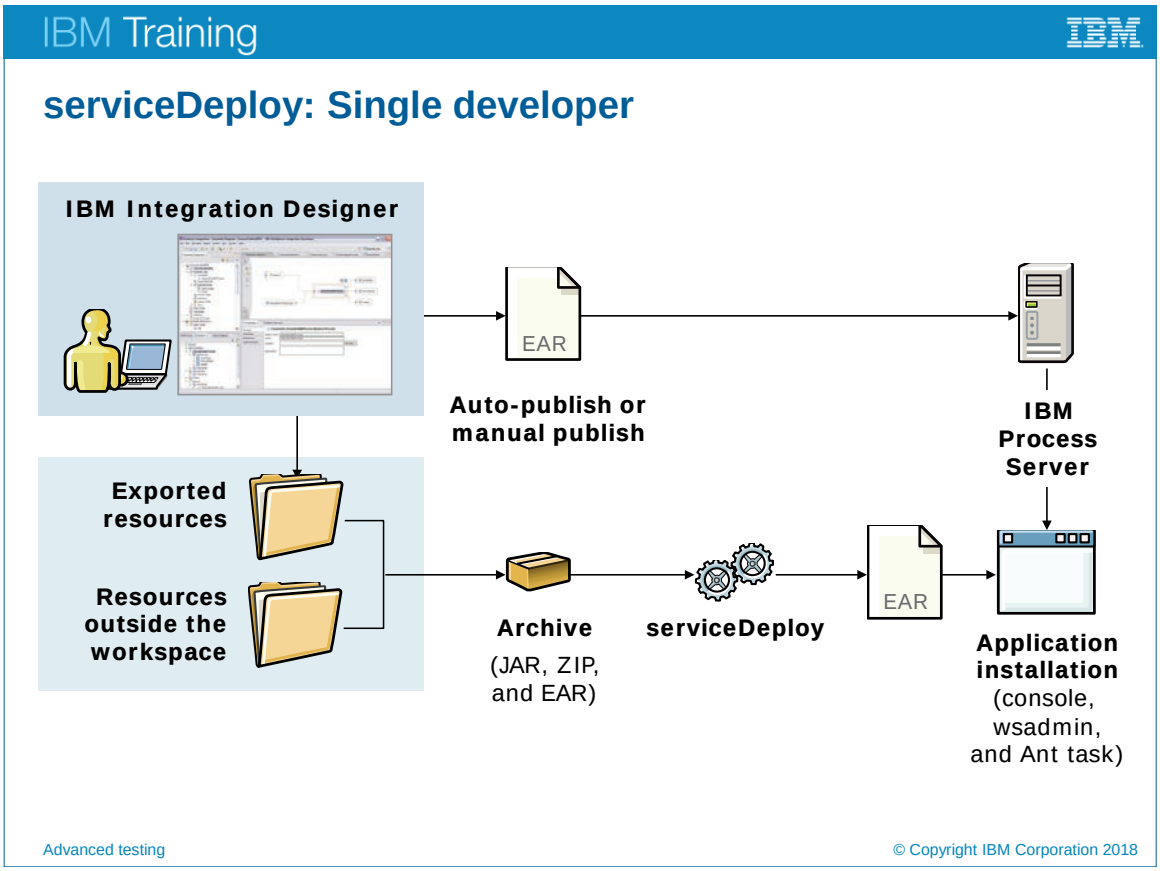
- Example of the serviceDeploy command:

  Consider the following command example:

```
servicedeploy MyValueModule.jar -classpath
"c:\java\myvaluemoduleres.rar;c:\java\commonres.jar" -freeform true
-keep
```

This command:

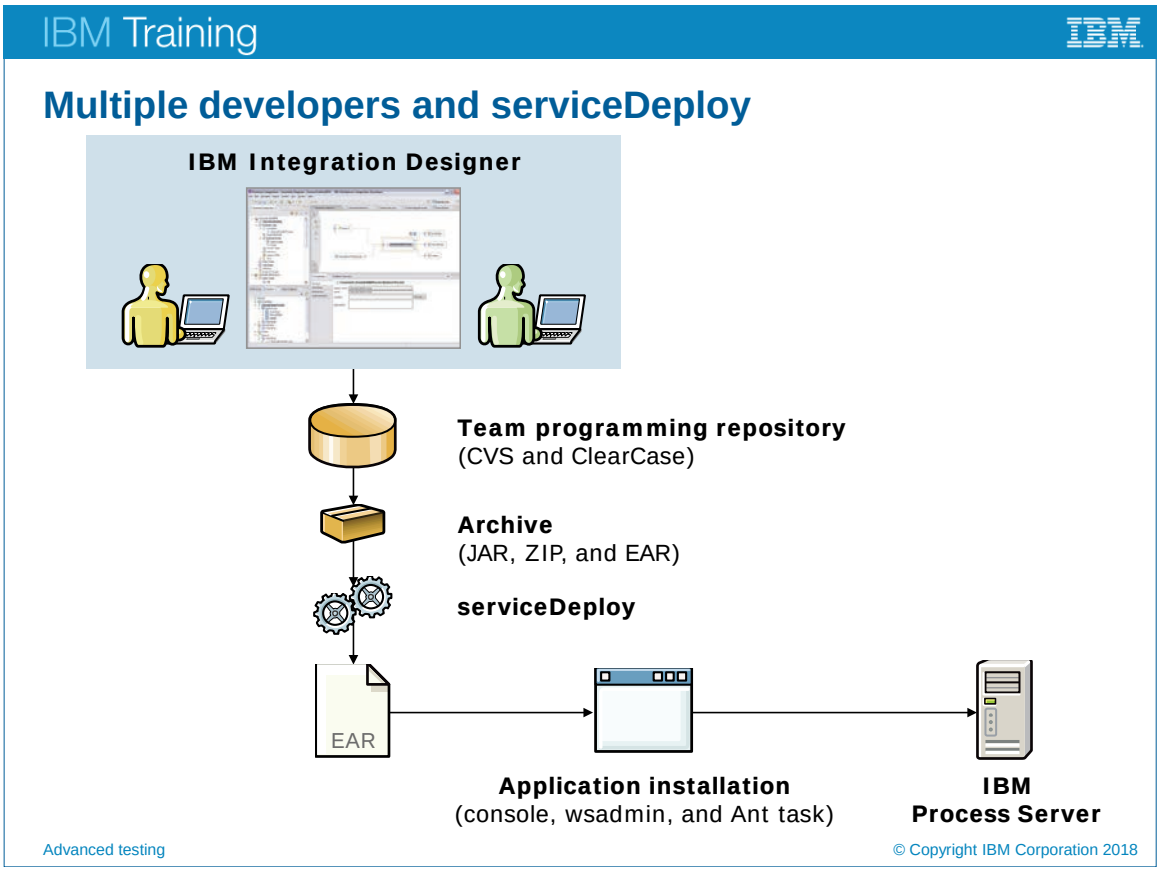
- Creates an application file that is called `MyValueModule.ear` from the `MyValueModule.jar` file
- Specifies that the resources exist in the directories `c:\java\myvaluemoduleres.rar` and `c:\java\commonres.jar`
- Enables the Java subdirectory within the `.jar` file as free-form
- Keeps the temporary files that are generated during deployment



serviceDeploy: Single developer

This page demonstrates a single developer who is using a `serviceDeploy` to do all the tasks necessary for deployment of an application. When all the components are placed on the assembly editor that is wired, and saved, all the necessary code is generated into an EAR file. This file can be deployed manually or by using the “add and remove projects” option to publish modules to the server. A single user might want to use `serviceDeploy` for situations where resources are available outside the workspace that the user wants to add into the EAR file.

You can also use `serviceDeploy` to bring multiple components together in a single EAR file; `serviceDeploy` can include other resources and dependencies (such as a web module). This capability includes artifacts that you do not have in your workspace but are available in another location, for instance on a network server. You first create an archive, you then pass it to a `serviceDeploy` script, and the output is an EAR file. At that point, you can use some other installation mechanism such as the administrative console, `wsadmin`, or an Ant task. `serviceDeploy` is a script to assist with deployment. `serviceDeploy` does not install the application.



Multiple developers and serviceDeploy

This page demonstrates the use of `serviceDeploy` in an environment with multiple developers who are working on different components. The work can be brought together, for instance in a nightly batch job, and the application can be assembled and installed on the server for testing the next morning.

Developers can work in a team programming environment, storing resources in a repository such as Rational ClearCase or Concurrent Versions System (CVS). The appropriate APIs or commands to retrieve those artifacts out of the repository are used. The artifacts are placed in an archive, and the archive is passed to the `serviceDeploy` tool. `serviceDeploy` compiles the resources into an EAR file, and the application is installed by using the administrative console, `wsadmin`, or an Ant task. When your SCA application is installed on IBM Process Server, you can manage the runtime environment through the administrative console.

IBM Training

Unit summary

- Describe the advanced testing facilities that are available in IBM Integration Designer, including the Component Test Explorer and cross-component trace
- Describe the integration debugger
- Define the purpose and function of the serviceDeploy tool
- Describe how to use serviceDeploy in single-developer and multiple-developer environments

Unit summary

IBM Training

Checkpoint questions

1. True or False: You can open multiple instances of the test client and use them to do simultaneous testing.

2. True or False: An EAR file that is ready for deployment to the IBM Process Server runtime environment is the output of the `serviceDeploy` tool.

3. True or False: Component projects include test suites, which are containers for test cases.

4. True or False: Cross-component trace provides a separate server logs view and a separate SCA trace view.

Checkpoint questions

IBM Training

Checkpoint answers

1. True.
2. True.
3. True.
4. False. Cross-component trace provides server logs and SCA trace in one single view.

Checkpoint answers

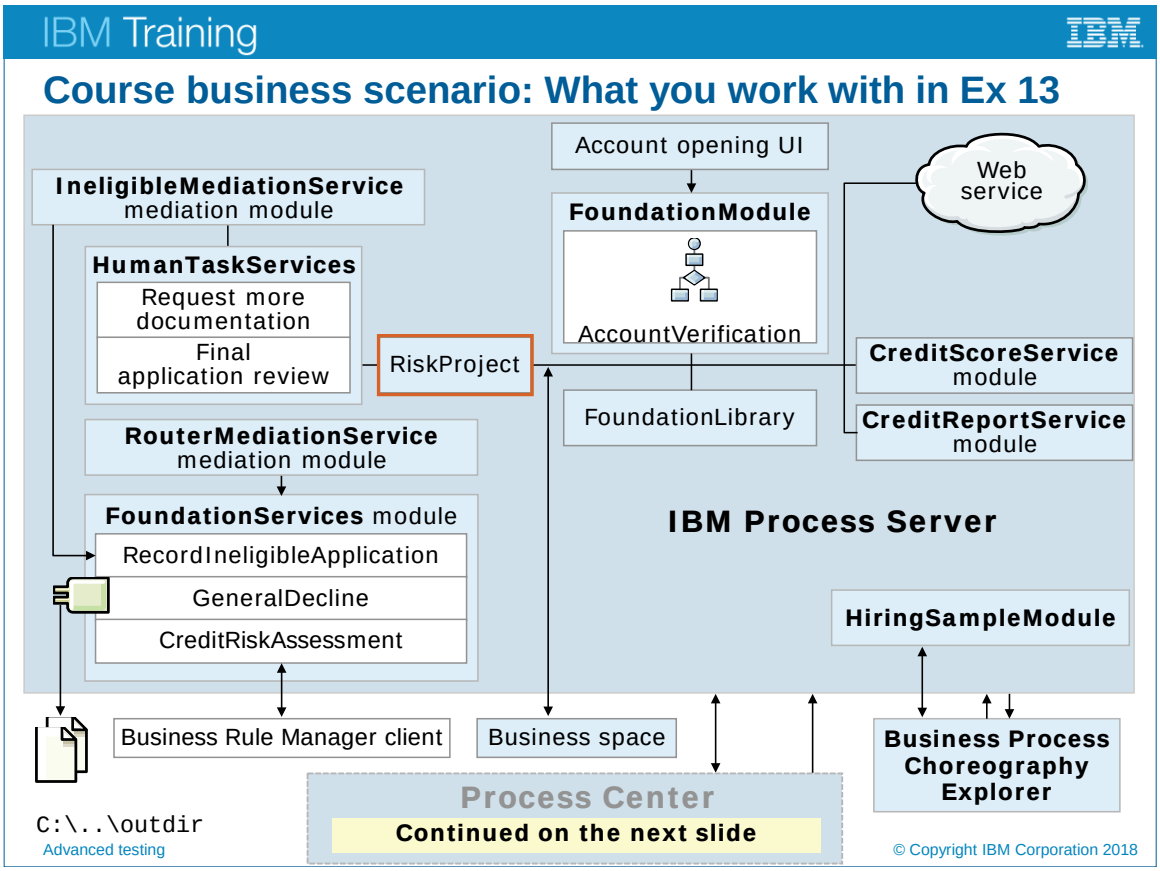
IBM Training

Exercise 13: Using component tests

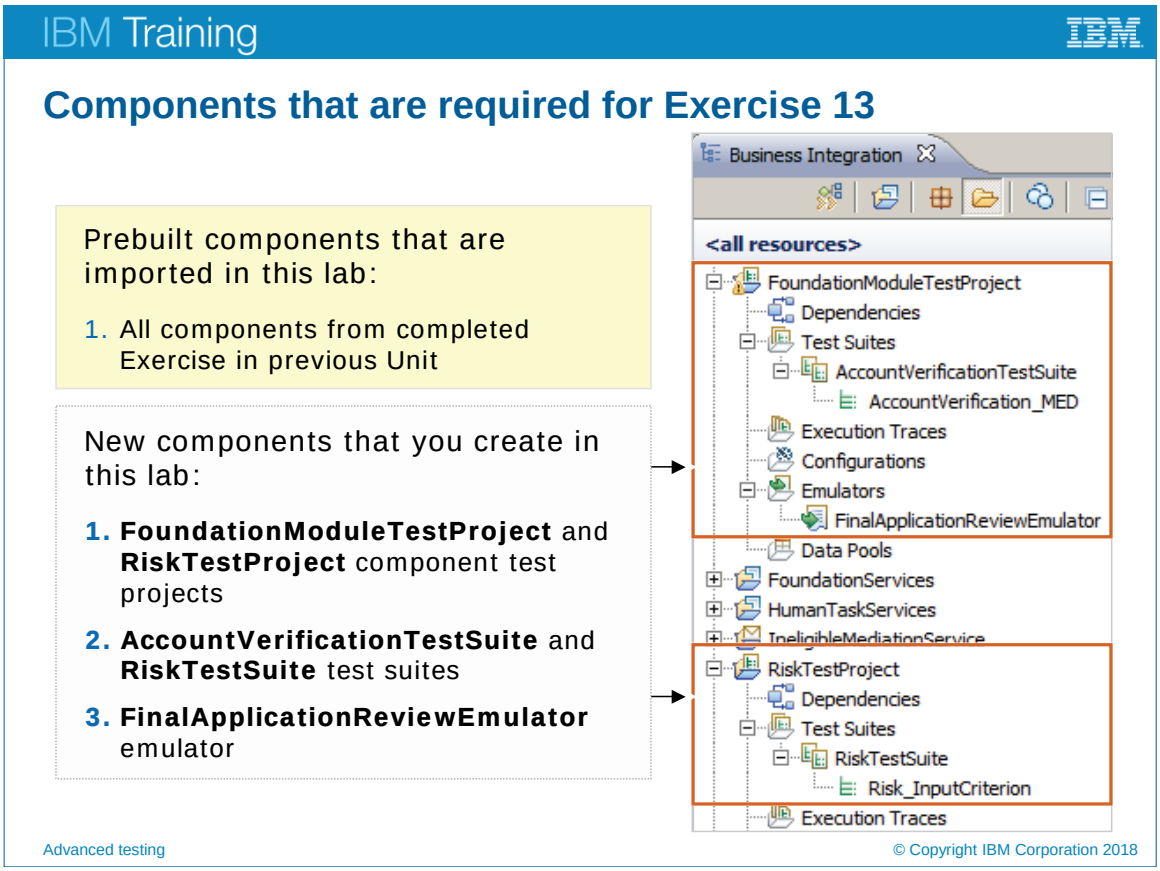
After completing this exercise, you should be able to:

- Create a component test project with an operation-level test case
- Create a component test project with a scenario-based test case
- Run component test project test suites in the IBM Integration Designer integrated test environment

Exercise 13: Using component tests



Course business scenario: What you work with in Ex 13



Components that are required for Exercise 13

If applicationDecision is set to false during FinalApplicationReview (the application is declined) and the customer's creditRisk is HIGH, the application is routed through the "generate decline" component. If applicationDecision is set to false during FinalApplicationReview and the customer's creditRisk is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the RouteRequest component. The RouteRequest flow component contains the mediation logic that routes the application to the appropriate decline service.

The RouteRequest mediation flow consists of both a request flow and a response flow. In the flow, the CustomerApplication is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the AccountVerification process.

Exercise 13: Using component tests

Purpose:
In this exercise, you use component tests to test applications. You also use cross-component tracing to inspect SCA components.

In IBM Integration Designer, the component test feature is the designated tool for setting up repeatable, predictable tests for both individual components and groups of components. You can use operation-level and scenario-based test cases. Component test projects contain test suites and are deployed to the IBM Process Server test environment as EAR files.

When deployed, the integrated test environment or the Component Test Explorer can be used to run the component tests for performance and load testing.

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Center, and IBM Integration Designer test environment.

Part 1. Create a component test project with an operation-level test case

In IBM Integration Designer, you can test your modules by creating unit tests and component tests. In unit testing (the method that you used in the previous exercises), you choose the components and interfaces that contain the operations that you want to test. Then, you test the operations one at a time in the integrated test client. In component testing, you use the test suite editor and associated wizards to create test cases that comprise one or more operations. In this way, you can automate and simultaneously test the operations. Component test projects are SCA modules and are built and deployed like other SCA modules.

With the ability to test individual modules, operations, or components, the developer can prove that each of the various parts of an end-to-end solution works correctly. Then, when the entire solution is tested, it is easier to identify where problems originate.

In this portion of the exercise, you create a component test project that contains a test suite, an operation-level test case, and variations of the test case. The test case that you create tests an individual operation: the InputCriterion operation of the CreditRiskAssessment interface. The CreditRiskAssessment service contains business rules that return predefined creditRisk values for each one of the four companyName test cases in your scenario. The creditRisk is `LOW` for IBM, `HIGH` for AbcCo and TestCo, and `MED` for ACME. The component test project uses IBM for the Default variation and AbcCo for the second variation. To save time, do not create variations for TestCo or ACME.

These tests require the right data inputs to match what the interfaces to the various modules expect. If you look again at the business objects and data maps, you can see the extensive data needed. In testing, it also includes expected values, allowing the tools to determine quickly and easily, whether the results are correct or not. To make testing easier and more automated, predefined data sets are made available here.

1. Create the component test project,

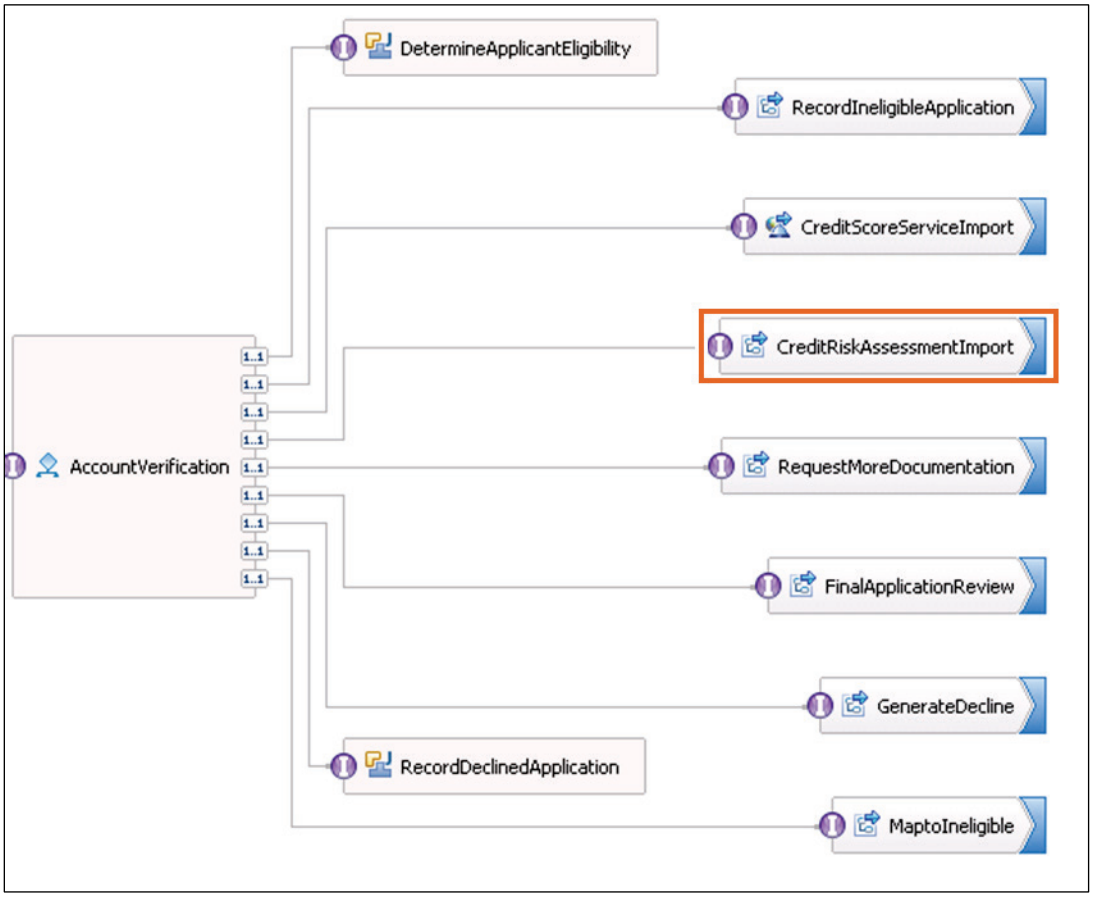
1. On your desktop, open the **Exercise Shortcuts** folder.
2. Double-click the **Exercise 13** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
3. If you get a message that the server is already set to publish, then click **OK**.
4. When the **Workspace Migration** window is displayed, clicked **Next**.
5. Click **Finish** to complete the migration.
6. Click **OK** in the **Migration Validation** window.
7. Wait for the workspace to build.
8. Close the **Getting Started** tab.

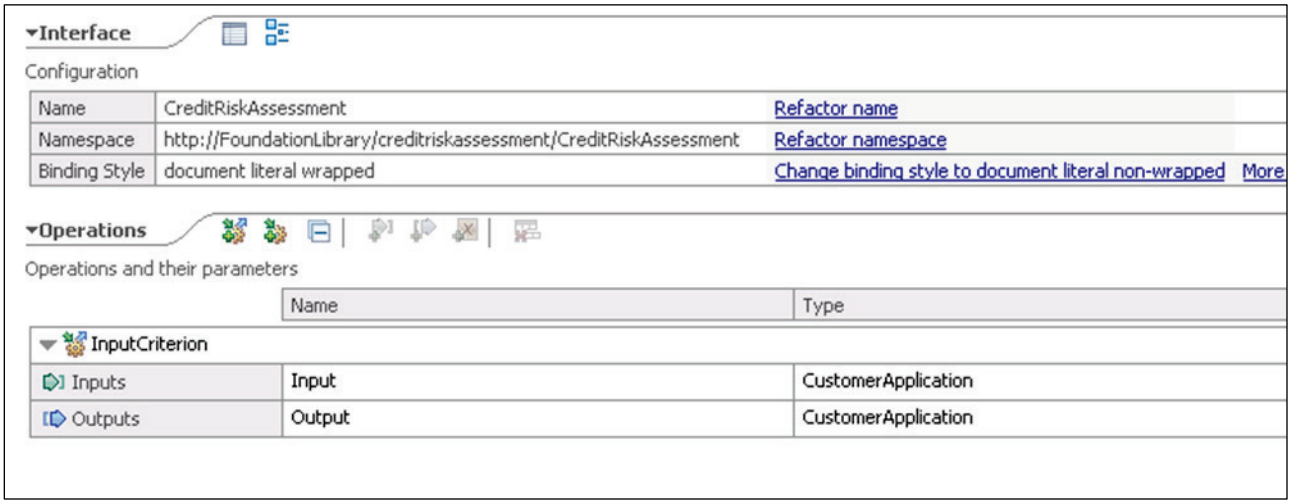
2. Create a test project that is named: `RiskTestProject`

1. Click File > New > Component Test Project from the menu options.
2. In the **Project name** field, type: `RiskTestProject`
3. Click **Finish**.

3. Create an operation-level test suite that is named: `RiskTestSuite`

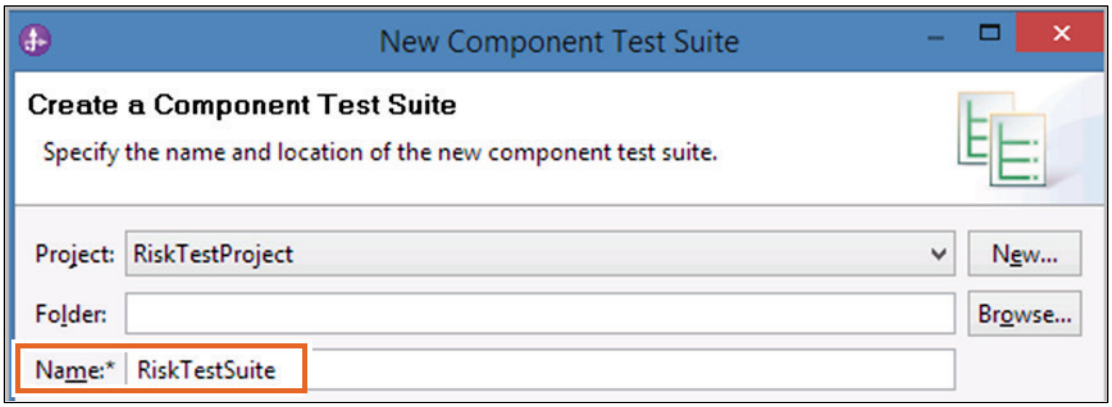
The test suite tests the InputCriterion operation of the CreditRiskAssessment interface.



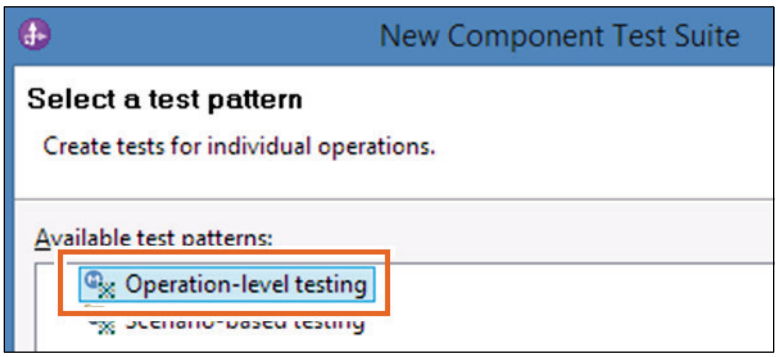


1. In the Business Integration view, expand **RiskTestProject**.
2. Right-click **Test Suites** and click **New > Component Test Suite** from the menu.

3. At the “Create a Component Test Suite” panel, take the following actions:
   - Leave the **Folder** field empty.
   - In the **Name** field, type: `RiskTestSuite`

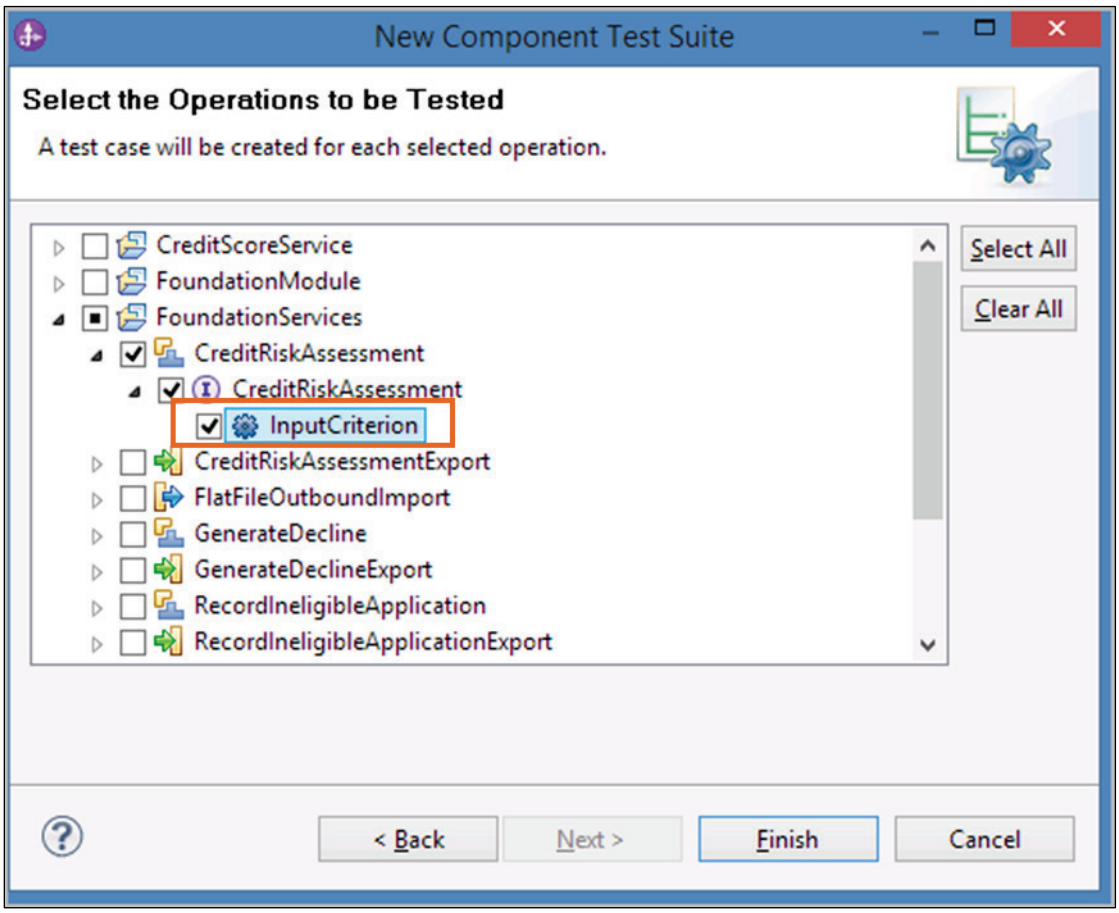


4. Click **Next**.
5. At the “Select a test pattern” panel, select **Operation-level testing**.



6. Click **Next**.

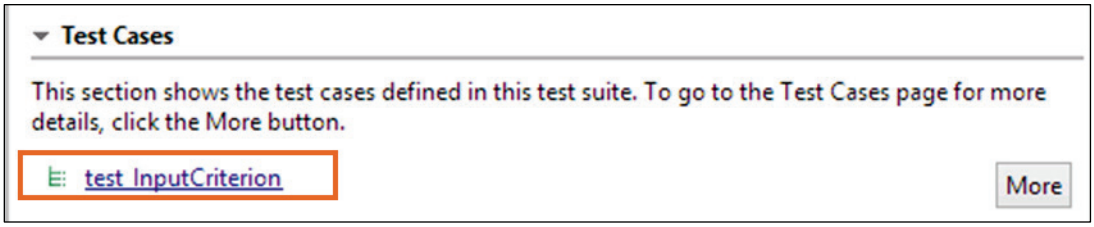
7. At the “Select the Operations to be Tested” panel, take the following actions:
   - Expand **FoundationServices > CreditRiskAssessment > CreditRiskAssessment**.
   - Select the **InputCriterion** operation in the **CreditRiskAssessment** interface.



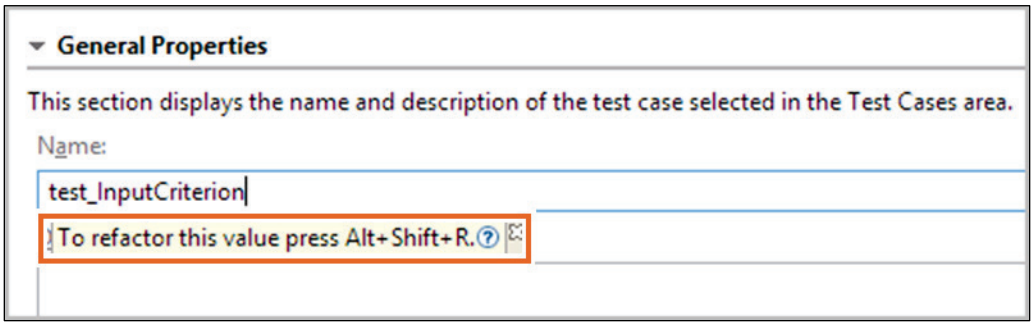
8. Click **Finish**.

4. Refactor the name of the test case from test_InputCriterion to: Risk_InputCriterion

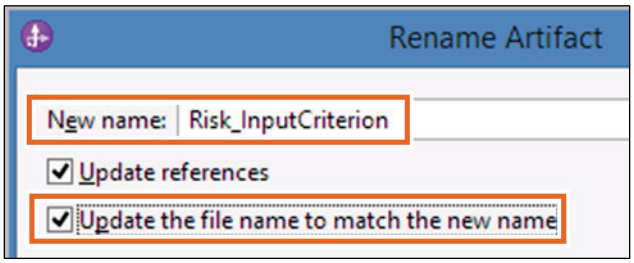
1. In the **Test Cases** section of the test suite editor, click the **test_InputCriterion** link.



2. In the **General Properties** section, place the cursor in the **Name** field and press Alt+Shift+R to refactor the name.



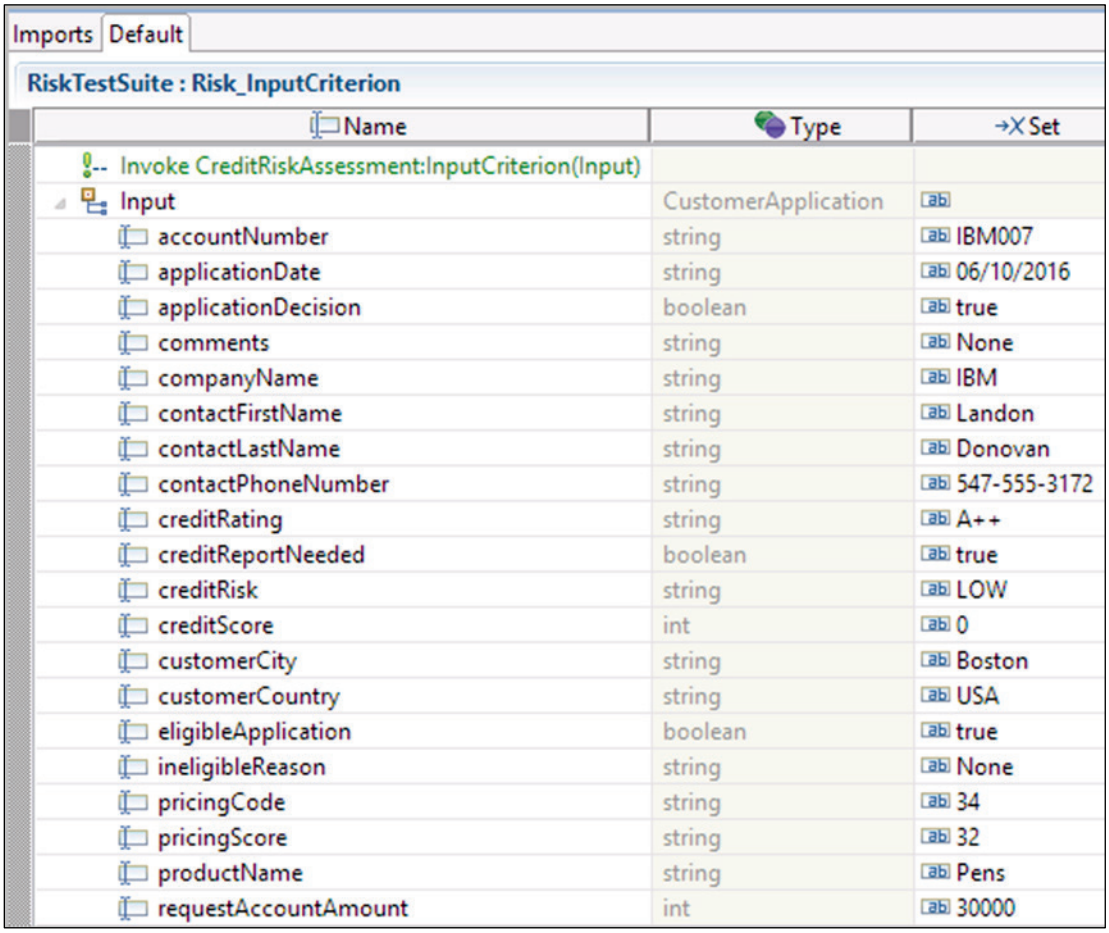
3. In the Rename Artifact dialog box, take the following actions:
   - In the **New name** field, type: `Risk_InputCriterion`
   - Select Update the file name to match the new name.



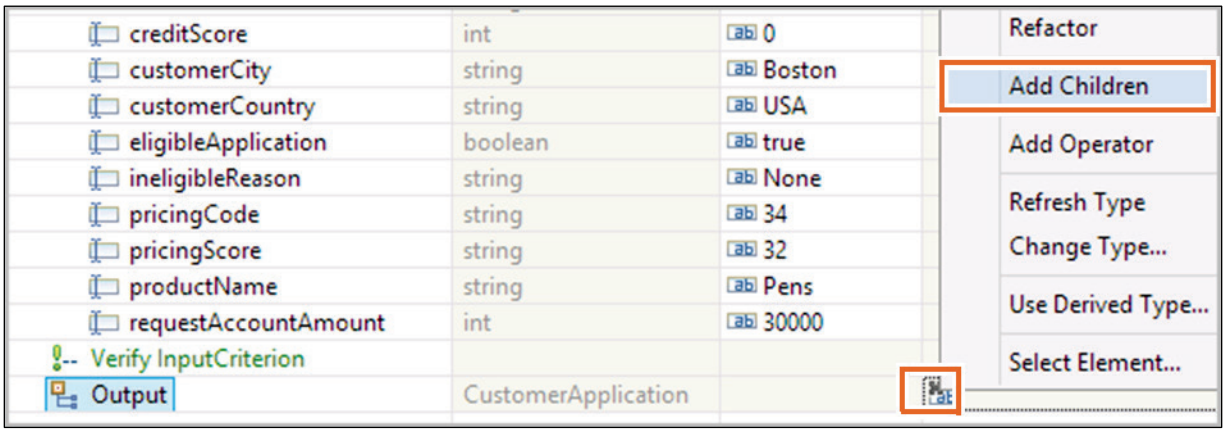
4. Click **Refactor**.

5. Use the values in `C:\labfiles\Support Files\Ex13\EX13_Default.xml` to enter the input data for the Default test variation.
   1. At the bottom of the window, switch to the **Default** tab in the **Test Data Table** view.
   2. Right-click **Input** and click **Import from File** from the menu.
   3. Browse to `C:\labfiles\Support Files\Ex13`.
   4. Select **EX13_Default.xml** and click **Open** to populate the XSet column with the required test data.

5. Alternatively, manually enter the following values on the **Default** tab (right-click the **Test Data Table** view and click **Maximize** to assist you with data entry).
   - accountNumber: `IBM007`
   - applicationDate: `06/10/2016`
   - applicationDecision: `true`
   - comments: `None`
   - companyName: `IBM`
   - contactFirstName: `Landon`
   - contactLastName: `Donovan`
   - contactPhoneNumber: `547-555-3172`
   - creditRating: `A++`
   - creditReportNeeded: `true`
   - creditRisk: `LOW`
   - creditScore: `0`
   - customerCity: `Boston`
   - customerCountry: `USA`
   - eligibleApplication: `true`
   - ineligibleReason: `None`
   - pricingCode: `34`
   - pricingScore: `32`
   - productName: `Pens`
   - requestAccountAmount: `30000`

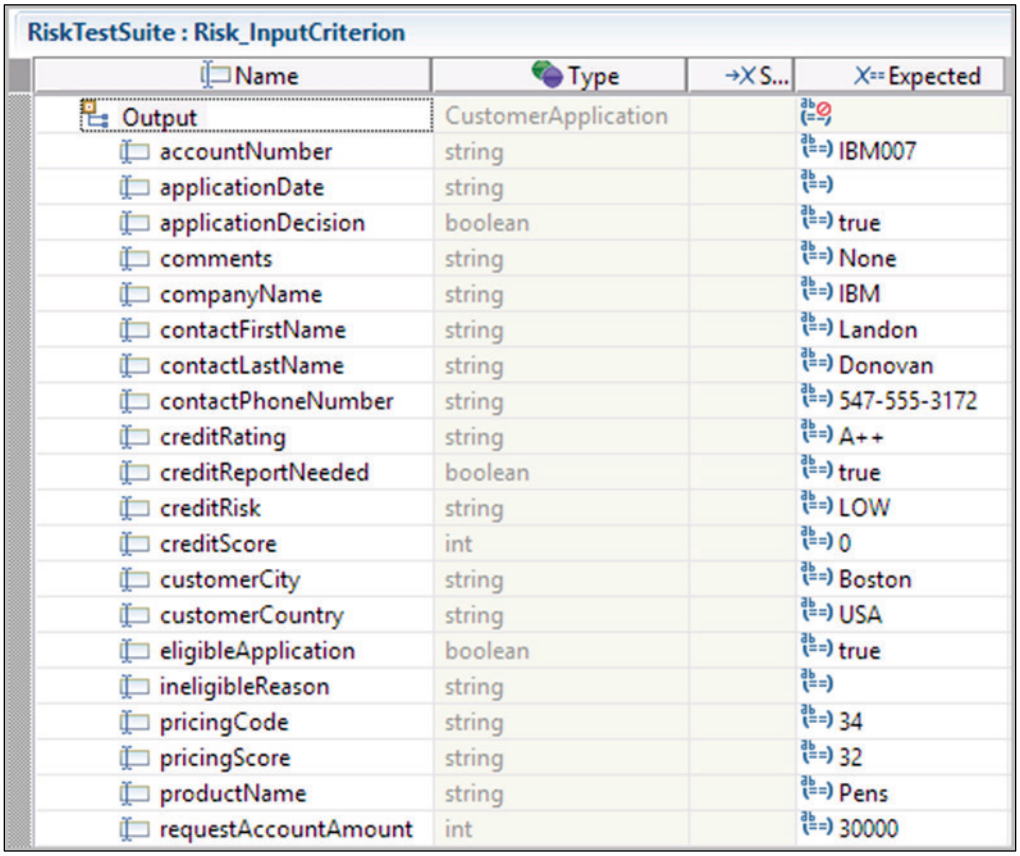


6. Use the values in `C:\labfiles\Support Files\Ex13\EX13_Default_Out.xml` to enter the output data that is expected from the CreditRiskAssessment service.
   1. In the **Test Data Table** view, on the **Default** tab, scroll to the **Output** section.
   2. Right-click the **Expected** column value and click **Add Children** from the menu.



   3. Right-click **Output** and click **Import from File**.
   4. Browse to `C:\labfiles\Support Files\Ex13`.

5. Select **EX13_Default_Out.xml** and click **Open** to populate the **Expected** column with the output test data.
6. Alternatively, manually enter the following values.
    - accountNumber: `IBM007`
    - applicationDate: [leave blank]
    - applicationDecision: `true`
    - comments: `None`
    - companyName: `IBM`
    - contactFirstName: `Landon`
    - contactLastName: `Donovan`
    - contactPhoneNumber: `547-555-3172`
    - creditRating: `A++`
    - creditReportNeeded: `true`
    - creditRisk: `LOW`
    - creditScore: `0`
    - customerCity: `Boston`
    - customerCountry: `USA`
    - eligibleApplication: `true`
    - ineligibleReason: [leave blank]
    - pricingCode: `34`
    - pricingScore: `32`
    - productName: `Pens`
    - requestAccountAmount: `30000`

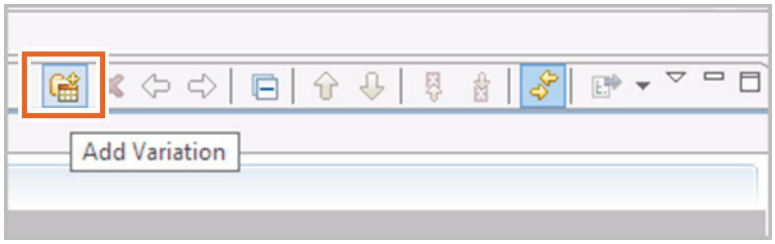


RiskTestSuite : Risk_InputCriterion

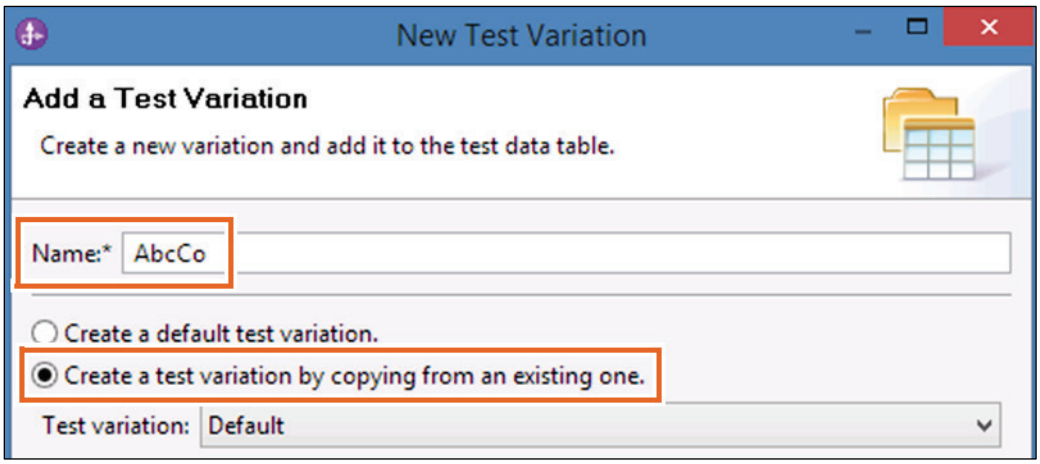
Name	Type	→X S...	X== Expected
Output	CustomerApplication		
accountNumber	string		IBM007
applicationDate	string		
applicationDecision	boolean		true
comments	string		None
companyName	string		IBM
contactFirstName	string		Landon
contactLastName	string		Donovan
contactPhoneNumber	string		547-555-3172
creditRating	string		A++
creditReportNeeded	boolean		true
creditRisk	string		LOW
creditScore	int		0
customerCity	string		Boston
customerCountry	string		USA
eligibleApplication	boolean		true
ineligibleReason	string		
pricingCode	string		34
pricingScore	string		32
productName	string		Pens
requestAccountAmount	int		30000

7. Whether you imported the test data or not, you must manually set the **applicationDate** and **ineligibleReason** values in **Output** by taking the following actions:
   - Right-click the empty **Expected** value for **applicationDate** and click **Set To > Not Equals Null** from the menu. It indicates that the result must contain some value, which can be anything.
   - Right-click the empty **Expected value** for **ineligibleReason** and click **Set To > Do Not Care** from the menu. It indicates that the result can be anything, including nothing.
8. Save your changes.

7. Use the values in `C:\labfiles\Support Files\Ex13\EX13_AbcCo.xml` and `EX13_AbcCo_Out.xml` to create another variation of the input and output data. Because the Default variation uses IBM, this test uses `AbcCo` for the companyName. IBM returns a creditRisk value of `LOW`. AbcCo returns a creditRisk value of `HIGH`.
   1. In the **Test Data Table** view, click the **Add Variation** icon (look far right to the **Test Data Table** view if your window is maximized).



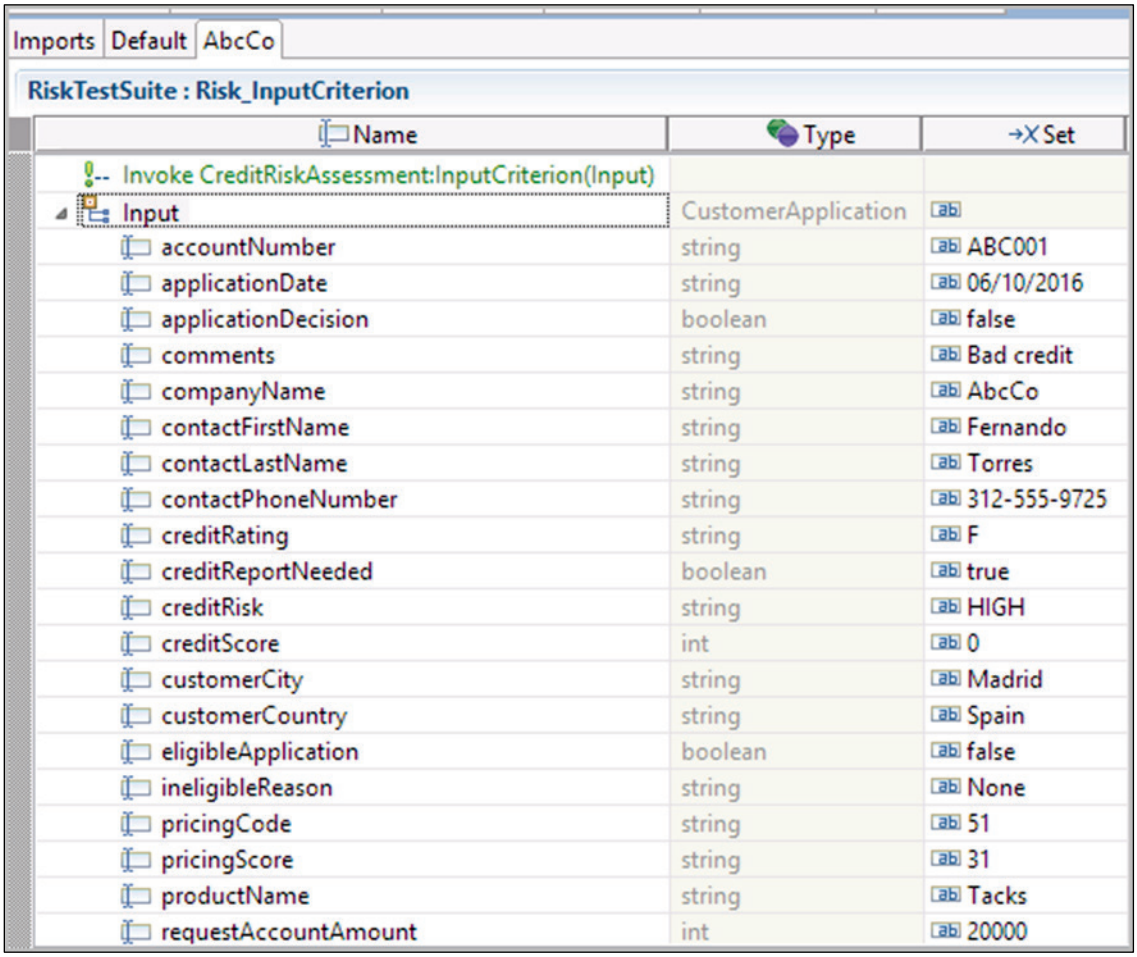
2. In the **Add a Test Variation** pane, take the following actions:
   - In the **Name** field, type: AbcCo
   - Select Create a test variation by copying from an existing one.
   - Verify that the **Test variation** field is set to **Default**.



3. Click **Finish** to create a set of data based on the Default test variation.
4. You now see a new tab in the Test Data Table view that is named AbcCo.

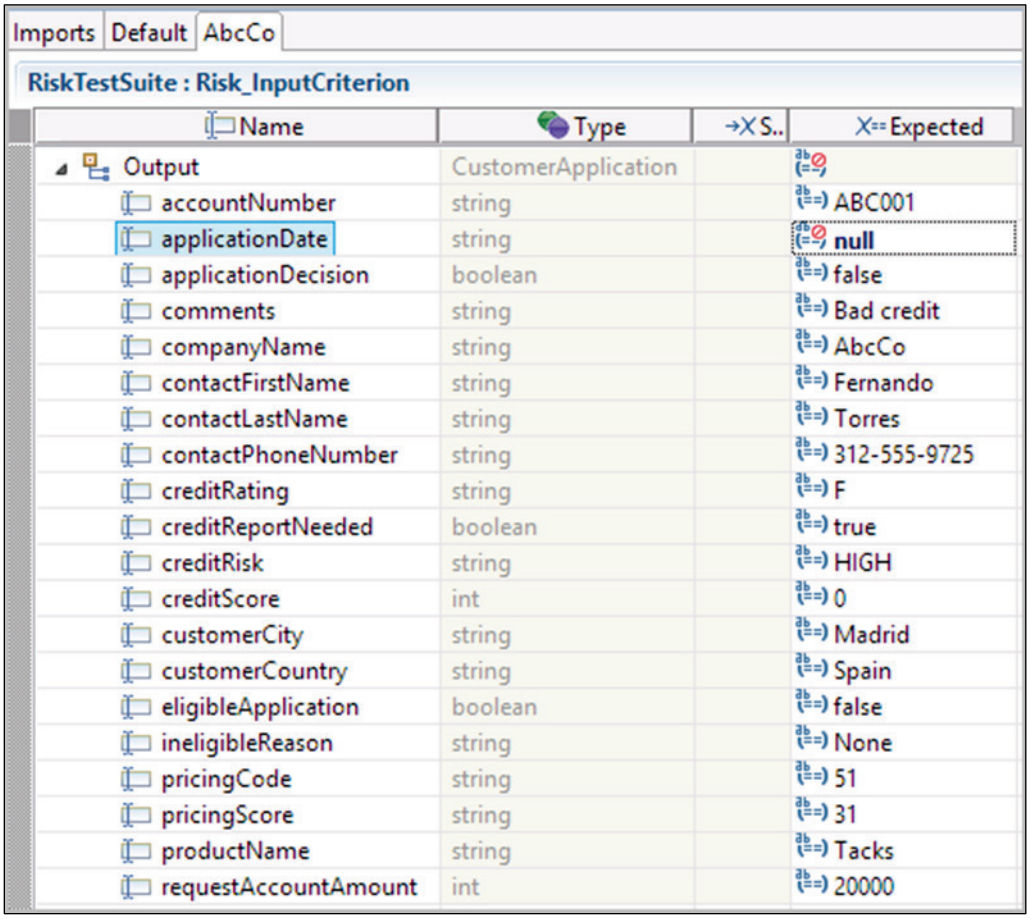
5. Right-click **Input** and click **Import from File**.
6. Browse to `C:\labfiles\Support Files\Ex13`.
7. Select **EX13_AbcCo.xml**, and click **Open** to populate the XSet column with the required test data.

8. If you did not import the test data, manually change the following values:
    - accountNumber: `ABC001`
    - applicationDate: `06/10/2016`
    - applicationDecision: `false`
    - comments: `Bad credit`
    - companyName: `AbcCo`
    - contactFirstName: `Fernando`
    - contactLastName: `Torres`
    - contactPhoneNumber: `312-555-9725`
    - creditRating: `F`
    - creditRisk: `HIGH`
    - customerCity: `Madrid`
    - customerCountry: `Spain`
    - eligibleApplication: `false`
    - pricingCode: `51`
    - pricingScore: `31`
    - productName: `Tacks`
    - requestAccountAmount: `20000`



9. Right-click **Output** and click **Import from File**.
10. Browse to `C:\labfiles\Support Files\Ex13`.
11. Select **EX13_AbcCo_Out.xml** and click **Open** to populate the **Expected** column with the required test data.
12. Right-click the **Expected** value for **applicationDate** and click **Set to > Not equals Null**.

13. If you did not import the test data, manually change the following values:
    - accountNumber: `ABC001`
    - applicationDecision: `false`
    - comments: `Bad credit`
    - companyName: `AbcCo`
    - contactFirstName: `Fernando`
    - contactLastName: `Torres`
    - contactPhoneNumber: `312-555-9725`
    - creditRating: `F`
    - creditRisk: `HIGH`
    - customerCity: `Madrid`
    - customerCountry: `Spain`
    - eligibleApplication: `false`
    - ineligibleReason: `None`
    - pricingCode: `51`
    - pricingScore: `31`
    - productName: `Tacks`
    - requestAccountAmount: `20000`

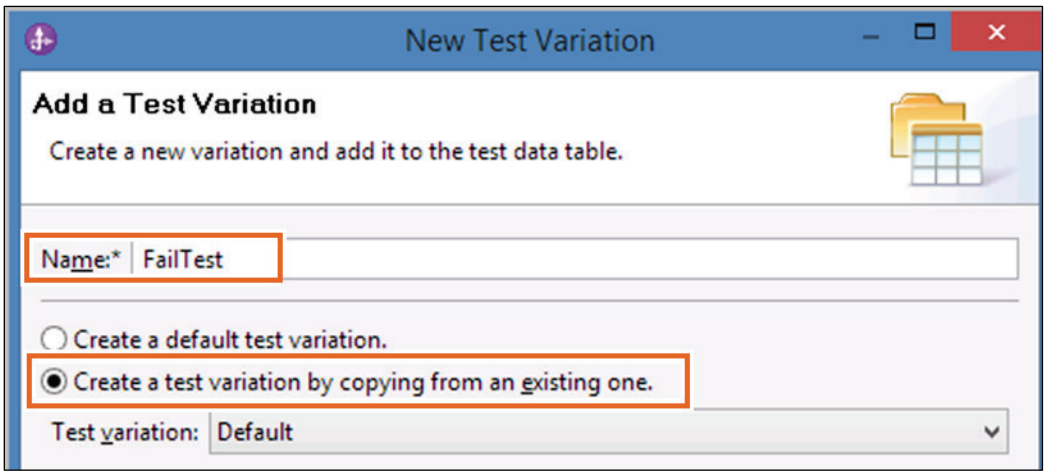


14. Save your changes.

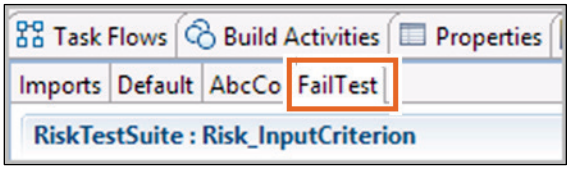
8. Create a FailTest variation by using an improper companyName value: BeanCo This variation demonstrates test failure.

1. In the **Test Data Table** view, click the **Add Variation** icon (look far right to the **Test Data Table** view if your window is maximized).

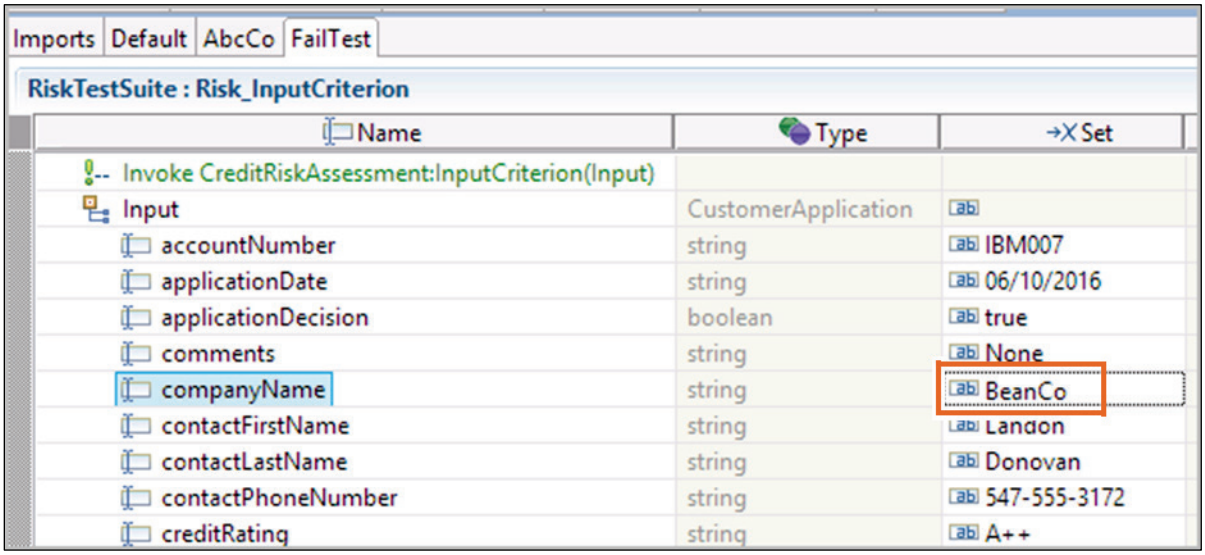
2. In the “Add a Test Variation” pane, enter the following information:
   - In **Name** field, type: `FailTest`
   - Select **Create a test variation by copying from an existing one**.
   - Verify that the **Test variation** field is set to **Default**.



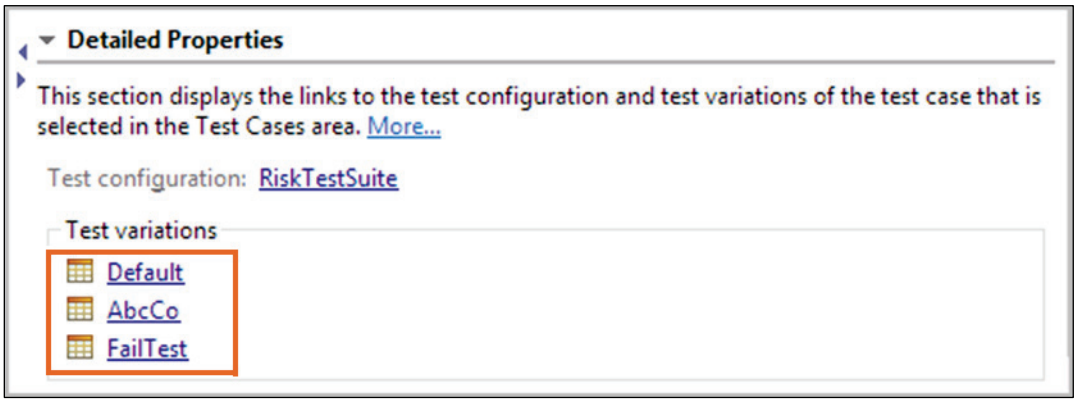
3. Click **Finish**.
   You now see a new tab in the **Test Data Table** that is named `FailTest`.



4. Change the companyName value in Input to BeanCo (a value that the CreditRiskAssessment rule set does not recognize).



5. Save your changes. Each one of your validation cases is listed in the **Detailed Properties** section of the test suite editor.



6. Close the **RiskTestSuite** tab.

Part 2. Running a test suite with operation-level test case

In this portion of the exercise, you run RiskTestSuite in the IBM Integration Designer test environment. The Default and AbcCo variations are expected to pass. As expected, the FailTest variation fails.

1. Run RiskTestSuite in the IBM Integration Designer test environment.

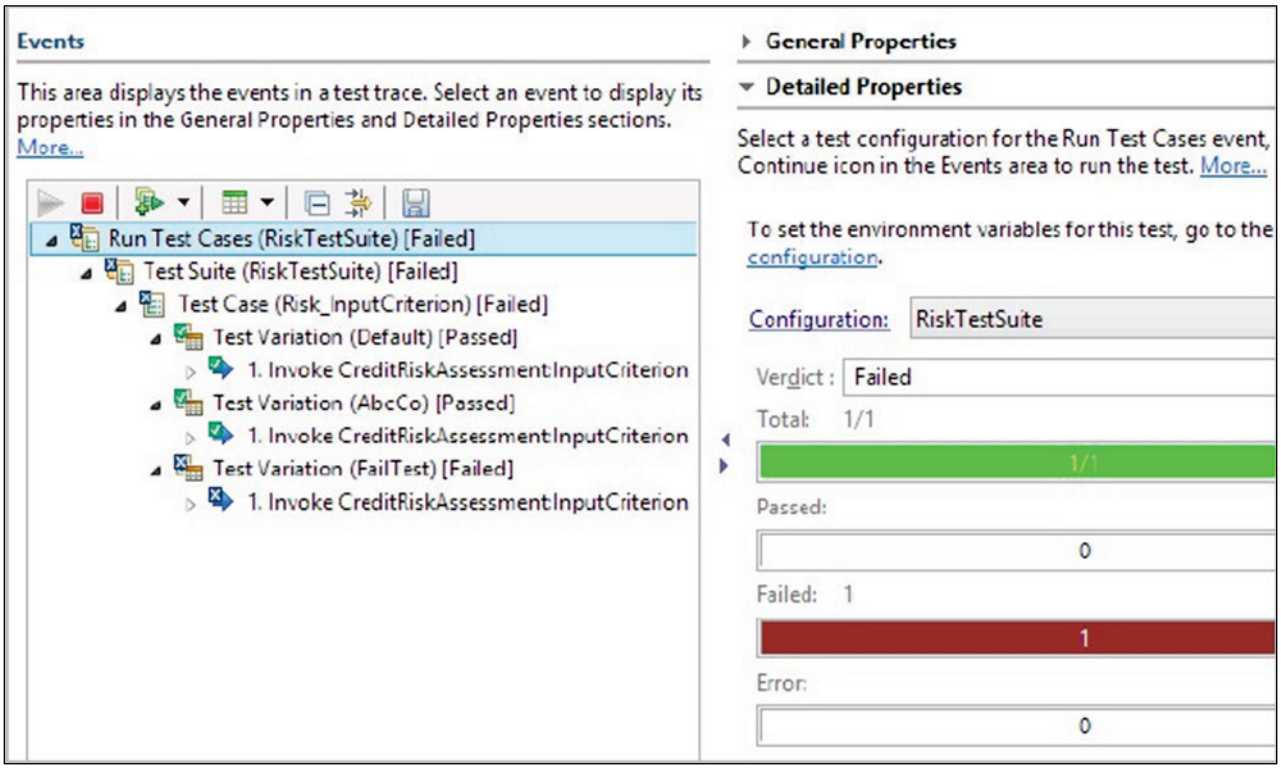
1. Double-click the desktop shortcut to start the process server, if stopped..
2. In the Business Integration view, expand **RiskTestProject**.
3. Right-click **RiskTestSuite** and click **Run Test Cases** from the menu.
4. Click the **Continue** icon in the Event window action bar.
5. At the “Select a Deployment Location” dialog box, click **IBM Process Server v8.6 at localhost** and click **Finish**.
6. At the User Login dialog box, accept the default values for **User ID** and **Password** and click **OK**. These fields are set to `admin` and `web1sphere` during product installation.

   When you run the suite, the modules that are required to do the operation-level tests are published and started: **FoundationModuleApp** and **RiskTestProjectApp**. It can take a few minutes depending on your system resources.

   After the modules are deployed and started, all three test cases run automatically.

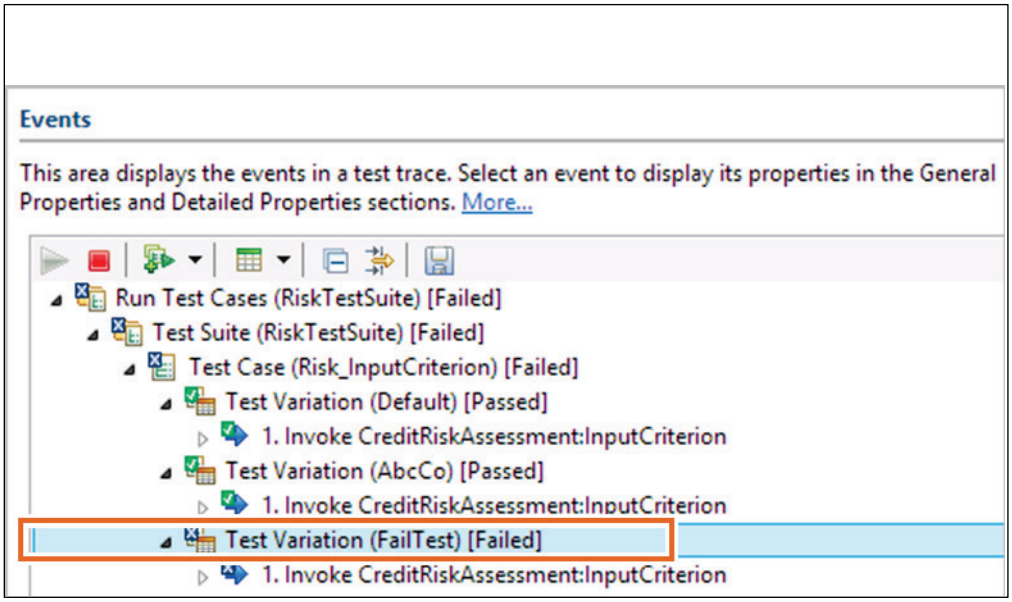
2. Examine the results from the test cases.

1. In the **Events** list, you see two test cases with a verdict of **[Passed]** and one with a verdict of **[Failed]**.



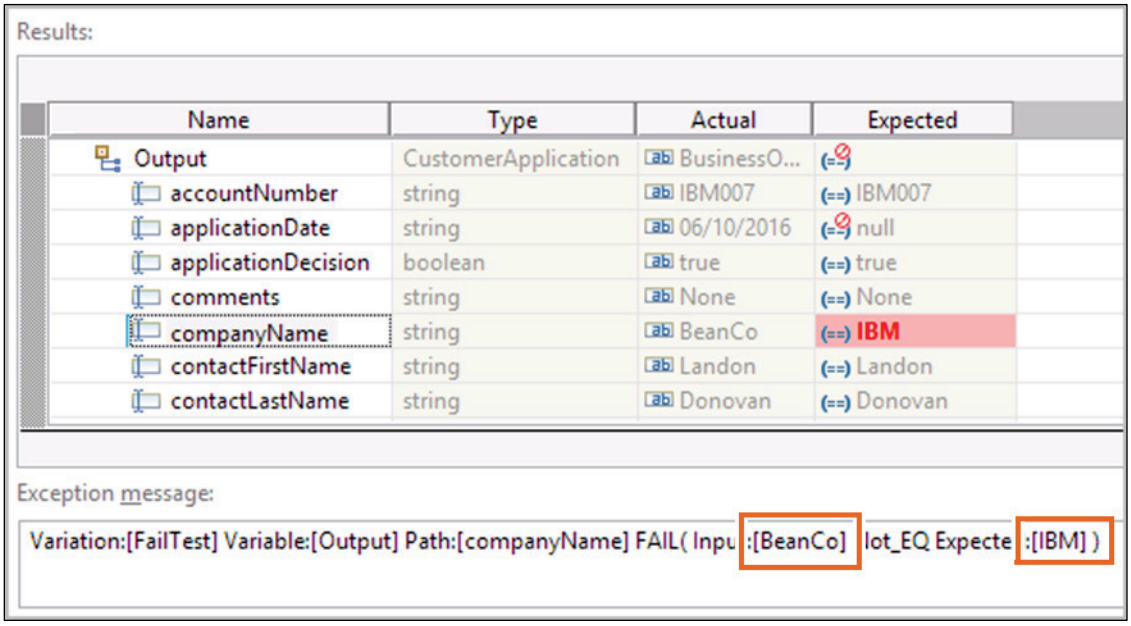
Note: If you do not see results similar to the screen capture above, you might want to check whether all the steps were followed correctly. Also, make sure that you save your changes.

2. In the **Events** list, select **Test Variation (FailTest) [Failed]**.



3. Examine the **Results** section.

   The cause of the test failure (the expected companyName versus the actual companyName) is highlighted, and the error is listed in the **Exception message** window. An **Exception trace** is also provided.



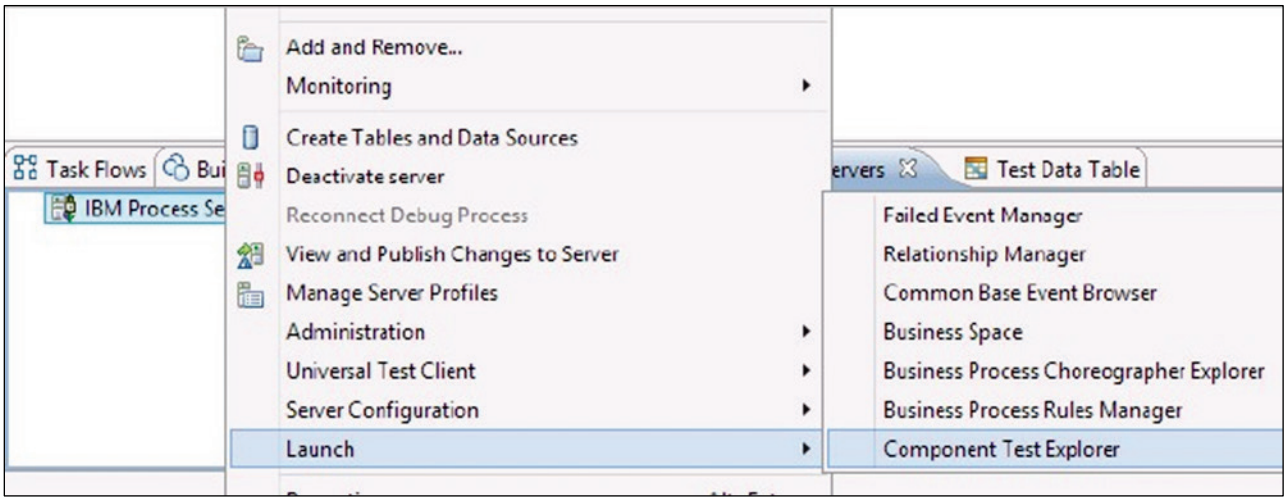
4. As time permits, explore the test variations that passed.
5. Close the test suite tab and click **No** when you are prompted to save the test trace.

3. Remove the applications from the server. Applications that are deployed during a component test run remain deployed until you manually remove them.

1. In the Servers view, right-click IBM Process Server v8.6 at localhost and click Add and Remove from the menu.
2. Click **Remove All** and click **Finish**. It takes a few minutes until the applications are removed from the server. Wait for the server status to change to **Started, Synchronized**.

Information: Though not used in this exercise, IBM Integration Designer also includes a web-based application that is named Component Test Explorer. You can use the Component Test Explorer to manage and run test suites that are deployed to an IBM Integration Designer test environment server. Using the Component Test Explorer, you can:

- Query and display deployed component test projects, test suites, and test cases
- Select and run one or more component test projects, test suites, or test cases and then display and automatically save the results
- Schedule specific times to automatically run one or more component test projects, test suites, or test cases



Part 3. Create a component test project with a scenario-based test case.

In this portion of the exercise, you create a component test project with a scenario-based test case. Scenario-based test cases invoke a series of interface operations. The test case that you create simulates one of the process paths in your solution. After creating the project, you run the test suite in the IBM Integration Designer integrated test client.

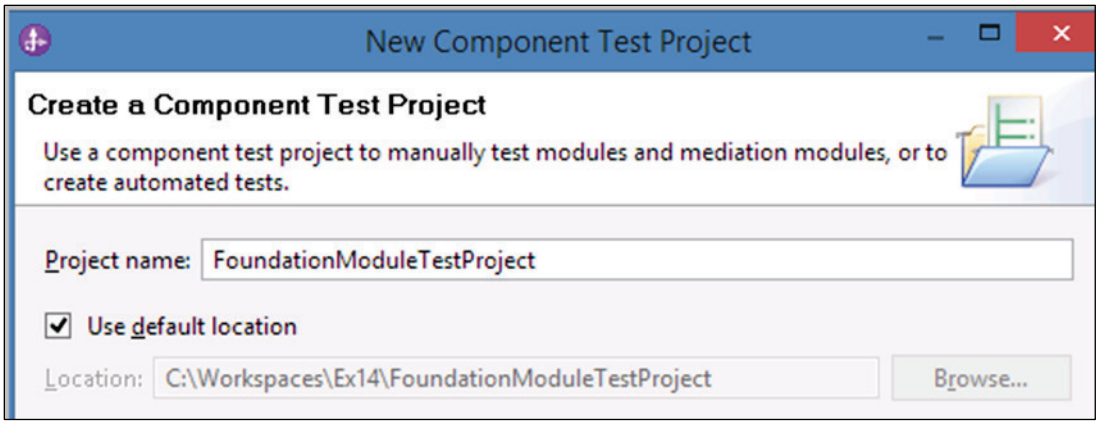
The path you test is for companyName ACME. Because the creditScore returned for ACME is 6, `ACME` has a creditRisk of `MED`. An application with a `MED` creditRisk takes this path through the AccountVerification process:

Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment > Final Application Review > Create Output > Account Verification Reply

1. Create a human task emulator to simulate human input.

In this portion, you create a test project named `FoundationModuleTestProject`.

1. Click **File > New > Component Test Project** from the menu options.
2. In the **Project name** field, type: `FoundationModuleTestProject`



3. Click **Finish**.

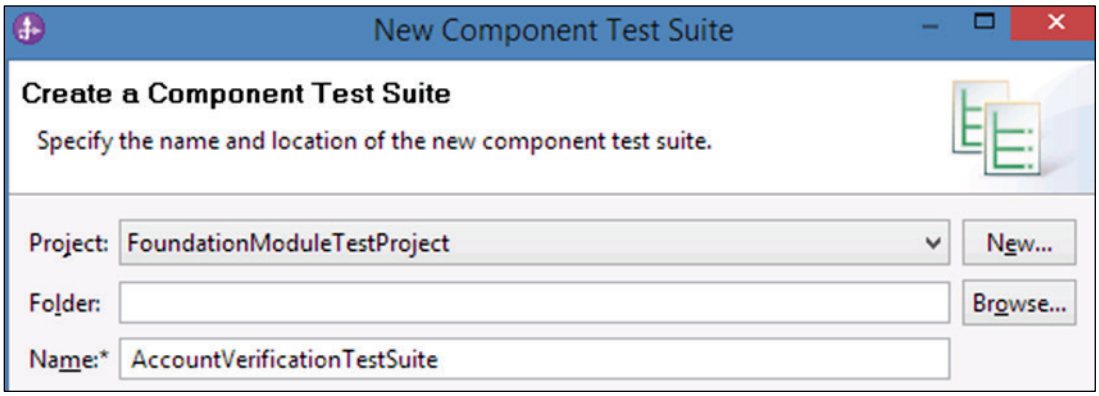
2. You will create a component test suite in FoundationModuleTestProject that is named AccountVerificationTestSuite.

The test suite is scenario-based and contains a test case that is named `AccountVerification_MED`. The project includes the following operations:

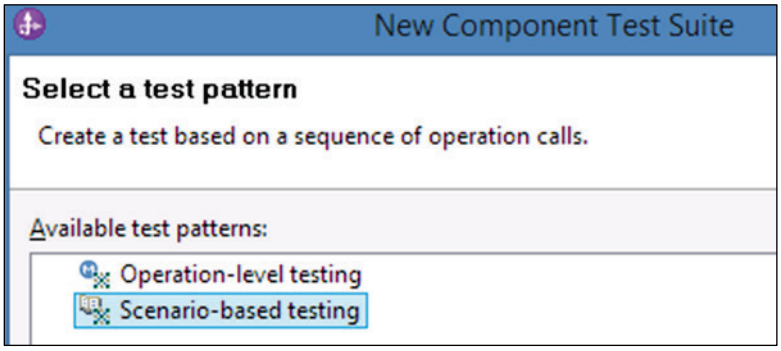
- InputCriterion operation in the AccountVerification interface
- InputCriterion operation in the DetermineApplicantEligibility interface
- calculateCreditScore operation in the creditScoreService interface
- InputCriterion operation in the CreditRiskAssessment interface
- InputCriterion operation in the FinalApplicationReview interface

1. In the Business Integration view, expand **FoundationModuleTestProject**.
2. Right-click **Test Suites** and click **New > Component Test Suite** from the menu.

3. At the “Create a Component Test Suite” pane, take the following actions:
   - Leave the **Folder** field empty.
   - In the **Name** field, type: `AccountVerificationTestSuite`

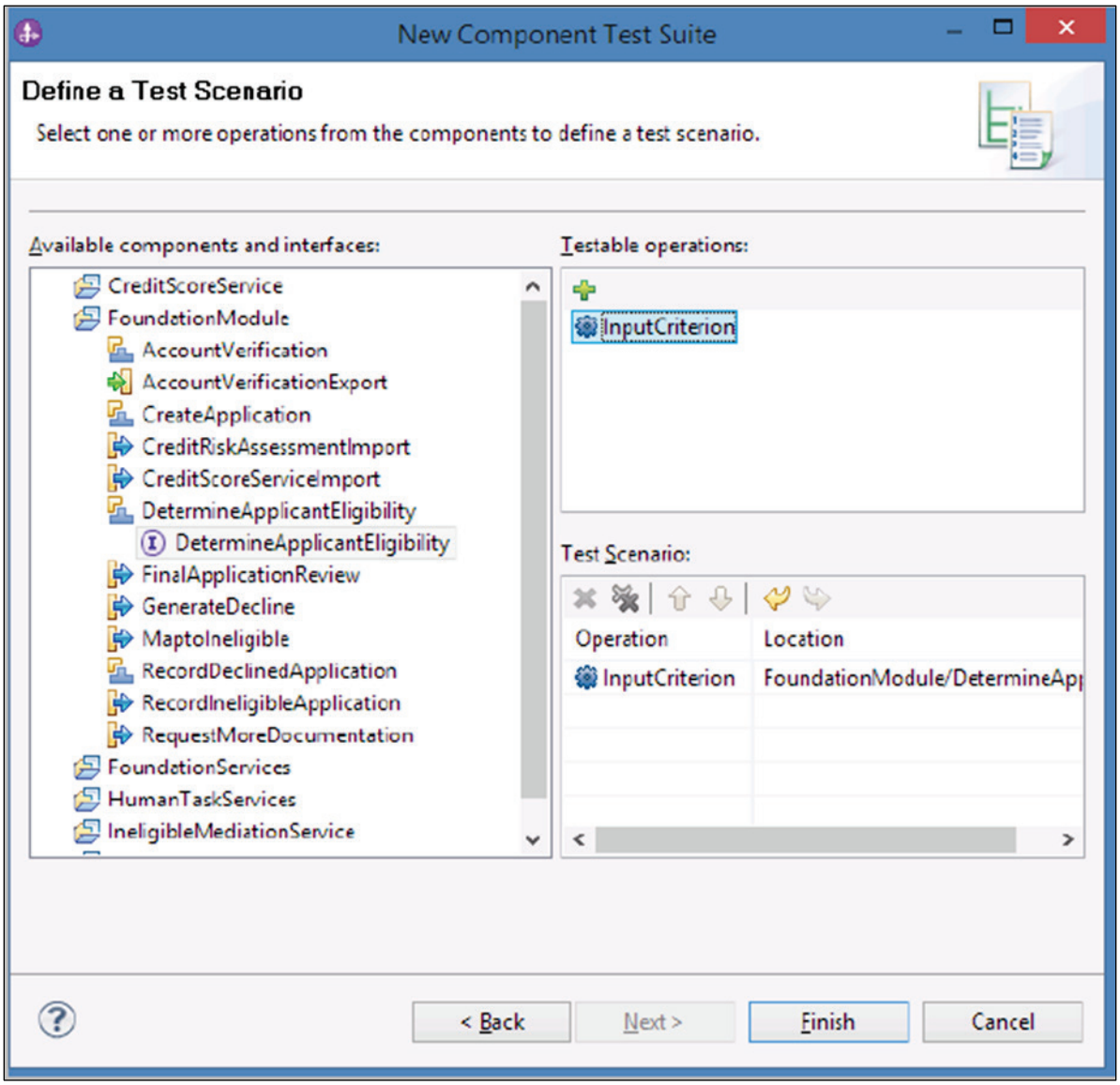


4. Click **Next**.
5. In the “Select a test pattern” pane, select **Scenario-based testing**.



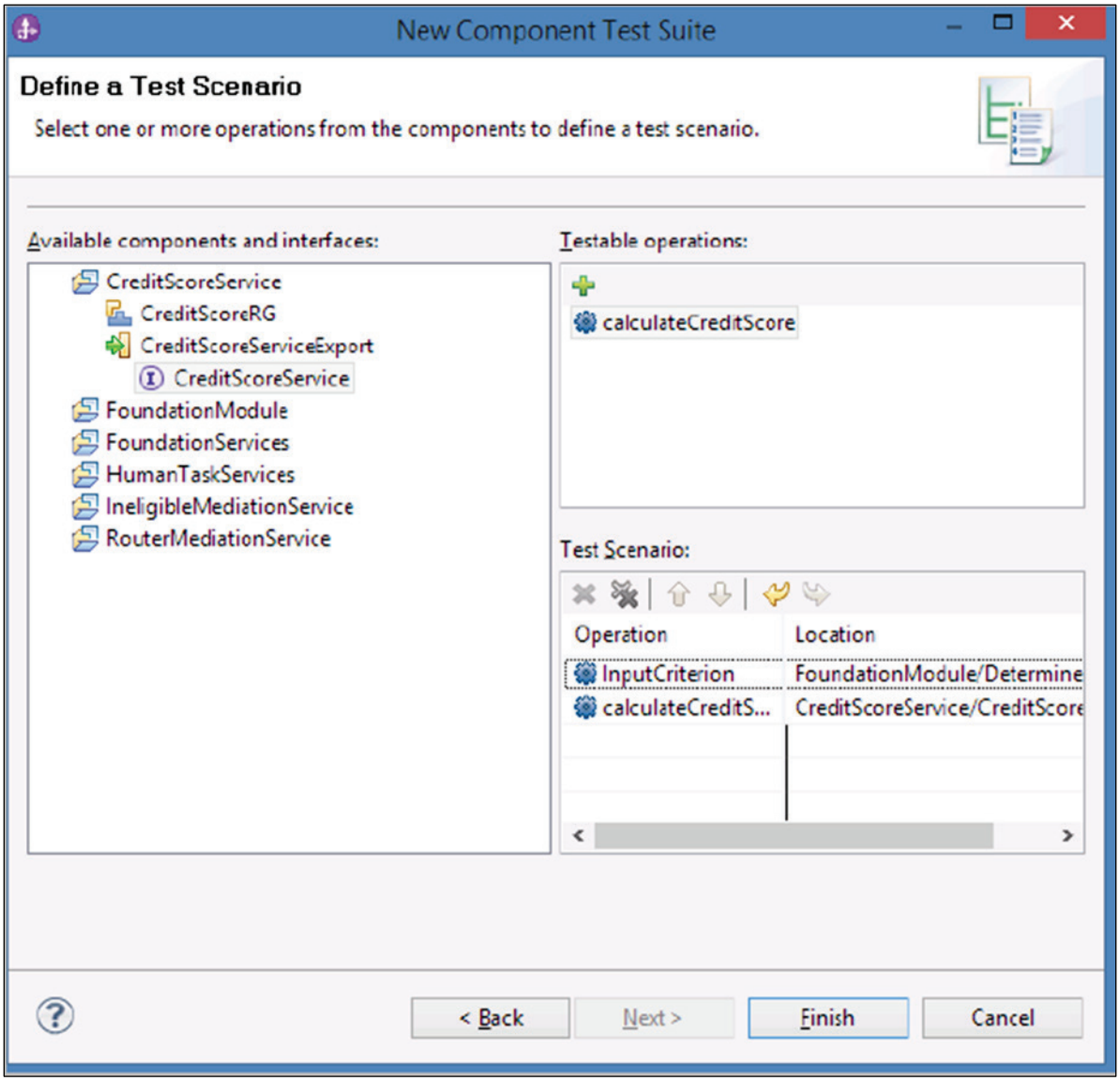
6. Click **Next**.

7. In the “Define a Test Scenario” pane, take the following actions:
   - In the **Test case name** field, type: `AccountVerification_MED`
   - In the “Available components and interfaces” window, expand **FoundationModule > DetermineApplicantEligibility** and select the **DetermineApplicantEligibility** interface.
   - In the “Testable operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.



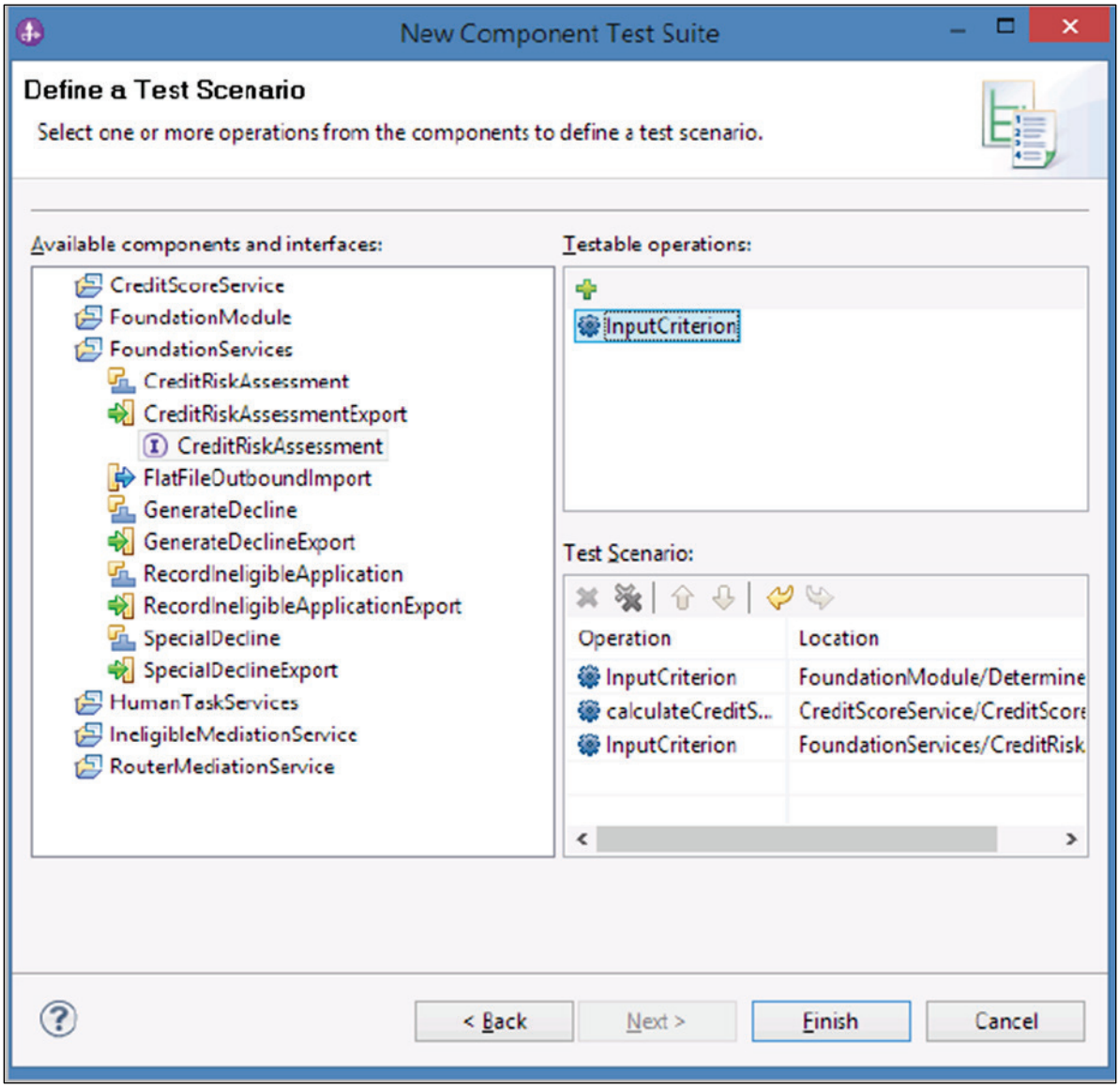
   - In the “Available components and interfaces” window, expand **CreditScoreService > CreditScoreServiceExport** and select the **CreditScoreService** interface.

- In the “Testable operations” window, double-click **calculateCreditScore** to add the operation to the “Test Scenario” window.



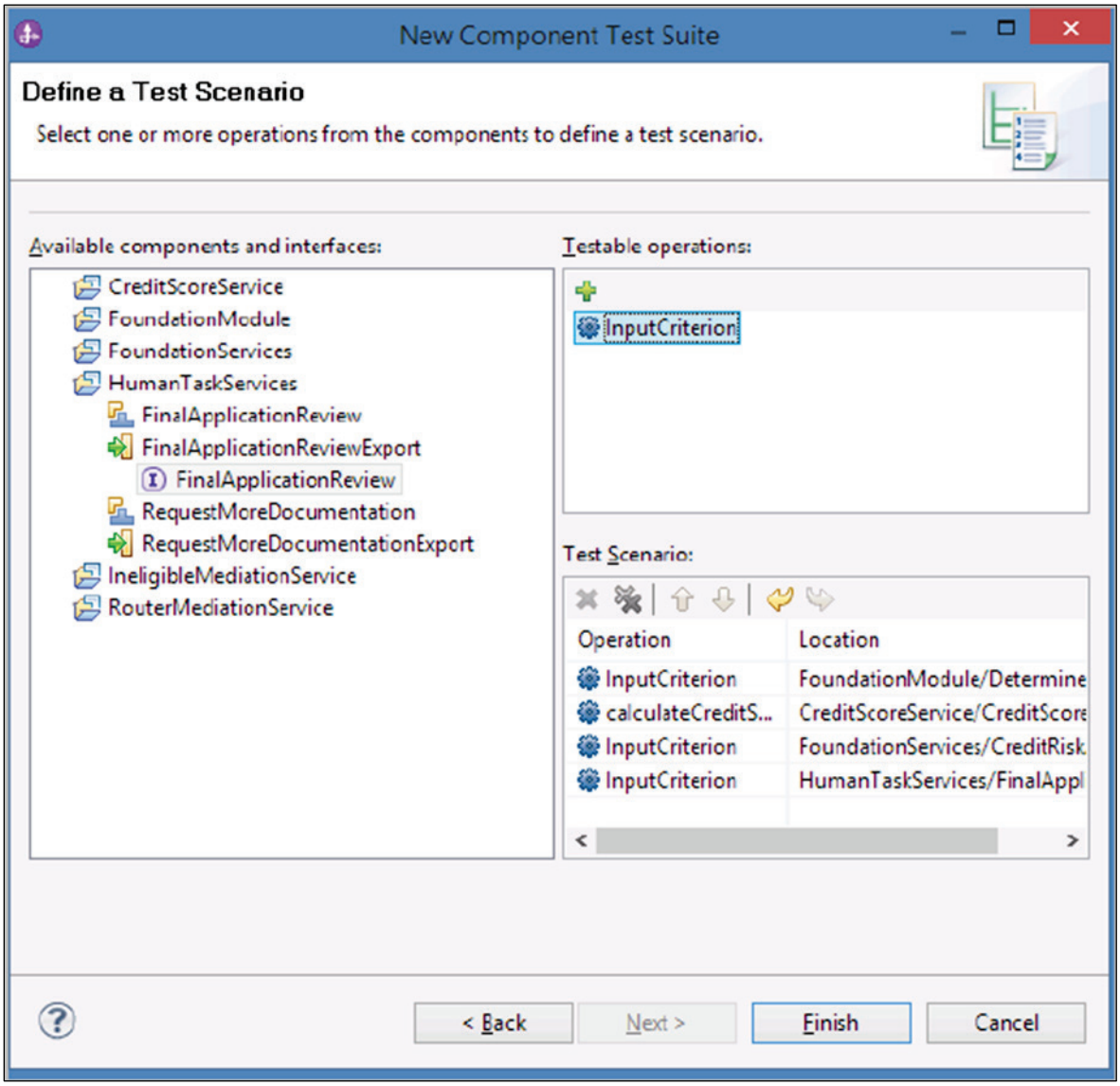
- In the “Available components and interfaces” window, expand **FoundationServices > CreditRiskAssessmentExport** and select the **CreditRiskAssessment** interface.

- In the “Testable operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.

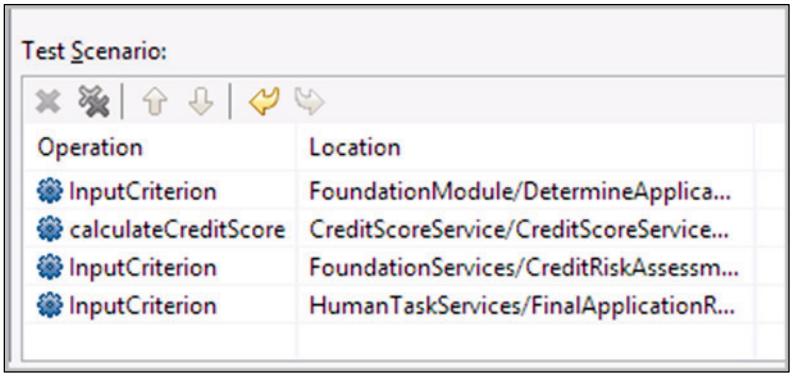


- In the “Available components and interfaces” window, expand **HumanTaskServices > FinalApplicationReviewExport** and select the **FinalApplicationReview** interface.

- In the “Testable Operations” window, double-click **InputCriterion** to add the operation to the “Test Scenario” window.



- Verify that the completed “Test Scenario” window contains the following four operations:



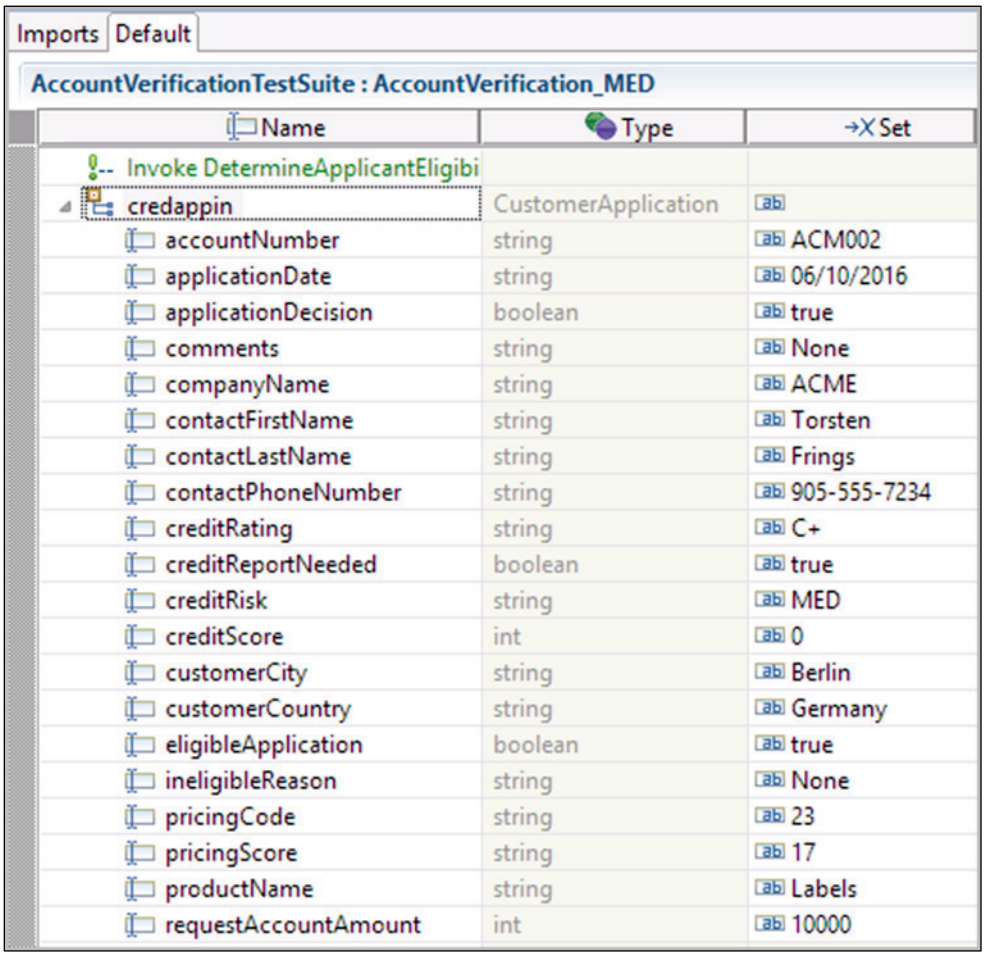
8. Click **Finish**.

2. Populate the credappin and credappout values for Invoke DetermineApplicantEligibility:InputCriterion in the Default test variation.

Use the credappin values in `C:\labfiles\Support Files\Ex13\EX13_Scenario_credappin.xml` and the credappout values in EX13_Scenario_credappout.xml. Set applicationDate in credappout to **Do Not Care**.

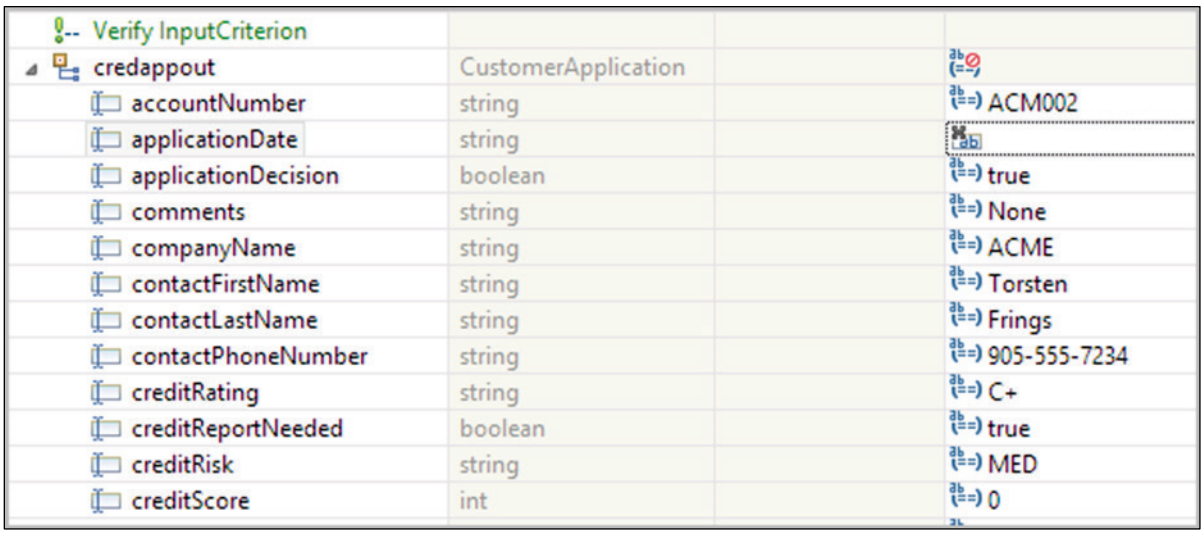
1. In the Test Data Table tab, go to Invoke DetermineApplicantEligibility:InputCriterion.

   You can double-click the test Data Table tab to maximize the view.
2. Right-click **credappin** and click **Import from File**.
3. Browse to `C:\labfiles\Support Files\Ex13\`.
4. Select **EX13_Scenario_credappin.xml** and click **Open**.



5. Scroll to the bottom of the input data and locate the `credappout` section under `Verify InputCriterion`.

6. In the `credappout` row, right-click the value in the **Expected** column and click **Add Children** from the menu.
7. Right-click **credappout** and click **Import from File** from the menu.
8. Browse to `C:\labfiles\Support Files\Ex13\`.
9. Select **EX13_Scenario_credappout.xml** and click **Open**.
10. When the values are imported, right-click the **Expected** value for `applicationDate` and click **Set To > Do Not Care** from the menu.

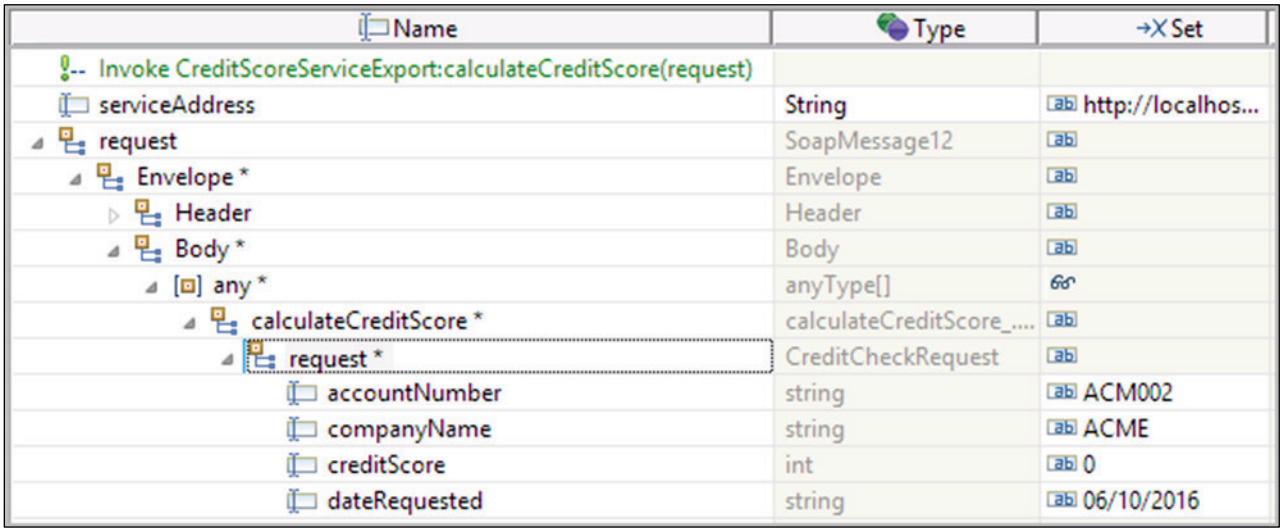


2. Populate the request and calculateCreditScoreReturn values for Invoke CreditScoreServiceExport:calculateCreditScore in the Default test variation.

Use the request values in `C:\labfiles\Support Files\Ex13\EX13_Scenario_request.xml` and the calculateCreditScoreReturn values in `EX13_Scenario_return.xml`. Set the dateRequested value in calculateCreditScoreReturn to **Do Not Care**.

1. Scroll to Invoke CreditScoreServiceExport:calculateCreditScore.
2. Expand **request > Envelope > Body > any > calculateCreditScore > request**.
3. Right-click **request** and click **Import from File** from the menu.
4. Browse to `C:\labfiles\Support Files\Ex13\`.

5. Select **EX13_Scenario_request.xml** and click **Open**.

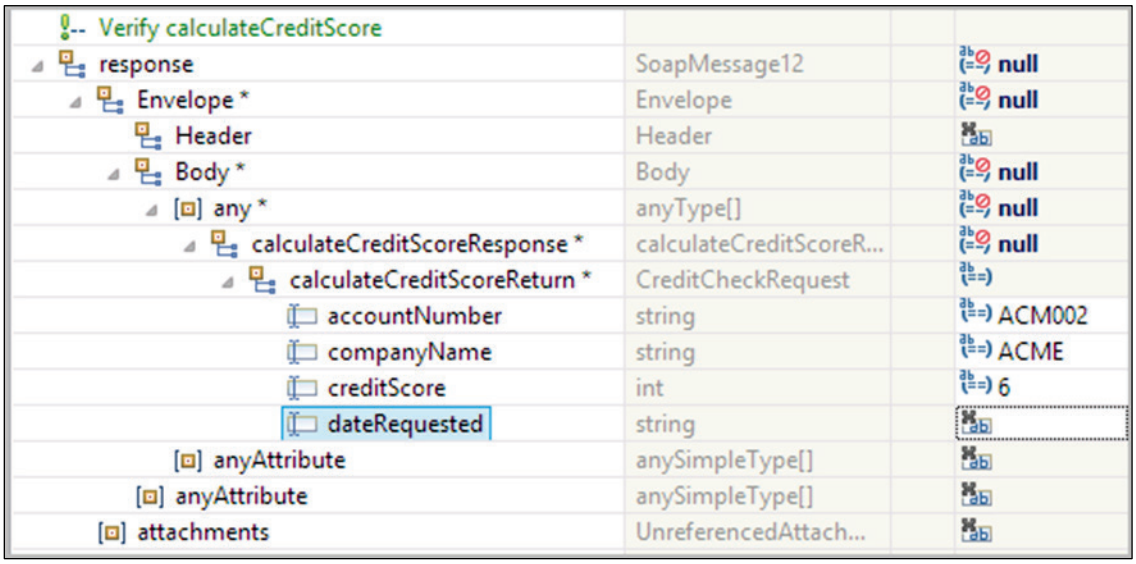


   To verify the values, you can expand **request** again after doing the import. Alternatively, enter the following values:

   - accountNumber: `ACM002`
   - companyName: `ACME`
   - creditScore: `0`
   - dateRequested: `06/10/2016`

6. Scroll to **Verify calculateCreditScore**.
7. In the **response** row, right-click the value in the **Expected** column and click **Add Children** from the menu.
8. Expand **response > Envelope > Body > any > calculateCreditScoreResponse > calculateCreditScoreReturn**.
9. Right-click **calculateCreditScoreReturn** and click **Import from File** from the menu.
10. Browse to `C:\labfiles\Support Files\Ex13\`.
11. Select **EX13_Scenario_return.xml** and click **Open**.
12. If necessary, expand **calculateCreditScoreReturn**.

13. In the **dateRequested** row, right-click the value in the **Expected** column and click **Set To > Do Not Care** from the menu.



Alternatively, enter the following values:

- accountNumber: `ACM002`
- companyName: `ACME`
- creditScore: 6
- dateRequested: (set to **Do Not Care**)

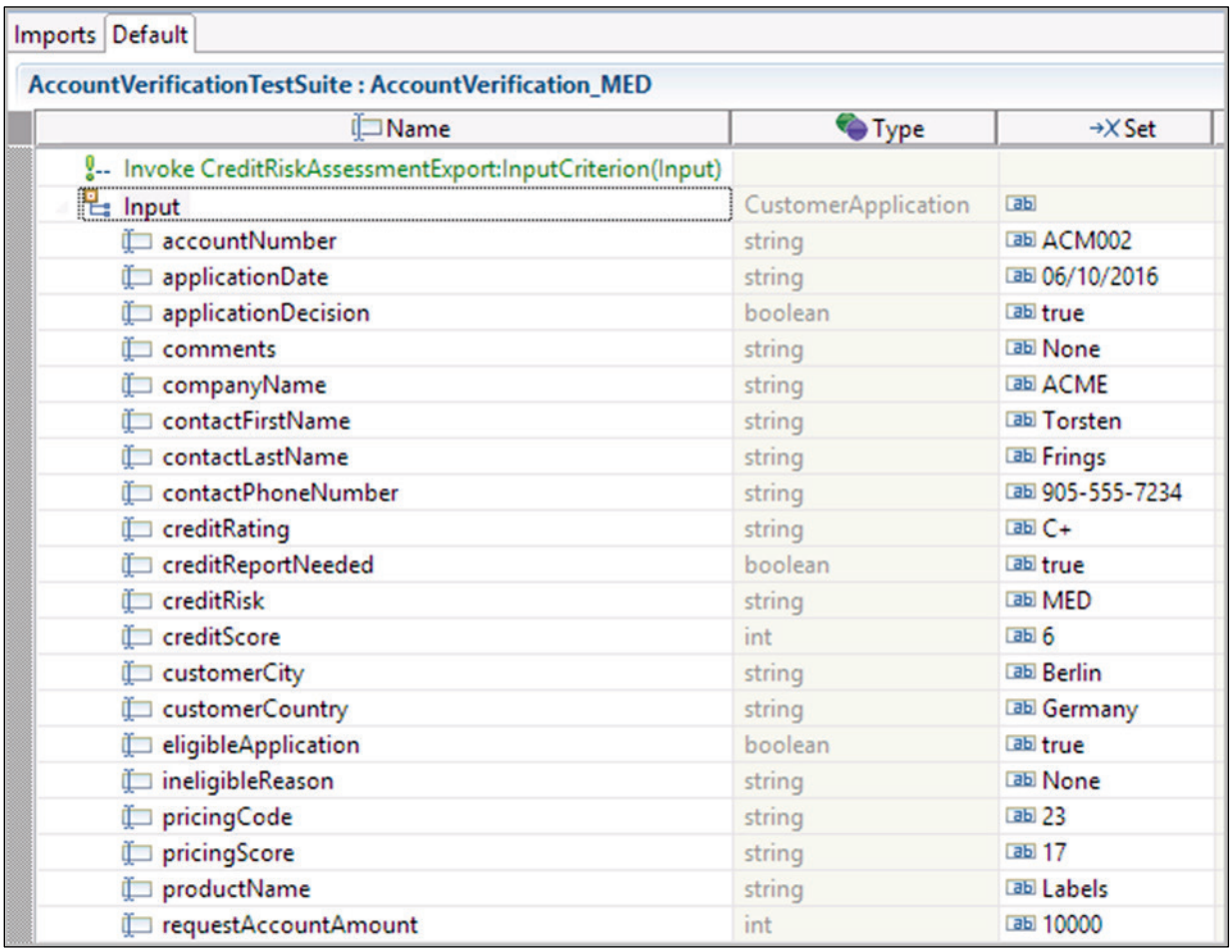
14. Save your changes.

3. Populate the Input and Output values for Invoke CreditRiskAssessmentExport:InputCriterion in the Default test variation.

Use the Input values in C:\labfiles\Support Files\Ex13\EX13_Scenario_input.xml and the Output values in EX13_Scenario_output.xml.

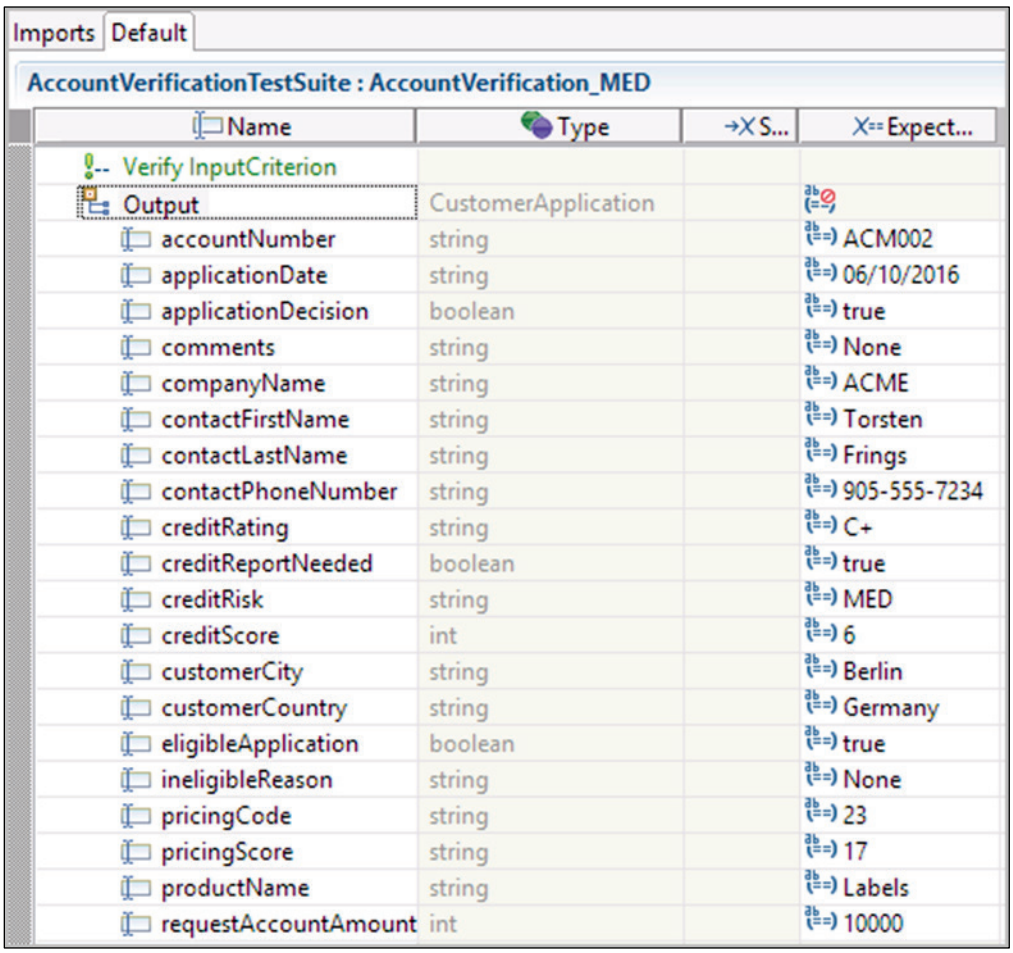
1. Scroll to Invoke CreditRiskAssessmentExport:InputCriterion(Input).
2. Right-click **Input** and click **Import from File**.
3. Browse to `C:\labfiles\Support Files\Ex13\`.

4. Select **EX13_Scenario_input.xml** and click **Open**.
   Alternatively, enter the values. If you enter the values manually, change the **creditScore** value to 6.



5. Scroll to **Verify InputCriterion**.
6. In the **Output** row, right-click the value in the **Expected** column and click **Add Children** from the menu.
7. Right-click **Output** and click **Import from File** from the menu.
8. Browse to `C:\labfiles\Support Files\Ex13\`.
9. Select **EX13_Scenario_output.xml** and click **Open**.

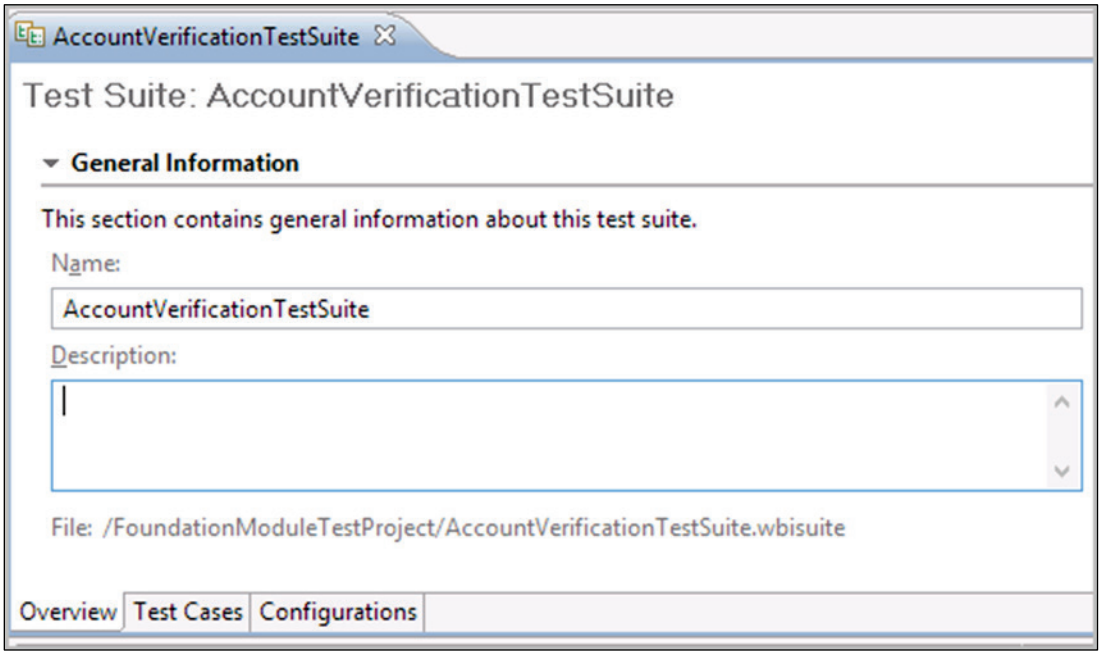
Alternatively, enter the values. If you enter the values manually, change the `creditScore` value to 6.



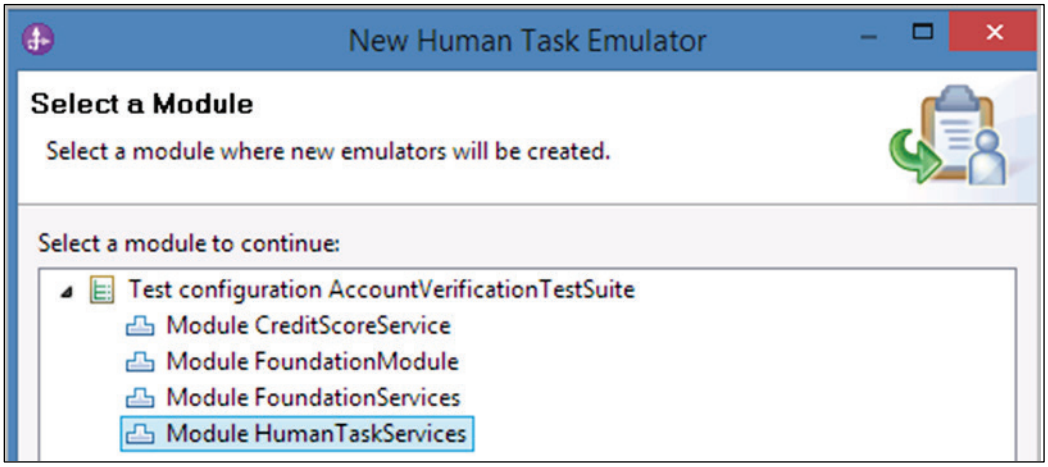
10. Save your changes.

4. Create a human task emulator for the final operation in the test case: the InputCriterion operation in the FinalApplicationReview interface.

1. On the AccountVerificationTestSuite tab, switch to the Configurations tab.

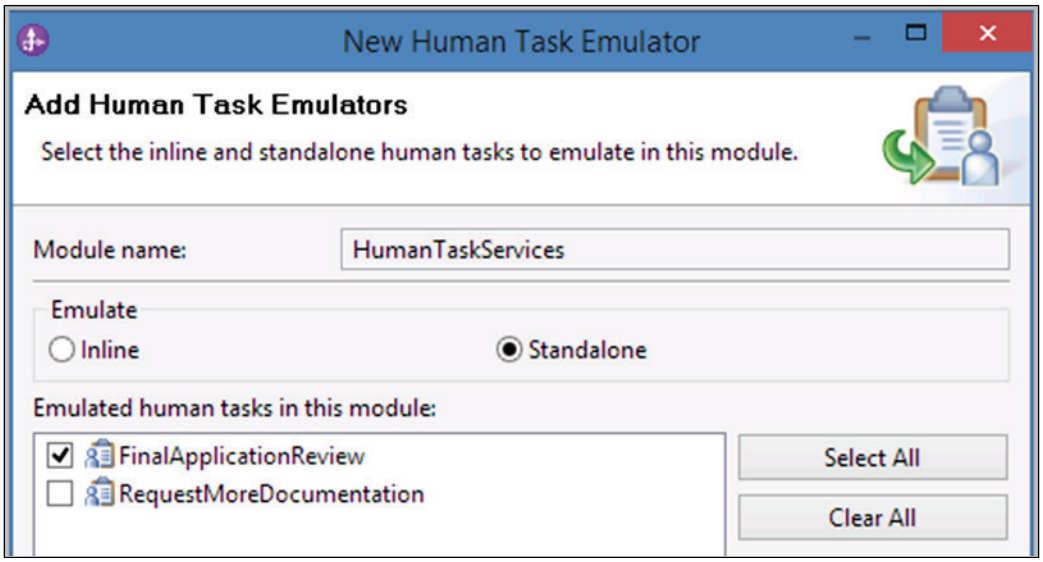


2. Under **Module HumanTaskServices**, right-click **Human Task Emulators** and click **Add > Human Task Emulator**.
3. At the “Select a Module” panel, select **Module HumanTaskServices**.

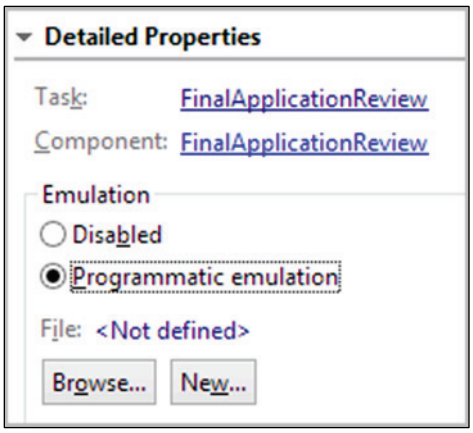


4. Click **Next**.

5. At the Add Human Task Emulators pane:
   - In the **Emulate** section, select the **Standalone** option.
   - In the “Emulated human tasks in this module” window, select **FinalApplicationReview**.

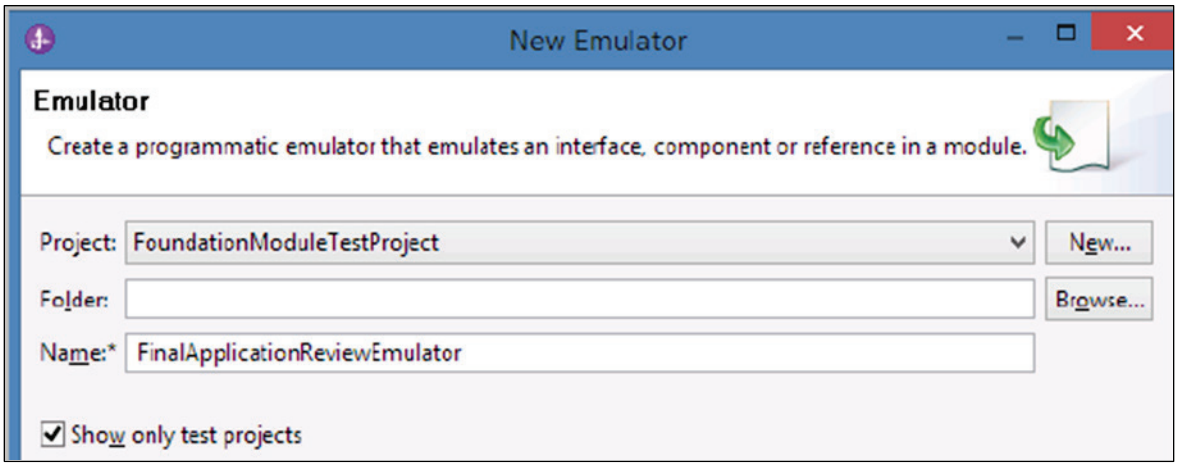


6. Click **Finish**.
7. In the **Detailed Properties** section, in the **Emulation** box, click **Programmatic emulation**.

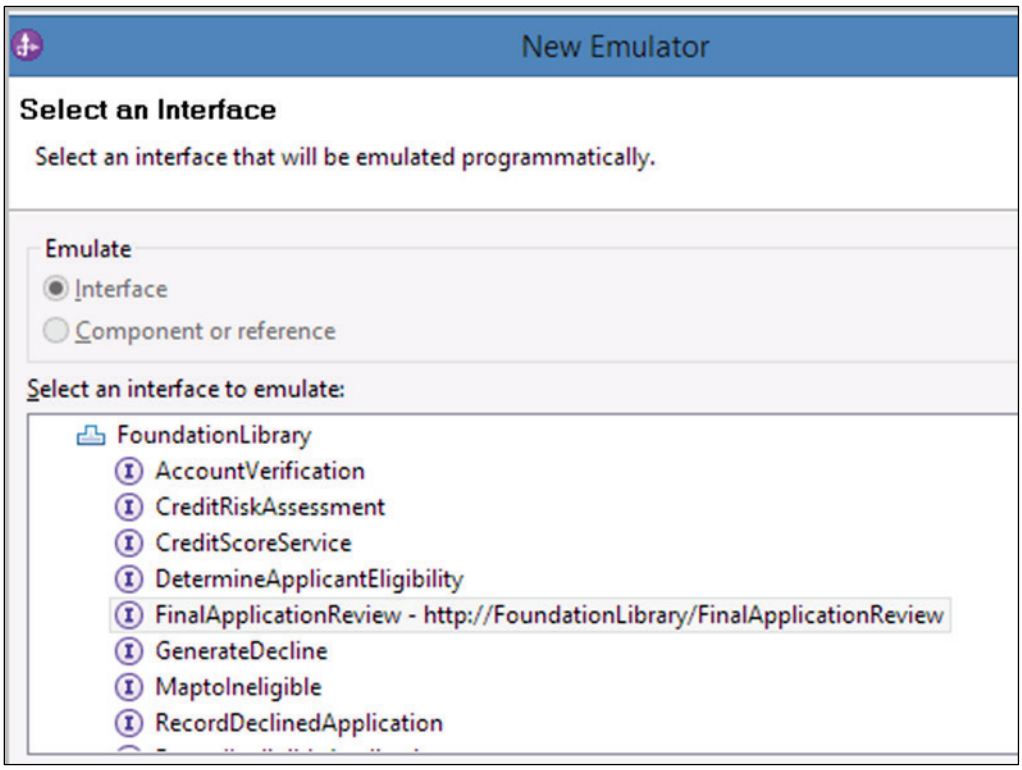


8. Click **New**.

9. In the New Emulator dialog box, on the Emulator pane, take the following actions:
   - Verify that the **Project** field is set to: `FoundationModuleTestProject`
   - Leave the **Folder** field empty.
   - In the **Name** field, type: `FinalApplicationReviewEmulator`

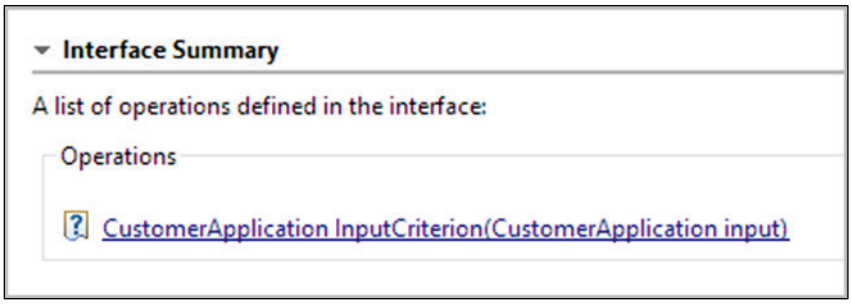


10. Click **Next**.
11. At the “Select an Interface” pane, verify that the **FinalApplicationReview** interface is selected.

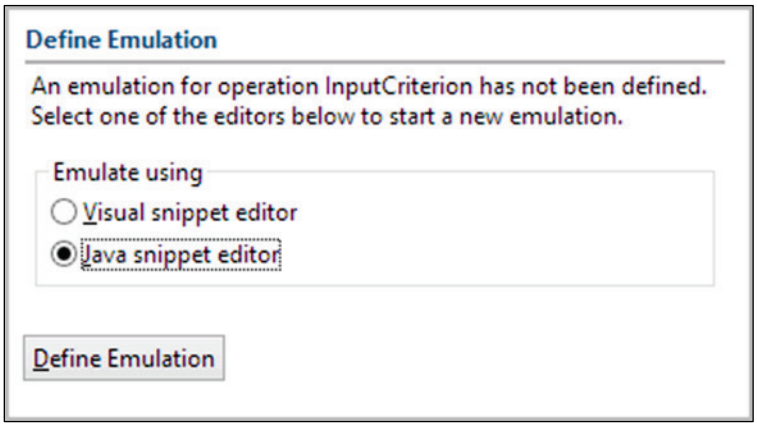


12. Click **Next**.
13. At the **Java Class** pane, accept the default options.

14. Click **Finish**.
15. In the **Interface Summary** section, click the **CustomerApplication_InputCriterion(CustomerApplication_input)** link.



16. Select **Java snippet editor**.



17. Click **Define Emulation**.
18. In Windows Explorer, browse to `C:\labfiles\Support Files\EX13\`.
19. Open `FinalApplicationReviewEmulatorCode.txt` in a text editor such as Notepad.
20. Copy the code and paste it in the Java Editor over the existing text. Be sure to remove `return null;`

Alternatively, enter the following Java code:

```
System.out.println
(" Start - FinalApplicationReview Emulation route ...");
System.out.println
(" End - FinalApplicationReview Emulation route ..."); return input;
```

21. Save your changes and close the **FinalApplicationReviewEmulator** tab.
22. Close `FinalApplicationReviewEmulatorCode.txt` and Windows Explorer.

23. On the test suite editor tab, in the **Detailed Properties** section, verify that the programmatic emulator file is listed in the **Emulation** box. Also, confirm that **Immediately** is selected in the **Claim** box.
24. Save your changes and close the **AccountVerificationTestSuite** tab.

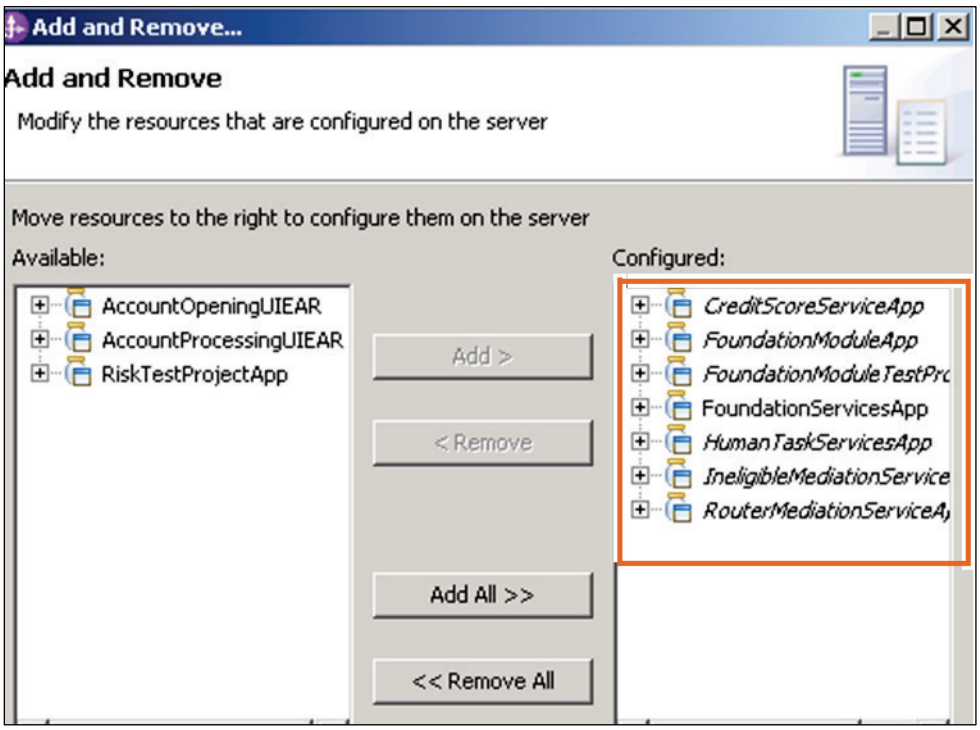
Part 4. Run component test project test suites in the IBM Integration Designer integrated test environment.

In this portion of the exercise, you run the AccountVerificationTestSuite in IBM Integration Designer. To run AccountVerificationTestSuite:

1. Start the server (if it is not already running) and deploy all of the modules.

1. Start the process server if not started.
2. In the **Servers** view, right-click **IBM Process Server v8.6 at localhost** and click **Add and Remove**.
3. Click **Add All** to include the projects in the **Configured** list.

   If system resources are an issue, do *not* publish AccountOpeningUIEAR, AccountProcessingUIEAR, or RiskTestProjectApp.



4. Click **Finish**.
5. Wait until the modules are published and started. It can take several minutes, depending on your system resources. To see the status of the modules, expand **IBM Process Server v8.6 at localhost** in the **Servers** view.

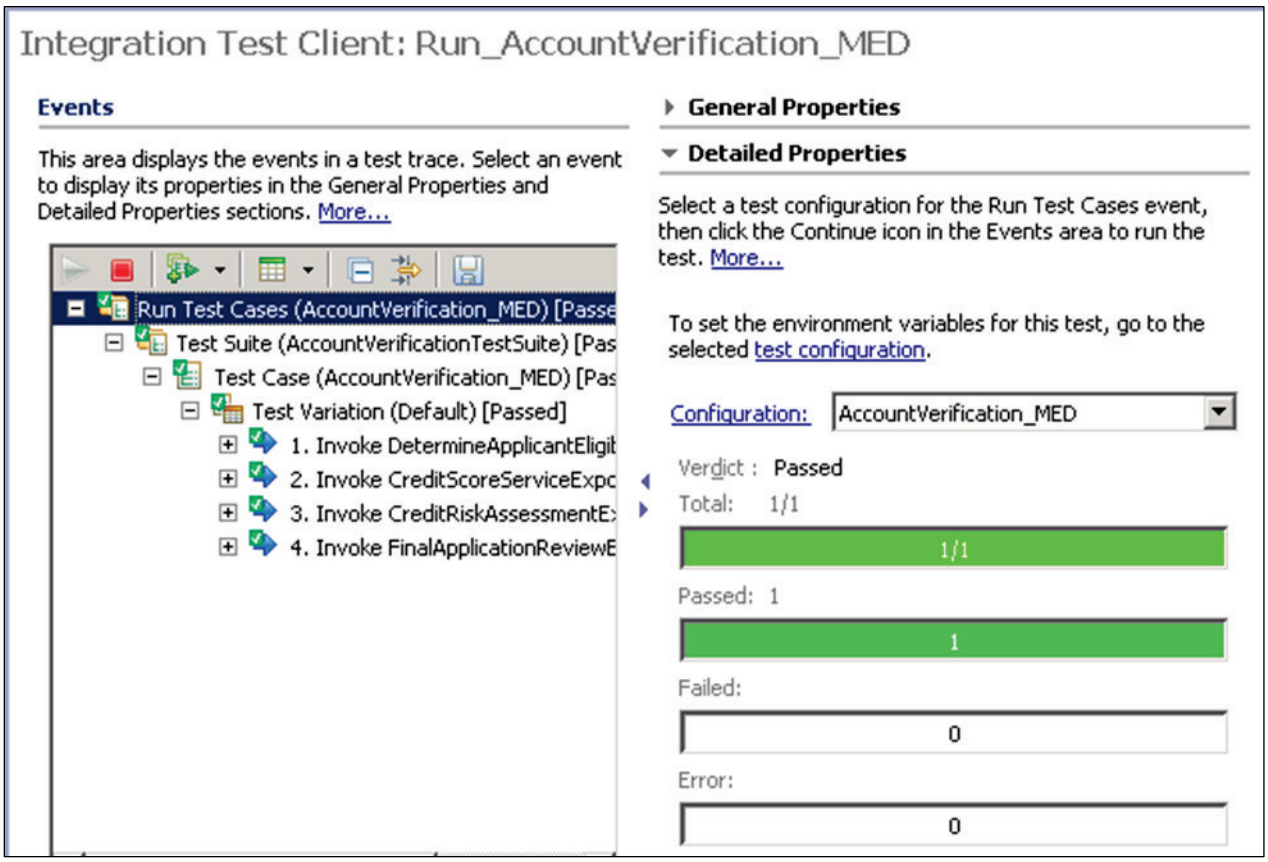
If any module is not started, then right-click the module and click **Restart**. If the status of the server is **Publishing**, then right-click the server and click **View and Publish Changes to Server**.

You can also look for `Application started: [ApplicationName]` messages in the **Server Logs** view.

2. Run the test suite in IBM Integration Designer.

1. In the Business Integration view, expand FoundationModuleTestProject > Test Suites > AccountVerificationTestSuite.
2. Right-click **AccountVerification_MED** and click **Run Test Cases** from the menu.
3. Click **Continue** on the **Events** toolbar.
4. When the **Deployment Location** dialog box is displayed, select **IBM Process Server v8.6 at localhost** and click **Finish**.
5. When the **User Login** prompt is displayed, accept the default options for **User Name** and **Password**; click **OK**.

When the test run is complete, you see four events that represent the invocation of the interface operations. If you receive an error, you can redo the component test scenario and confirm that all the entries were done correctly.



If you do not see the results similar to the screen capture, then make sure that you completed all the steps. Also, make sure that you save your changes.

6. As time permits, explore the test results.
7. Close the test tab and do not save the test trace.

3. Remove all of the projects from the server.

1. In the Servers view, right-click IBM Process Server v8.6 at localhost and click Add and Remove from the menu.
2. Click **Remove All** and click **Finish**.
3. Close IBM Integration Designer.

Results:
In this exercise, you created and ran component test projects in the IBM Integration Designer test environment.

IBM Training

IBM Process Center

IBM Business Process Manager V8.6

IBM Training

Unit objectives

- Describe the purpose and business value of IBM Process Center
- Define the components of IBM Process Center
- Describe how to use the IBM Process Center Console to deploy, test, and manage IBM BPM applications

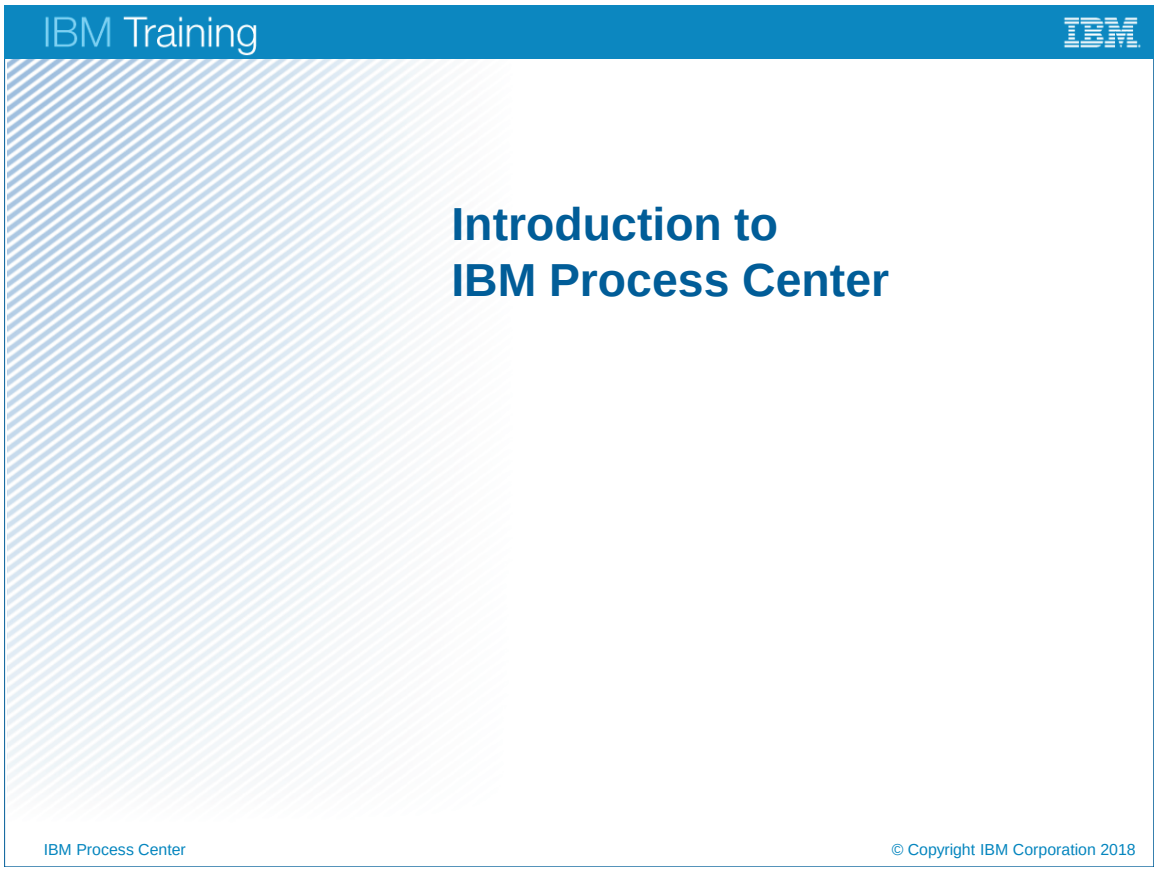
Unit objectives

IBM Training

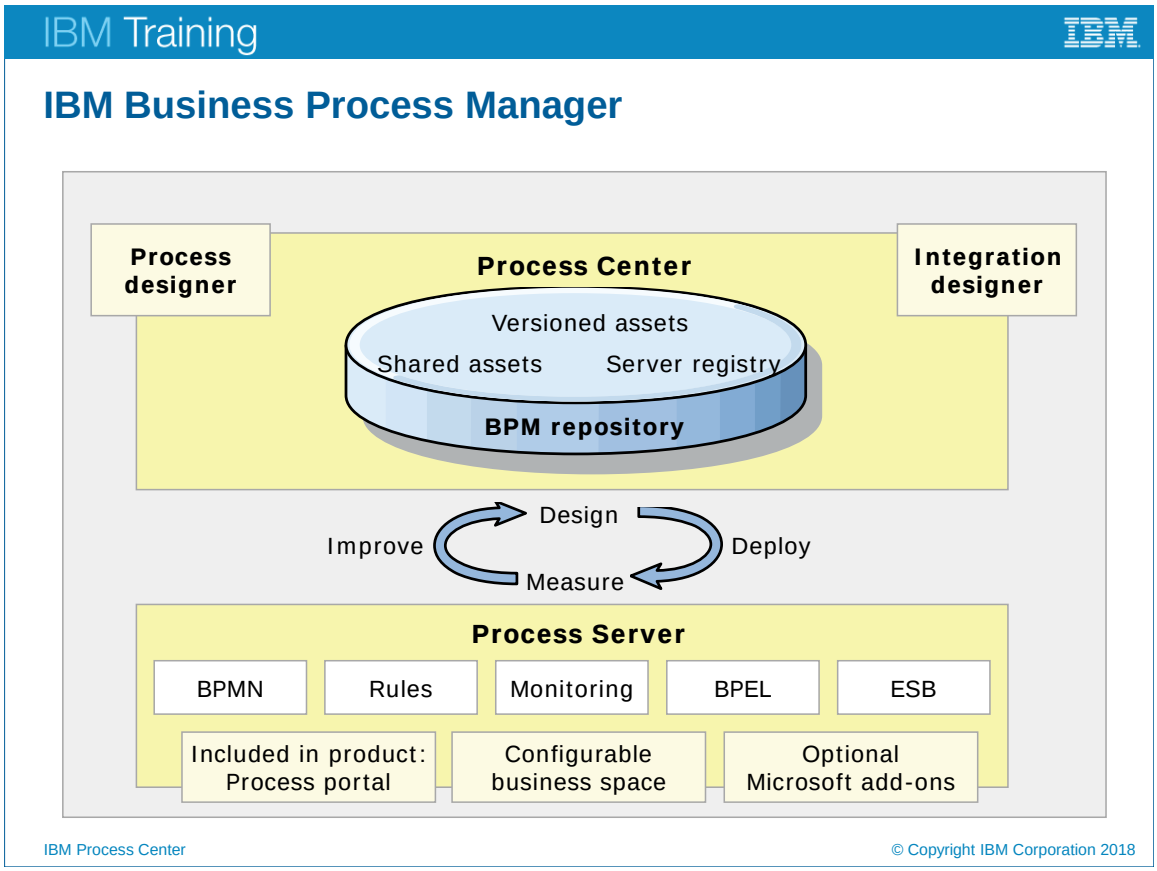
Topics

- Introduction to IBM Process Center
- Components of IBM Process Center
- Manage, test, and deploy IBM BPM applications

Topics



Introduction to IBM Process Center

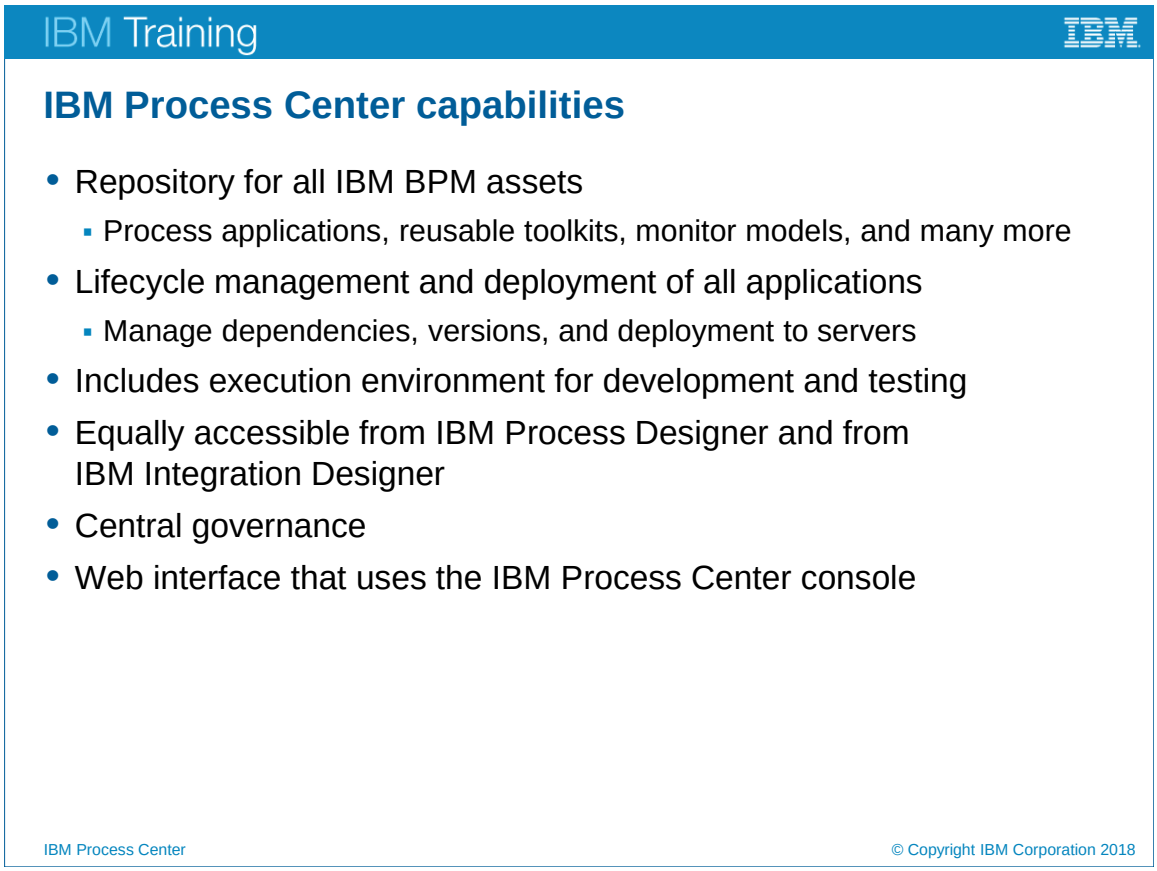


IBM Business Process Manager

IBM Business Process Manager uses a concept that is called the shared model. It means that no matter what is being done within the overall solution, it has only one common repository along with a single representation of that solution. Therefore, it is impossible to get two phases of the same solution out of sync with each other.

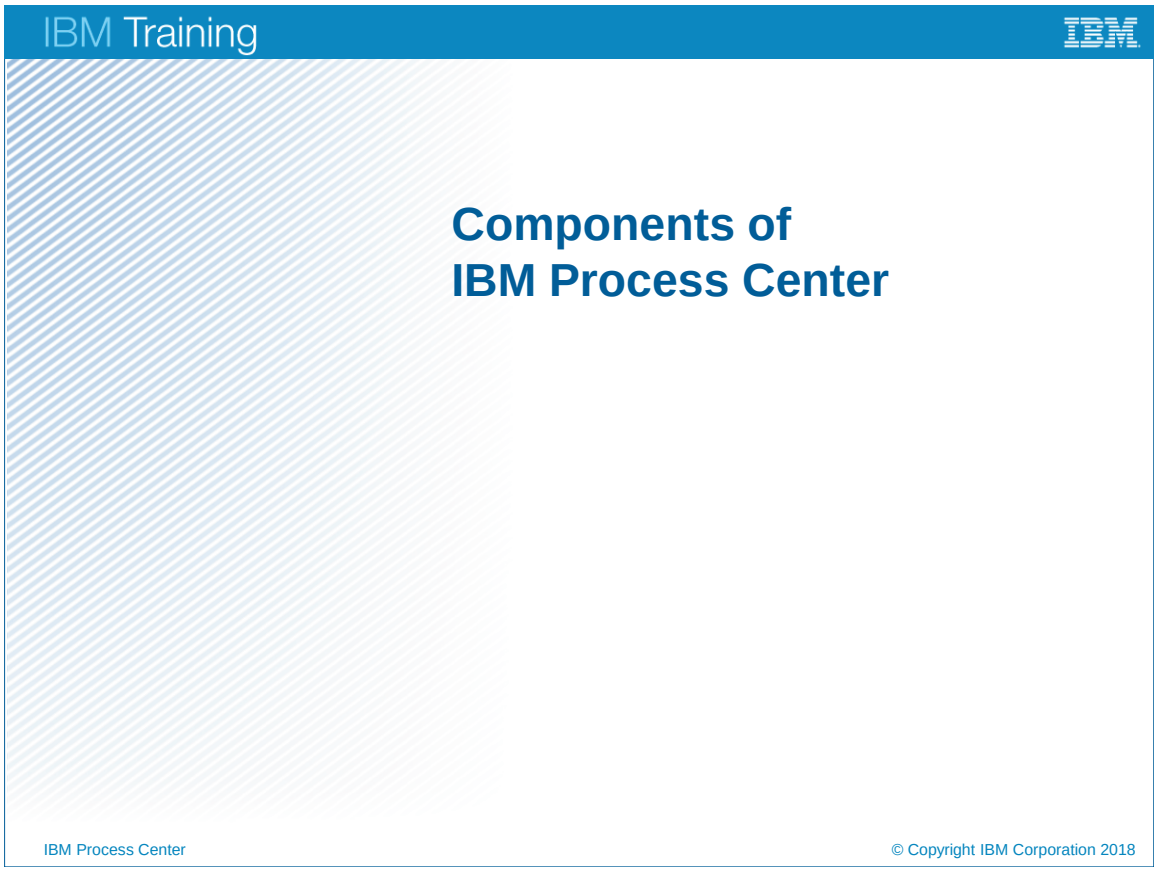
This shared model is realized through the IBM Business Process Manager component that is called the IBM Process Center. IBM Process Center is a key component within IBM Business Process Manager. Part of the IBM Process Center is a data repository, which is called the repository. Within the repository, there exists the representation of the solution. The IBM BPM tool connects as a client to the IBM Process Center to obtain copies of the solution to work on. When a user changes an artifact and saves those changes, the results are written back to the repository.

The IBM Process Center repository is implemented as tables within a database (commonly DB2).



IBM Process Center capabilities

You can use the IBM Process Center repository to share artifacts with other users who are developing process applications and toolkits.

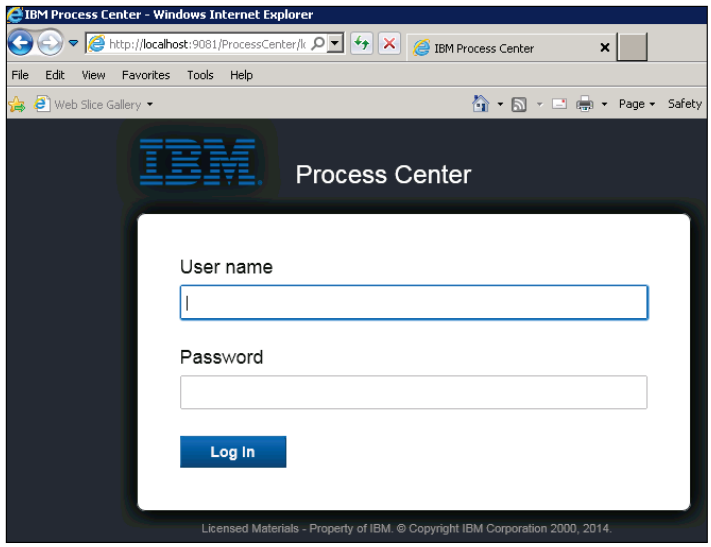


Components of IBM Process Center

IBM Training

IBM Process Center Console

- The IBM Process Center Console provides a web-based interface for managing the Process Center maintained projects
- It provides the tools that are needed to maintain the repository



IBM Process Center

IBM Process Center Console

The Process Center Console provides a web-based interface for managing the Process Center maintained projects. The default URL for IBM Process Center Console is: `http://localhost:9080/ProcessCenter`

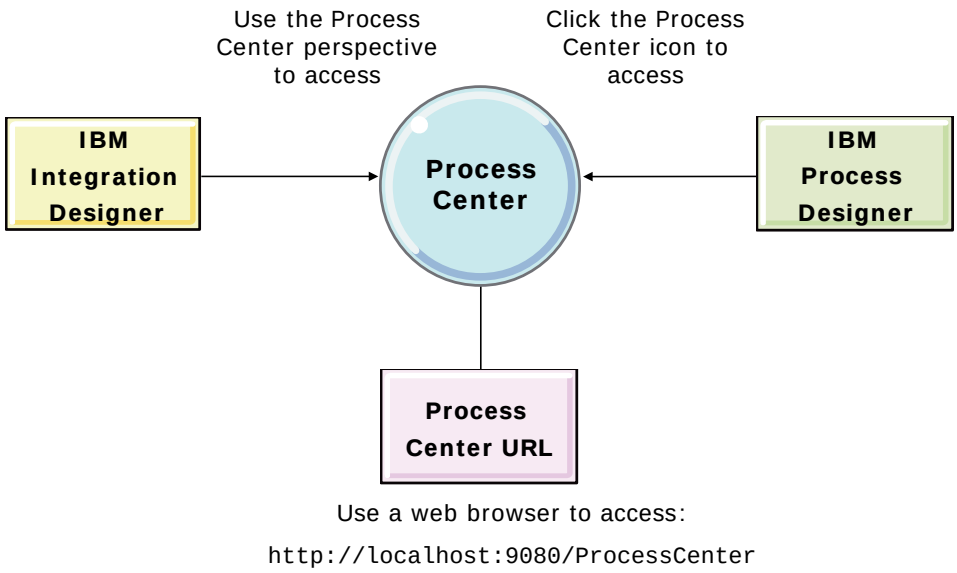
The IBM Process Center includes a repository for all processes, services, and other assets. The IBM Process Center Console provides the tools that you need for maintaining the repository.

From the Process Center console:

- You can create process applications and toolkits and grant other users access to those process applications and toolkits.
- Administrators install process applications that are ready for testing or production on the IBM Process Servers in those environments.
- Administrators manage running instances of process applications in configured environments.

IBM Training

Accessing IBM Process Center



IBM Process Center

Accessing IBM Process Center

IBM Process Center can be accessed in several ways:

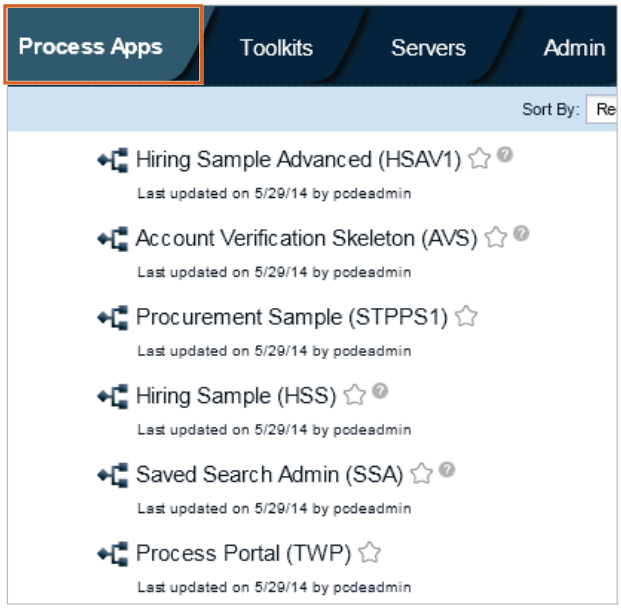
- Switching to the Process Center perspective in the IBM Integration Designer
- Clicking the IBM Process Center icon at the upper right in the IBM Process Designer
- Using a web browser at the default URL: `http://localhost:9080/ProcessCenter`

The Process Center view and capability can vary slightly, depending which tool you are using. The tool dictates the capability, which depends on its function.

IBM Training

IBM Process Center: Process Apps

- A process application is the container for a solution
- Initially created through the Process Center console
- It is given a name and a tag that is called an *acronym*
- The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages



IBM Process Center

IBM Process Center: Process Apps

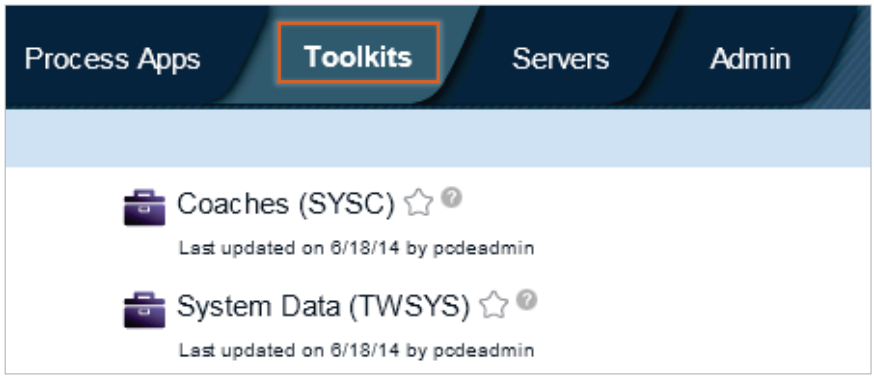
A process application is the container for a solution. You can loosely think of it as a project. The process application is initially created through the Process Center console. It is given a name and a tag that is called an acronym. The acronym must be unique and can be no more than 7 characters in length. When the process application container is created, artifacts can then be further created within it using the IBM Process Designer.

The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages. The main Process Apps page has a button to create a process application.

IBM Training

IBM Process Center: Toolkits

- Container for artifacts that are used in solutions
  - Does not result in a deployable application
- Can be “included” or “used” by one or more process applications
  - Similar to a library with artifacts
- Can be added as a dependency to a process application



IBM Process Center

IBM Process Center: Toolkits

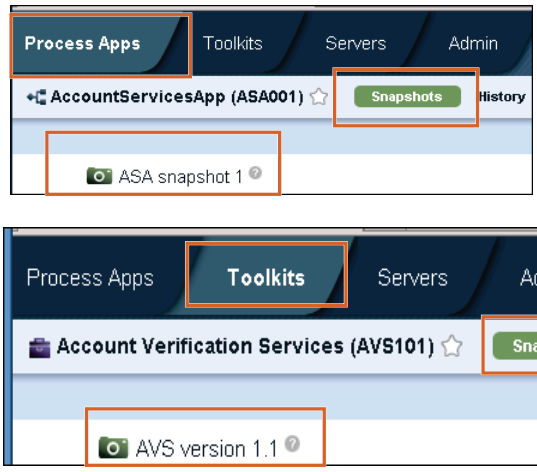
Similar to process applications, a toolkit can also be thought of as a container for artifacts that are used in solutions. Unlike a process application, a toolkit does not result in a deployable application. Instead, the contents of the toolkit can be “included” or “used” by one or more process applications.

When Process Center is installed and configured, an IBM supplied toolkit that is called “System Data” is automatically imported into the repository. This toolkit is marked as read-only and is implicitly dependent on all other process applications and toolkits. It is the System Data toolkit that contains the core definitions for data structures and other items that are common across all process applications.

Toolkits have their own tabs in the Process Center consoles. From that point, new toolkits can be created or exported, and otherwise managed in a similar fashion to the ones of the process applications.

IBM Training

IBM Process Center: Snapshots



- A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made
- Allows users to revert in time to the state of the snapshot
- Allows for creating a version of toolkits and process applications

IBM Process Center

IBM Process Center: Snapshots

A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made. The purpose of taking a snapshot is so you can revert in time to the state of the snapshot that is needed. A snapshot can be captured by clicking the **Snapshot** icon in IBM Process Designer.

A snapshot is required in some circumstances such as:

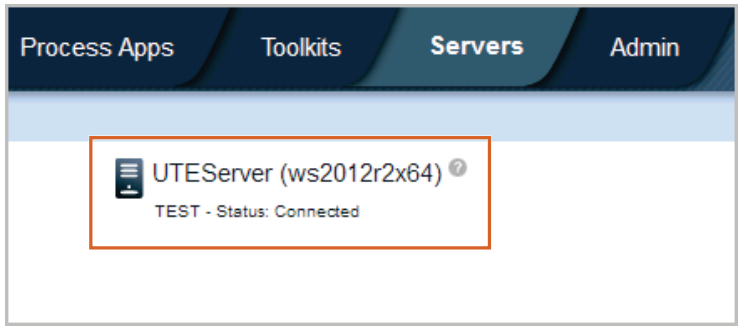
- A snapshot of a toolkit is required before it can be added as a dependency on other toolkits or process applications.
- A snapshot of a process application is required before that application can be installed on IBM Process Server.
- A snapshot is required before a workspace can be created.

Just like process applications, toolkits can have snapshots that are taken of them, allowing all the artifacts in a toolkit to have versions.

To add a toolkit as a dependency to a process application, the toolkit must first have a snapshot that is associated with it. This requirement is because the dependency added to the process application is **not** just the name of the toolkit, but is instead a specific snapshot of that toolkit.

IBM Training

IBM Process Center: Servers



- The Servers tab lists the IBM Process Servers that are connected to the IBM Process Center
  - IBM Process Server can be a stand-alone server
  - IBM Process Server can be a server that is running inside the IBM Integration Designer test environment
  - Multiple servers can be connected
  - Multiple environments can be connected: development, testing, staging, and production

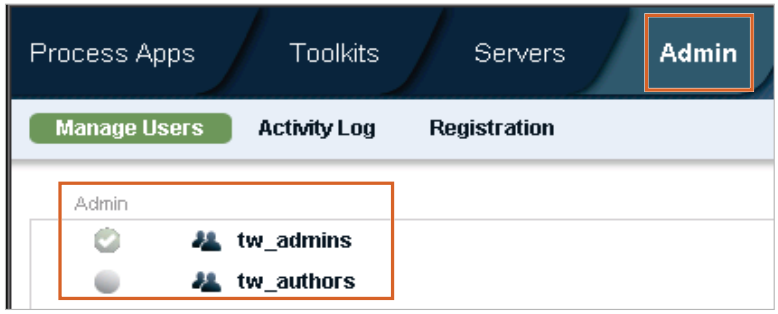
IBM Process Center

IBM Process Center: Servers

The servers that are shown on the Servers tab are the IBM Process Servers that are connected to the IBM Process Center. Authorized users can install snapshots of process applications on connected IBM Process Servers. For each server, you can view the snapshots that are currently installed.

IBM Training

IBM Process Center: Admin



- Add new users and groups to the list of authorized users
  - Granting users authority to access the repository allows them to log in
- Process Apps and Toolkits are individually controlled with their own access control

IBM Process Center

IBM Process Center: Admin

Granting users authority to access the repository allows them to log in to the IBM Process Center console. However, this access does not give privileges to work on or even see all the process applications in the environment. Process applications and toolkits are individually controlled with their own access control lists. From within the Process Apps section of the IBM Process Designer or the IBM Process Center console, an application can be selected and the **Manage** tab clicked. On that tab, it contains a section with which users and groups can be associated. These groups define the permissions for those entities.

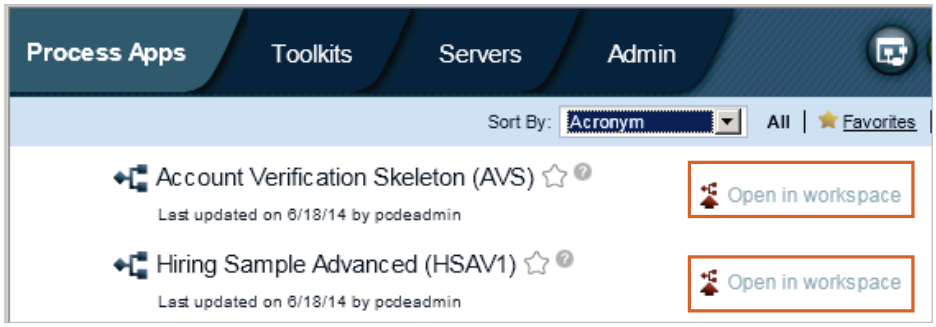
Three roles that a user or group can have are provided here:

- **Read**: This role allows a user or group to see the project and see the artifacts within it. The read role cannot be removed without removing the user or group association completely. If a user or group is not associated with a process application, then the user or group has no authorities on that application. If an artifact is opened and the user has read authority only, the artifact is flagged as read-only in the editor.
- **Write**: This role allows the user or group to update or add artifacts into the process application.
- **Admin**: This role allows the user or group to administer the process application.

Manage, test, and deploy IBM BPM applications

IBM Training

Importing from the IBM Process Center repository



- Import process applications and toolkits into your workspace from the Process Center repository
  - Can now use them with your modules and libraries
- Must be in the Process Center perspective to import the process application or toolkit into the workspace
- View and work with the newly imported workspace in the Business Integration perspective

IBM Process Center

Importing from the IBM Process Center repository

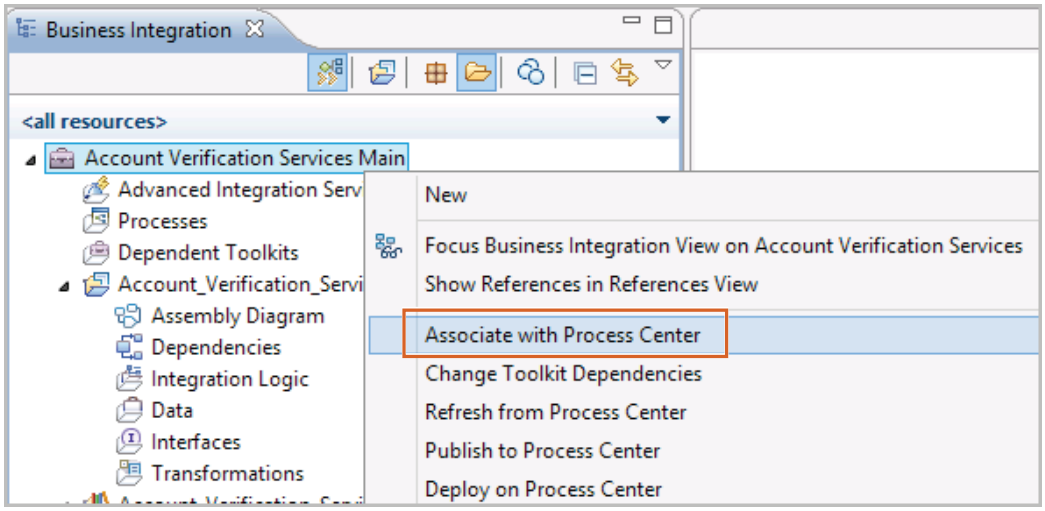
You can import process applications and toolkits into your workspace from the IBM Process Center repository, and then you can use them with your modules and libraries.

To open the process application into a workspace in the Integration Designer, you must use the IBM Integration Designer to access the IBM Process Center.

IBM Training

Associating a module or library

- Associate a module or library with a process application or toolkit to:
  - Add more functions to the application
  - Take advantage of version control on the Process Center



IBM Process Center

Associating a module or library

You can associate a module or library with a process application or toolkit to add more functions to the application, or to take advantage of version control on the IBM Process Center.

Suppose a module that contains a long-running process is deployed to a process server and you now want to associate the module with a process application. In this case, you must first consider whether you want to migrate your process instance:

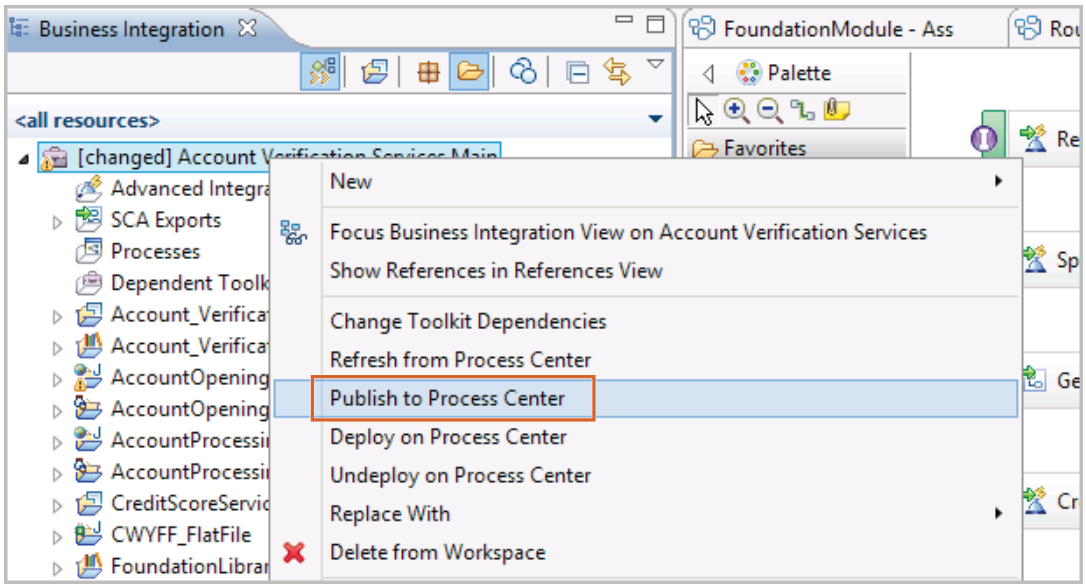
- If you want to migrate the process instance, you must create a process version before you associate the module with the process application. You can create a process version by right-clicking your module in the Business Integration view and clicking **New Process Version**.
- If you do not want to migrate the process instance, you can proceed to associate the module with the process application.

Associating a module or library with a process application or toolkit has wider implications than you might first see. Modules and libraries that are associated with toolkits can be shared with other process applications in addition to the one in your workspace. Modules and libraries that are associated with process applications are also visible within the process application. Remember too that when you bring a process application or toolkit into your workspace, you might be bringing in a snapshot from a previous point in time.

IBM Training

Publish and synchronize

- The publish command updates the IBM Process Center with the changes in your workspace



IBM Process Center

Publish and synchronize

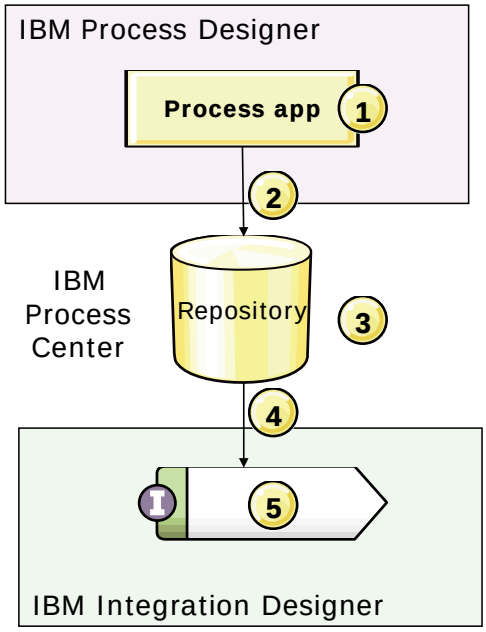
When you update your process applications and toolkits, you must update the corresponding process applications and toolkits in the IBM Process Center with your changes. The publish command updates the IBM Process Center with the changes in your workspace.

The process application or toolkit is then updated. If a conflict exists because an element in your workspace and an element in the IBM Process Center are identical, then you are warned if you want to proceed. If you proceed, the element in the IBM Process Center is overridden.

While you are updating an artifact, another user might be updating the same artifact at the same time. To make sure that you are using the current version of an artifact, you must synchronize the versions that exist in the workspace and the Process Center repository. When you try to publish the artifact to the IBM Process Center, synchronization automatically starts to merge changes into the workspace. If a conflict exists, the conflict shows in the synchronization dialog box. You must resolve the conflict before you can publish your changes to the IBM Process Center. To make sure that you are using the current version of an artifact, you must synchronize the versions that exist in the workspace and the IBM Process Center repository. Select the file that contains the changes you want to keep, and then click **Commit** to publish your changes to the repository.

IBM Training

IBM Process Designer artifacts in IBM Integration Designer



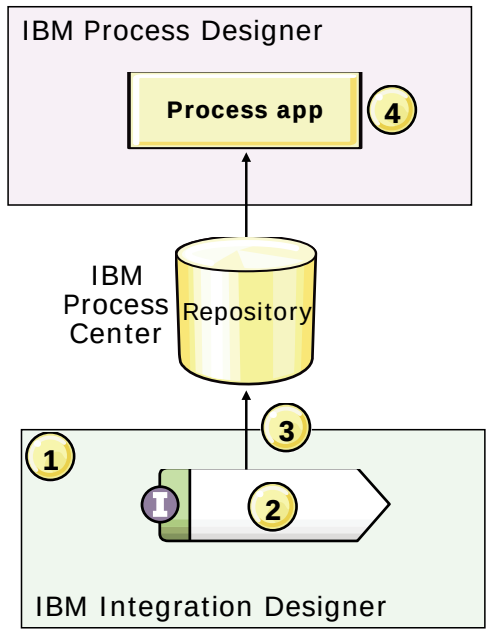
1. Assets are built in IBM Process Designer
2. A snapshot of an asset is stored in IBM Process Center
3. Developer uses the Process Center perspective in IBM Integration Designer to read a repository
4. Click **Open in Workspace**
   - Assets are added to synchronized project
5. Use business processes as export or import components

IBM Process Designer artifacts in IBM Integration Designer

IBM Training

IBM Integration Designer artifacts in IBM Process Designer

- Modules that are associated with a process app or toolkit
- “Make operations visible to IBM Process Designer” on import or export component
- Changes published to repository
- Open updated process app or toolkit
  - Imports and exports are displayed as “Advanced Integration service” implementations
  - Supporting artifacts, such as business objects, are read-only

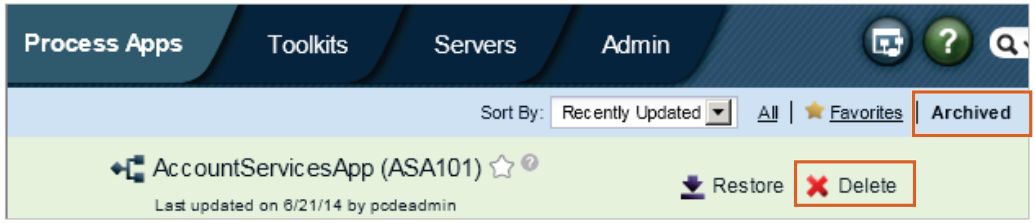


IBM Process Center

IBM Integration Designer artifacts in IBM Process Designer

IBM Training

Manage



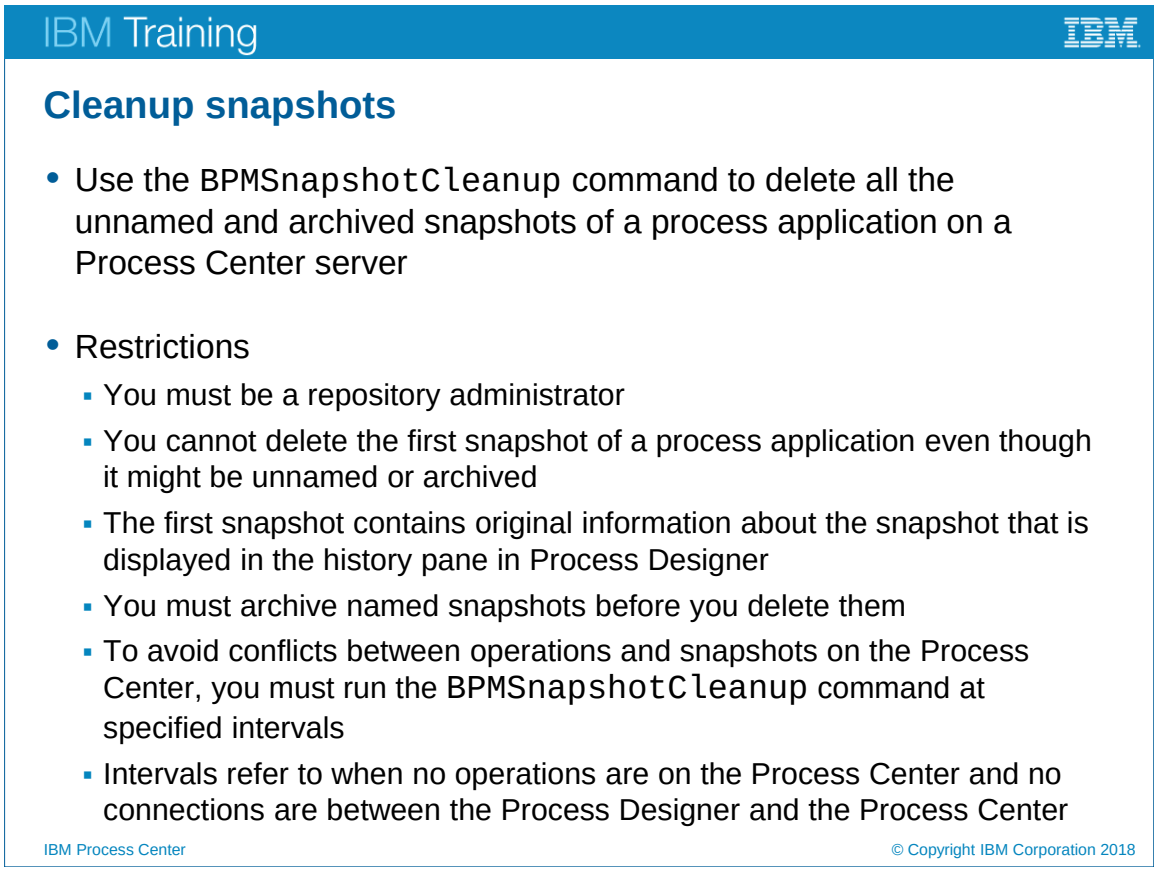
- Removing process applications from the Process Center repository
  - Use the Process Designer to first archive the process application and then delete it

IBM Process Center

Manage

To remove the process application, follow these steps:

1. From the Process Apps view of the Process Designer, click the process application that you want to remove.
2. Click **Manage**.
3. From the Manage page, click **Archive Process App**.
4. Click **Process Apps** to return to the list of process applications.
5. Click **Archived** to display a list of archived processes.
6. Click the **X** icon next to the process application you want to remove. When you are prompted, confirm the action.



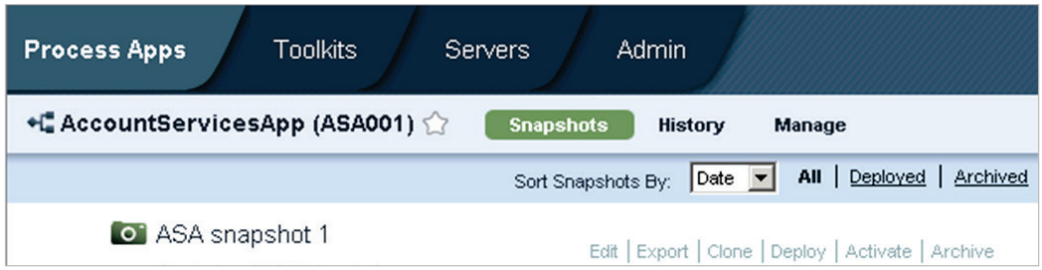
Cleanup snapshots

You can delete unnamed snapshots and archived snapshots with the same `BPMSnapshotCleanup` command by using different parameters. You can also delete snapshots in batches for better performance.

For details about command parameters and examples for Jython and Jacl, see the BPMSnapshotCleanup command in the product documentation.

IBM Training

Deploy



- Snapshots can be deployed to IBM Process Servers connected to the IBM Process Center
- Both the IBM Process Center server and the IBM Process Server where the application is being deployed must be running
- When deployed, dependencies are deployed as well

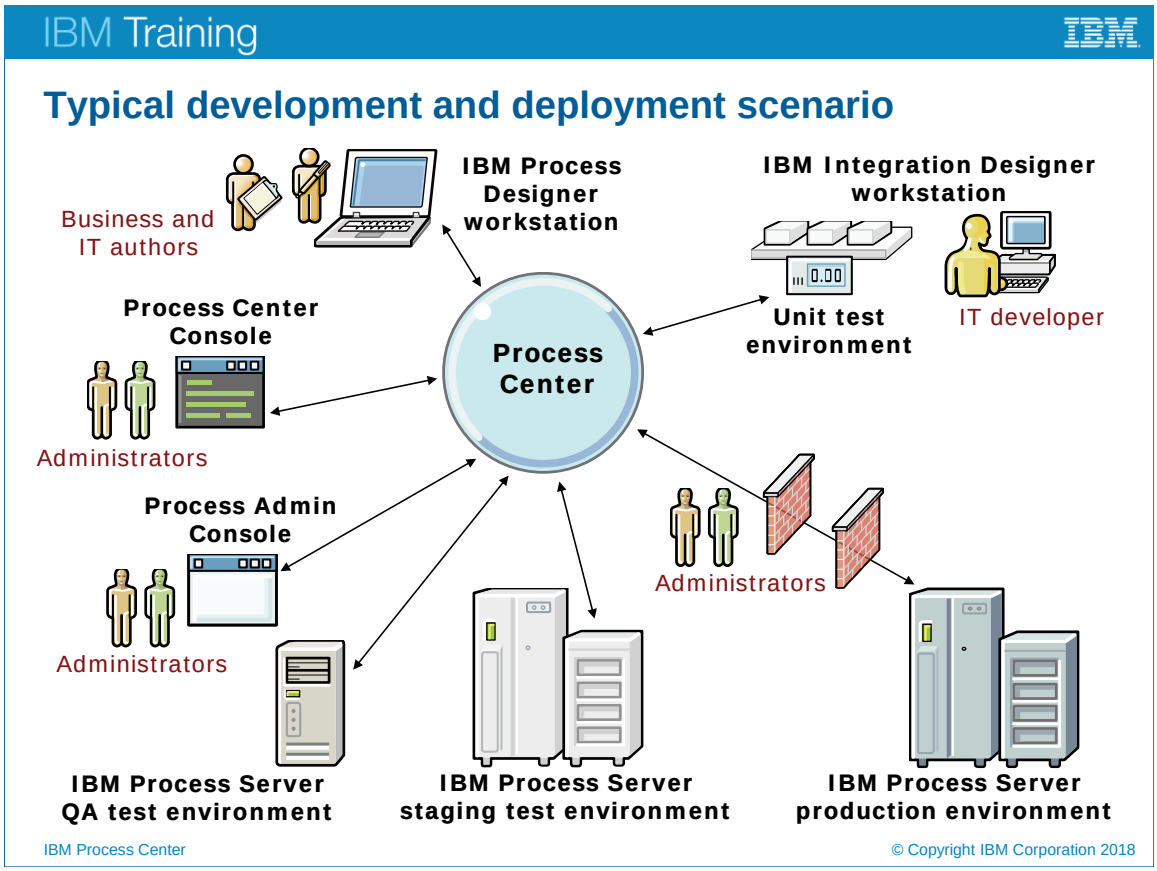
IBM Process Center

Deploy

Snapshots can be deployed to IBM Process Servers that are connected to the IBM Process Center.

Both the IBM Process Center server and the IBM Process Server where the application is being deployed must be running.

When applications are deployed, dependencies are also deployed.



Typical development and deployment scenario

The diagram displays a typical development and deployment topology that you saw in the first unit. Now that you are already familiar with the course lab environment after working on exercises, it helps to reinforce some concepts.

A test process server is installed with IBM Integration Designer in a unit test environment (UTE). In this mode, the unit tester can test SCA modules locally in the IBM Process Server that is running inside the UTE or test them directly on the Process Center. Either way is acceptable, depending on the requirements.

The IBM Process Designer workstation communicates directly with the Process Center Console in the center. Connected to the central IBM Process Center are independent QA, staging, and production environments, each with its own full stand-alone IBM Process Server. Artifacts are being published and synchronized back and forth between all the environments while the IBM Process Center is managing the central repository.

This scenario is just a sample, and the topology can be modified to suit the organization requirements.

IBM Training

Unit summary

- Describe the purpose and business value of IBM Process Center
- Define the components of IBM Process Center
- Describe how to use the IBM Process Center Console to deploy, test, and manage IBM BPM applications

Unit summary

IBM Training

Checkpoint questions

1. True or False: The IBM Process Center console can be accessed in only two ways; the first is through IBM Process Designer, and the second is through a web browser by using the Process Center URL.

2. True or False: Snapshots must be created before you can deploy a process application.

3. True or False: You can import both the toolkit and the process application inside an IBM Integration Designer workspace.

Checkpoint questions

IBM Training

Checkpoint answers

1. False. The third way is to use the IBM Integration Designer perspective.

2. True.

3. True. You import both the toolkit and the process application through the IBM Process Center repository.

Checkpoint answers

IBM Training

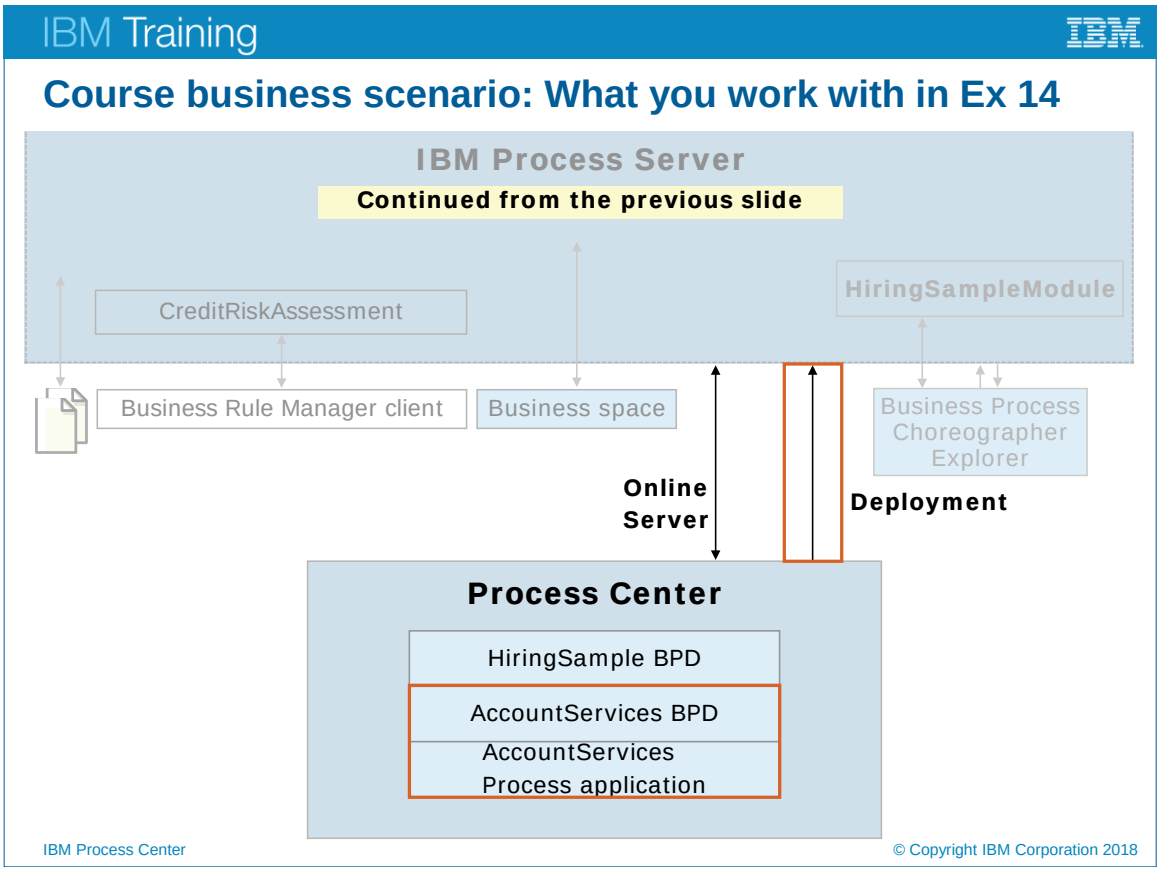
Exercise 14: Exploring IBM Process Center

After completing this exercise, you should be able to:

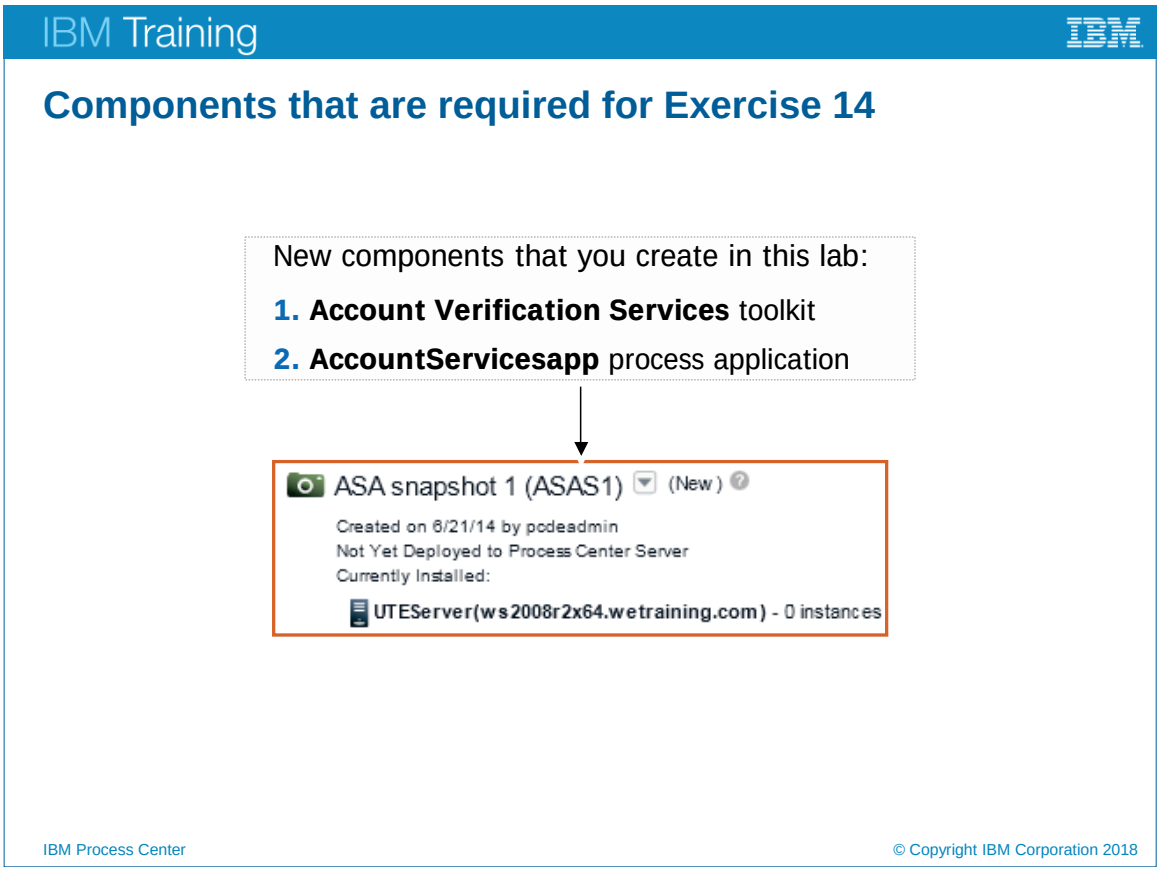
- Explore the IBM Process Center repository
- Create a toolkit in IBM Process Center
- Associate IBM Integration Designer artifacts with the toolkit
- Generate access to process applications and toolkits
- Archive and delete process applications

IBM Process Center

Exercise 14: Exploring the IBM Process Center



Course business scenario: What you work with in Ex 14



Components that are required for Exercise 14

If applicationDecision is set to false during FinalApplicationReview (the application is declined) and the customer's creditRisk is HIGH, the application is routed through the "generate decline" component. If applicationDecision is set to false during FinalApplicationReview and the customer's creditRisk is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the RouteRequest component. The RouteRequest flow component contains the mediation logic that routes the application to the appropriate decline service.

The RouteRequest mediation flow consists of both a request flow and a response flow. In the flow, the CustomerApplication is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the AccountVerification process.

Exercise 14: Exploring IBM Process Center

Purpose:
In this exercise, you use the IBM Process Center repository to manage your applications.

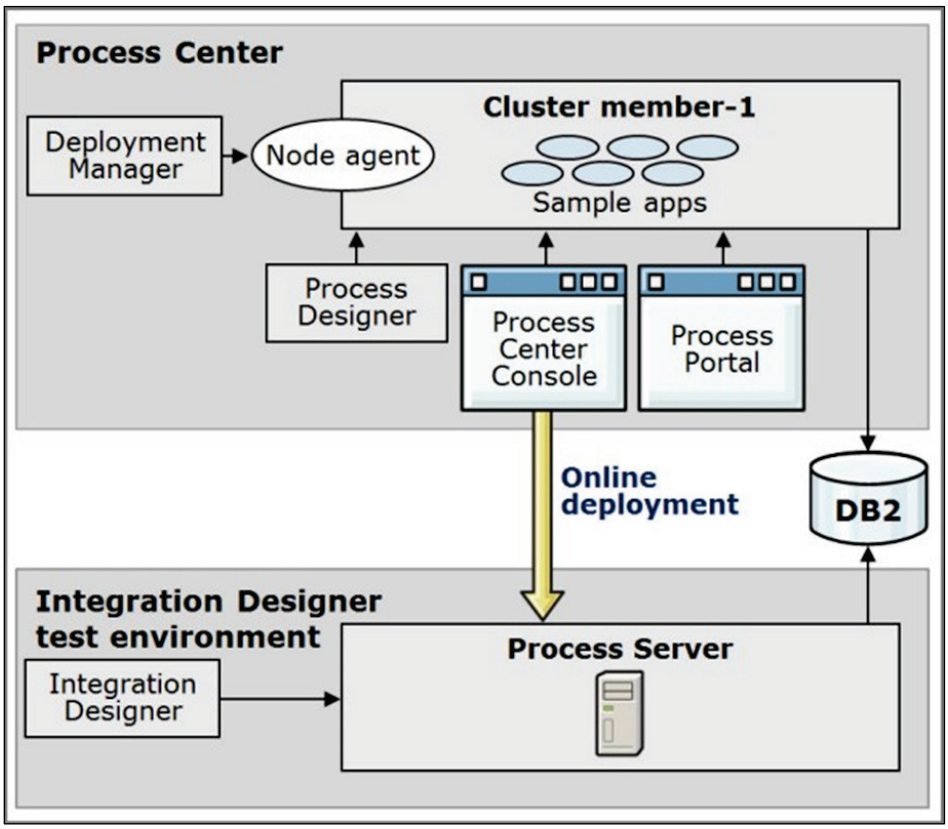
IBM Process Center is a runtime environment where IBM Process Designer and IBM Integration Designer share assets. It allows the development of business processes cooperatively in a highly interactive manner. IBM Process Center includes a server and performance data warehouse. IT employees who work in the authoring environments run processes and store performance data for testing and playback purposes.

The IBM Process Center console provides a convenient location in which to create and maintain high-level containers such as process applications and toolkits. Administrators can use the IBM Process Center console to provide a framework in which BPM analysts and developers can build their processes and underlying implementations. Another primary task for administrators is managing access to the IBM Process Center repository by setting up the appropriate authorization for users and groups.

From the Process Center console, you can:

- Create process applications and toolkits and grant other users access to those process applications and toolkits
- Create process models, services, and other assets within process applications
- Install process applications that are ready for testing or production on the process servers in those environments
- Manage running instances of process applications in configured environments

Process Center is installed and configured as a single cluster topology, whereas Process Server is installed as a test server (as a stand-alone single server) while installing integration designer. The following diagram depicts different processes that you start and their relationship with Process Center and Process server Cells.



Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Center, and IBM Process Server test environment.

Part 1. Start the Process Center environment.

One way of accessing the IBM Process Center Console is through a perspective within the IBM Integration Designer.

To start the IBM Process Center server:

1. Start the deployment manager.

1. On your Windows desktop in your lab environment, select the **Start Process Center deployment manager** shortcut. Double-click the shortcut or press Enter to start the server.

When the deployment manager starts, you are prompted to press any key to continue.

2. Press any key to close the command window.

2. Start the node agent.

1. On your Windows desktop, select the **Start Process Center node agent** shortcut.
2. Double-click the shortcut or press Enter to start the server.

3. A DOS command window is displayed; press any key to continue when prompted.

3. Start the process center single cluster.

1. On your Windows desktop, select the **Start Process Center Cluster** shortcut.

2. Double-click the shortcut or press Enter to start the cluster member.

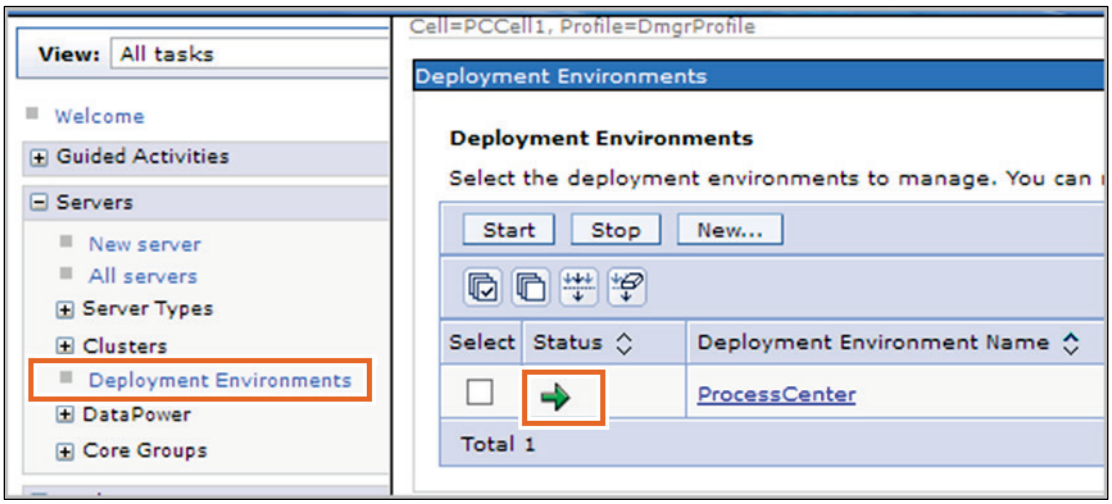
A DOS command window is displayed, and the IBM Process Center server instance starts. IBM Process Center is an application that runs in its own profile of WebSphere Application Server. That profile is connected to a DB2 repository where IBM Process Center stores its BPD artifacts.

3. Press any key to continue when prompted.

4. Verify that the Process Center Single Cluster is started.

1. Start Firefox and go to `https://ws2012r2x64:9044/ibm/console` to launch the administrative console.
2. Click **Continue to this website (not recommended)**, if prompted.
3. Log in by entering `bpmadmin` and `web1sphere` in the **User ID** and **Password** fields.
4. Select **Servers > Deployment Environments**.
5. Verify that the deployment environment is started. When it starts, the status turns green.

   It might be necessary to refresh the browser or click the **Deployment Environments** link to view the status change.



6. Click **Servers > Clusters > WebSphere application server clusters** to verify that the single cluster is running.
7. Click **Logout** to exit.
8. Close the browser.

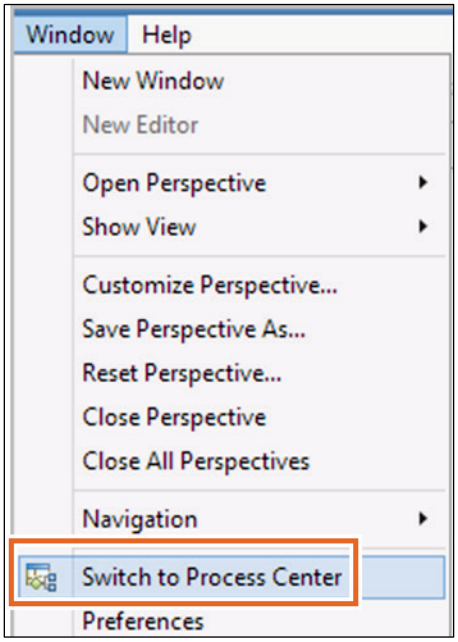
Part 2. Work with the Process Center Repository

1. Open the Exercise 14 workspace.

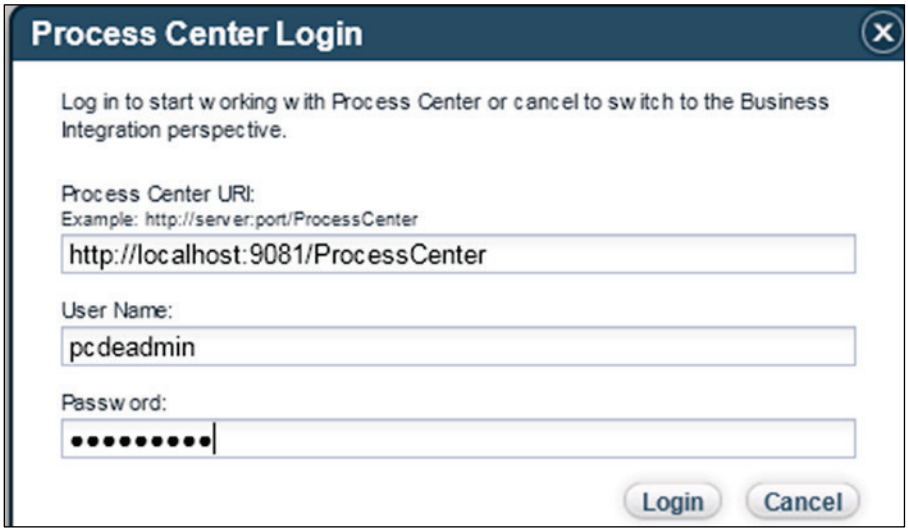
1. On your desktop, open the **Exercise Shortcuts** folder.
2. Double-click the **Exercise 14** shortcut. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.
3. When the **Workspace Migration** window is displayed, clicked **Next**.
4. Click **Finish** to complete the migration.
5. Click **OK** in the **Migration Validation** window.
6. Wait for the workspace to build.
7. Close the **Getting Started** tab and the **Migration Results** tab.

2. Switch to the Process Center perspective.

1. Click **Window > Switch to Process Center** in the menu options.



2. In the Process Center Login window, enter the following credentials:
   - **Process Center URI**: `http://localhost:9081/ProcessCenter`
   - **User Name**: `pcdeadmin`
   - **Password**: `web1sphere`



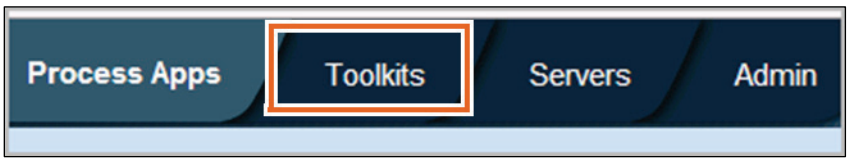
3. Click **Login**.
4. When the Secure Storage window opens, click **Cancel**.
5. If a Security Alert window is displayed asking to proceed, then click **Yes** each time.
6. When the Secure Storage Warning window opens, click **OK**.
7. Close the “Getting Started with IBM Process Center 8.6.0” welcome screen, by clicking the **X** at the upper-right corner of the window.

3. Create a Toolkit in IBM Process Center.

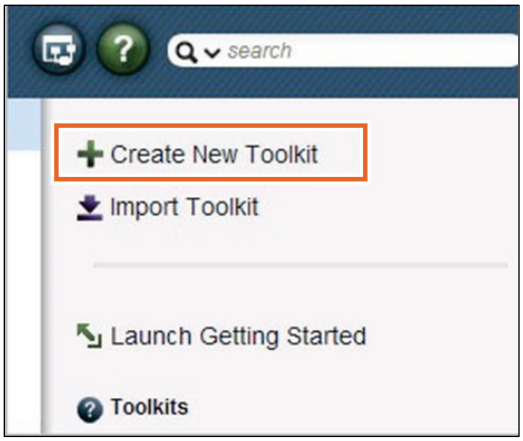
You can create toolkits to enable IBM Process Designer users to share library items across process applications. The Process Center perspective is the way to establish and maintain relationships between toolkits, process applications, and SCA services. You can do tasks such as:

- Create a process application or toolkit
- Associate SCA services with a process application or toolkit
- Import process application and toolkit artifacts into the IBM Integration Designer
- Publish process applications and toolkits to the Process Center

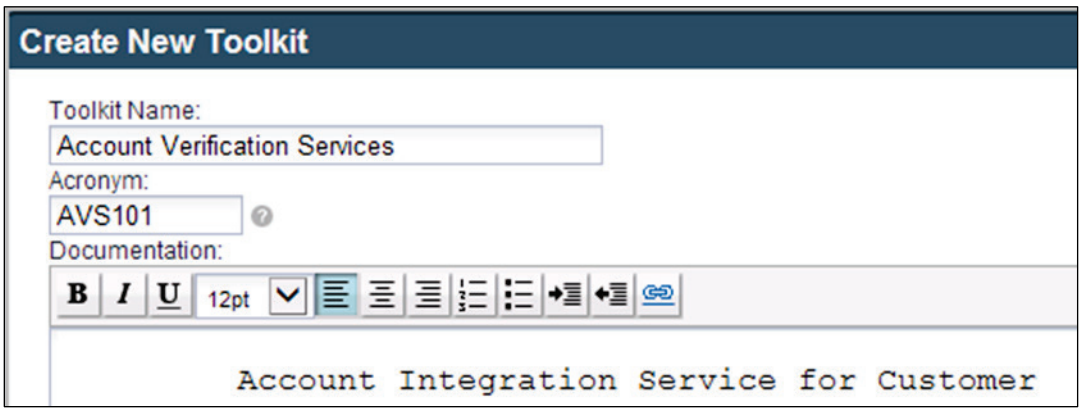
1. In the Process Center perspective, click the **Toolkits** tab.



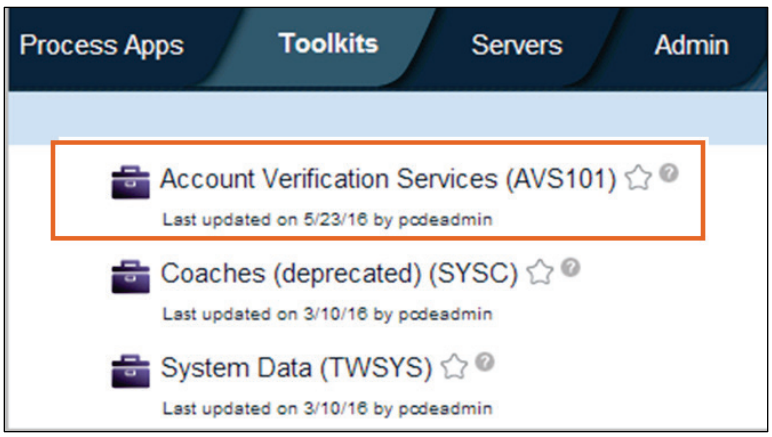
2. Verify that one toolkit is named **System Data**, which is listed under the Toolkits tab. This toolkit is the default, which is created during installation and is imported into the Process Center repository. Each process application and toolkit that you create automatically includes a System Data toolkit dependency. You have access to the assets that all IBM Business Process Manager projects require, such as standard variable types and standard charts for reports.
3. In the upper-right corner of the Process Center perspective, click **Create New Toolkit**.



4. In the Create New Toolkit window, enter the following information:
   - Enter `Account Verification Services` in the **Toolkit Name** field.
   - Enter `AVS101` in the **Acronym** field. The acronym must be unique in the repository. You can change this value in case this acronym is in use in the repository.
   - Enter `Account Integration Service for Customer` in the **Description** field.



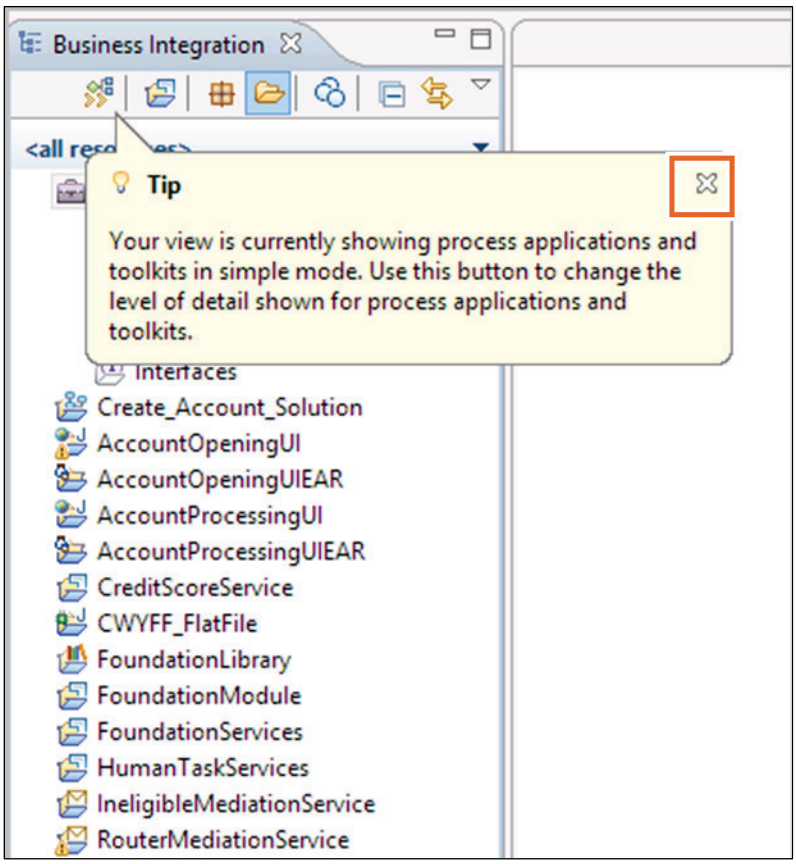
   Note the rich text formatting options available in the toolbar for the **Description** field. You can apply different font options, font sizes, alignment, and indentation to the text. Feel free to explore the different formatting options.
5. Click **Create** to create the toolkit. When it is created, the **Create New Toolkit** window closes.
6. Verify that the newly created toolkit is listed in the **Toolkit** tab.



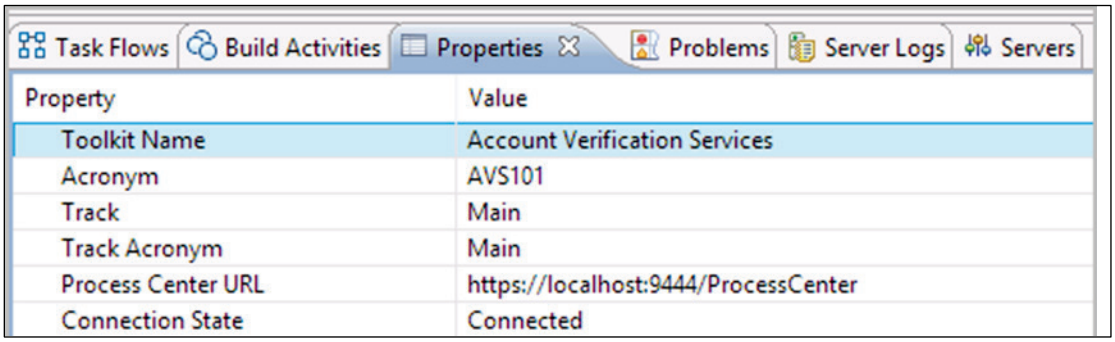
4. Associate IBM Integration Designer artifacts with the toolkit.

1. In the **Toolkits** tab, click **Open in workspace** next to the **Account Verification Services** toolkit.

2. The Business Integration perspective opens with the **Account Verification Services** project. If displayed, close the Tip window by clicking **X**. Wait for the workspace to build before going to the next step. It might take a while.



3. Select **Account Verification Services Main** and click the **Properties** tab. Examine the values.

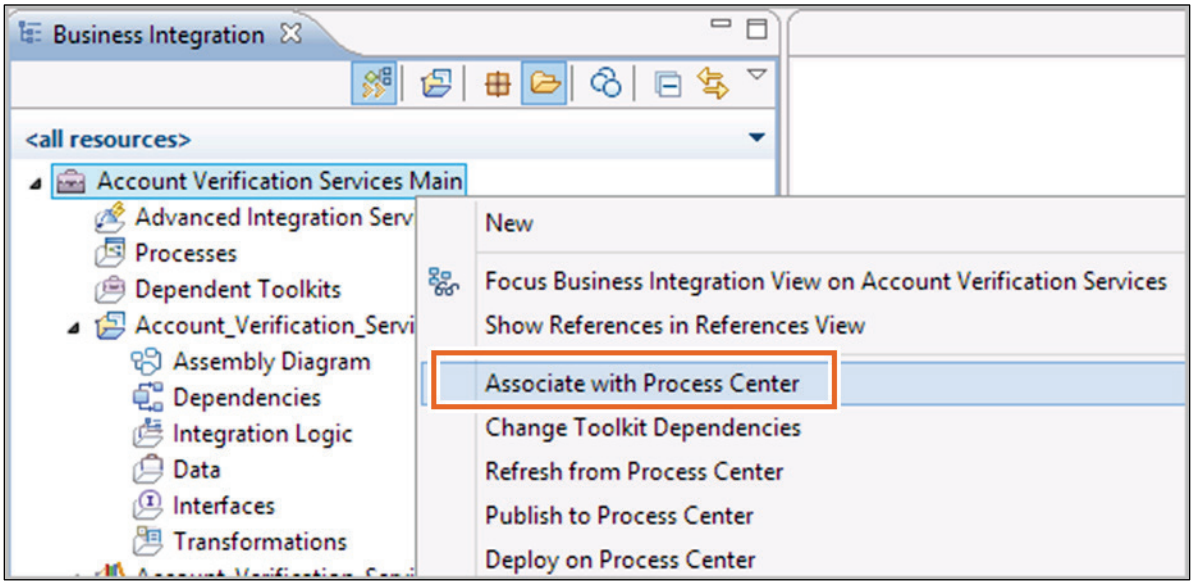


If you do not see the implementation and library projects, switch your view to Detailed mode.

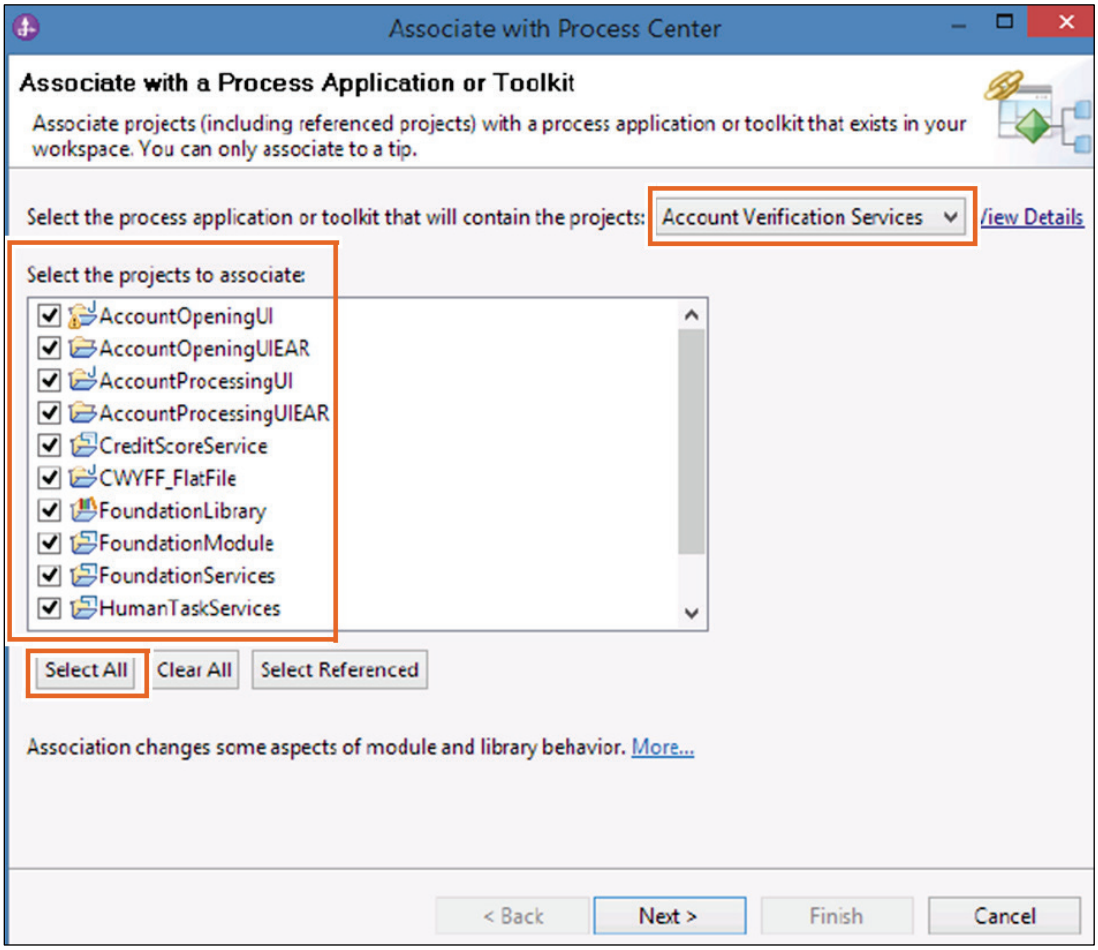
You can switch views by clicking the **Switch Process Applications and Toolkits to Detailed Mode** icon.

5. Associate the imported project with the Account Verification Services toolkit you created earlier.

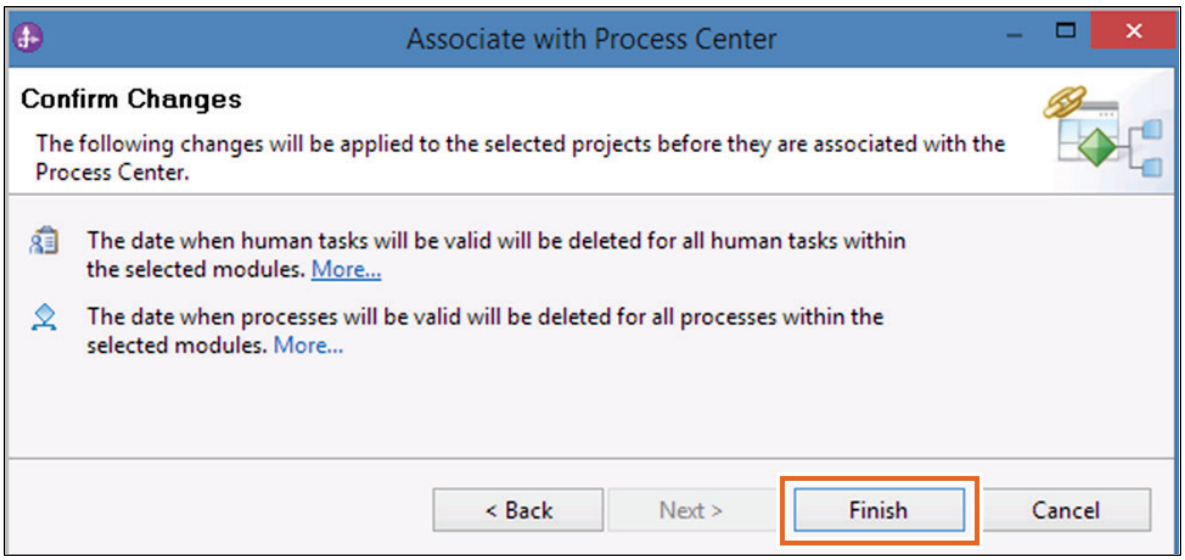
1. Right-click Account Verification Services Main, and click Associate with Process Center.



2. In the “Associate with Process Center” window, verify that **Account Verification Services** is selected as the toolkit. Click **Select All** to associate the toolkit with the other projects in IBM Integration Designer.

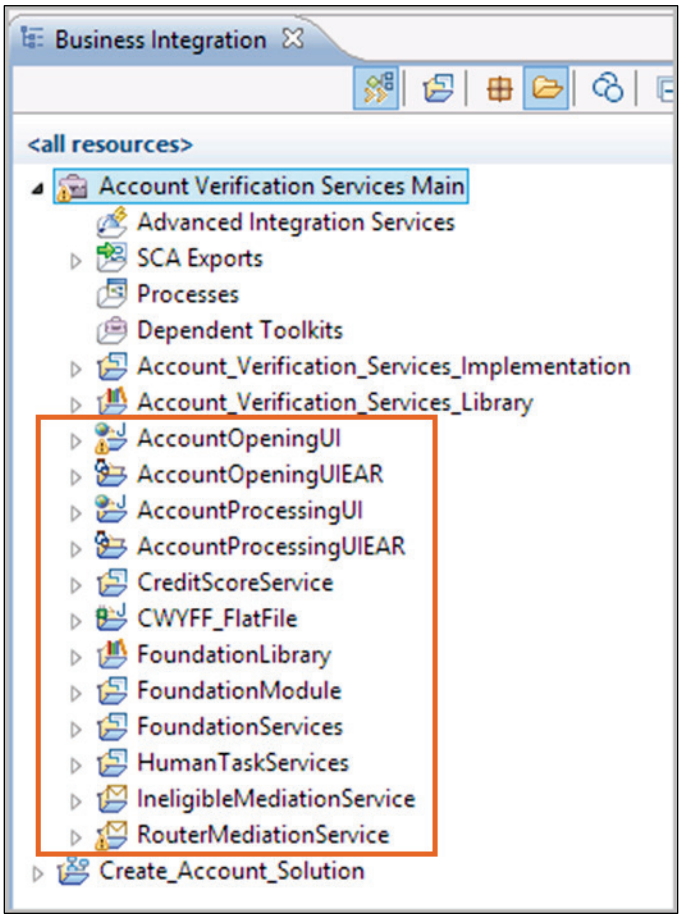


3. Click **Next**.
4. Click **Finish** in the “Confirm changes” pane to complete the association.

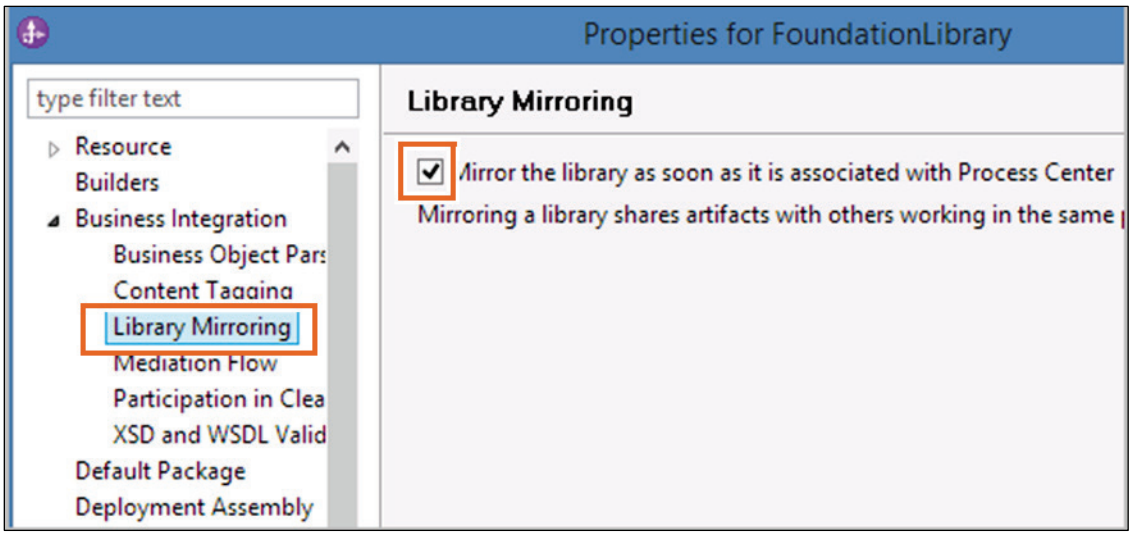


It takes a while for the association with the Process Center to take place. Watch the status at the lower right. Wait until the status reaches 100 percent, which indicates that the update is complete.

5. Examine the **Account Verification Services Main** project on the left. All the modules are moved into it.



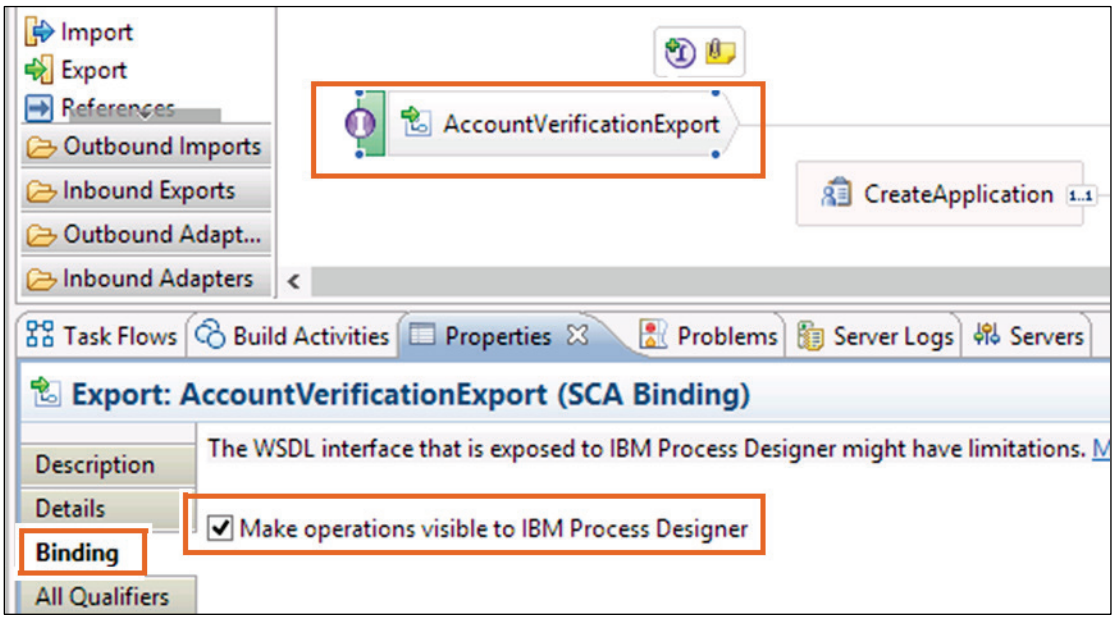
6. Right-click **FoundationLibrary** and click **Properties**.
7. In the “Properties for FoundationLibrary” pane, expand **Business Integration** and select **Library Mirroring**.
8. Select **Mirror the library as soon as it is associated with Process Center** and click **OK**.



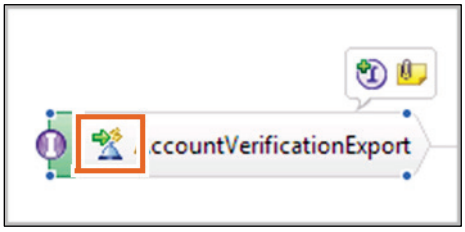
Information: In a collaborative development environment between IBM Integration Designer and IBM Process Designer, artifacts like business objects are shared in libraries. When you put an artifact in your library in Integration Designer, it is made available to others who work with the same library in Process Designer. This sharing of artifacts is called library mirroring.

6. Make the operations available in IBM Process Designer by setting the attributes in the IBM Integration Designer.

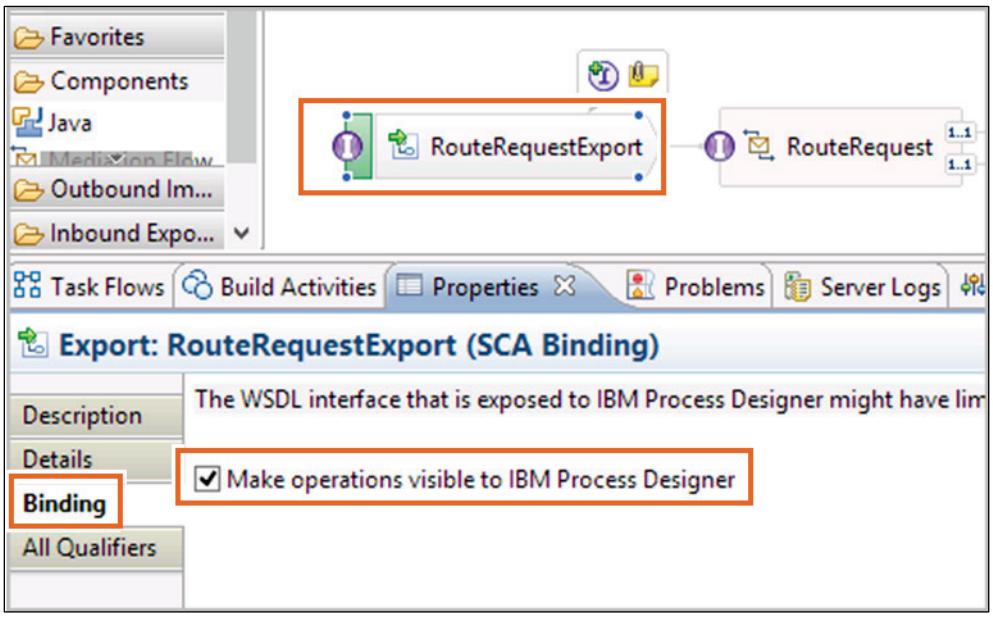
1. Expand FoundationModule and double-click Assembly Diagram.
2. In the assembly diagram, select **AccountVerificationExport** and click the **Properties** tab.
3. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.



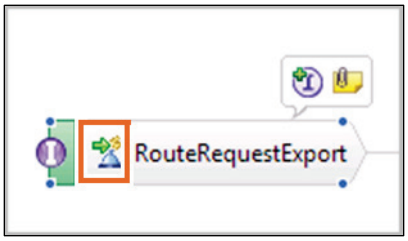
4. Click **File > Save All** to save your changes.
5. Notice the new icon for the export.



6. Expand **RouterMediationService** and double-click **Assembly Diagram**.
7. In the assembly diagram, select **RouteRequestExport** and click the **Properties** tab.
8. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.

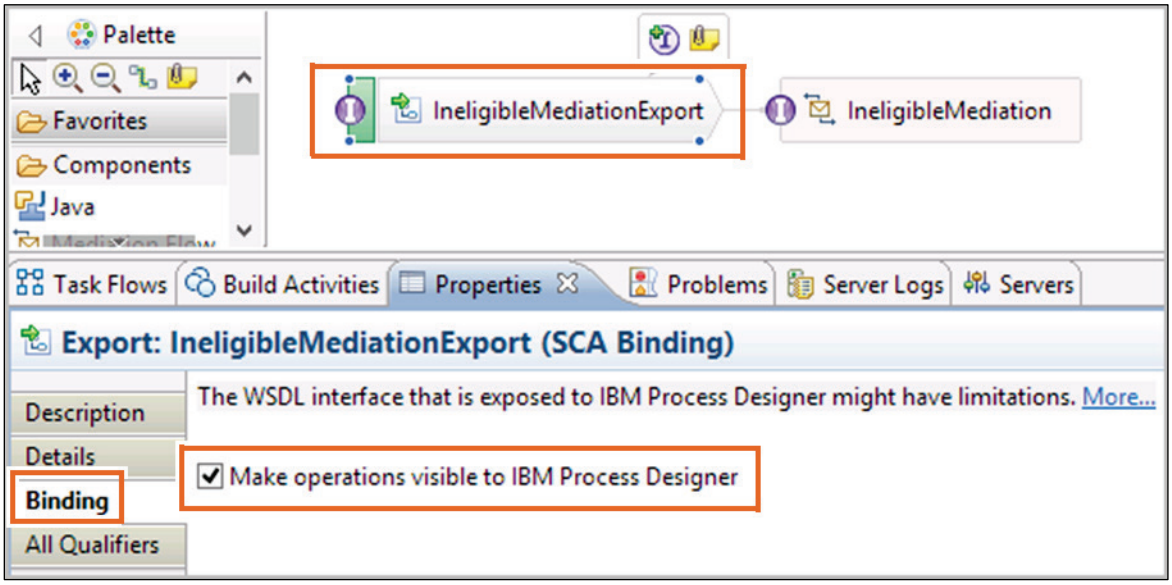


9. Click **File > Save All** to save your changes.
10. Notice the new icon for the export.



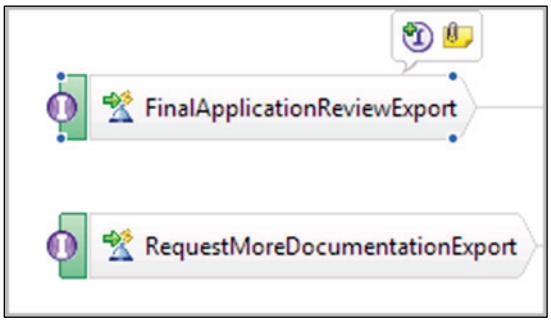
11. Expand **IneligibleMediationService** and double-click **Assembly Diagram**.
12. In the assembly diagram, select **IneligibleMediationExport** and click the **Properties** tab.

13. In the **Binding** tab, select **Make operations visible to IBM Process Designer**.



14. Click **File > Save All** to save your changes.
15. Notice the new icon for the export.

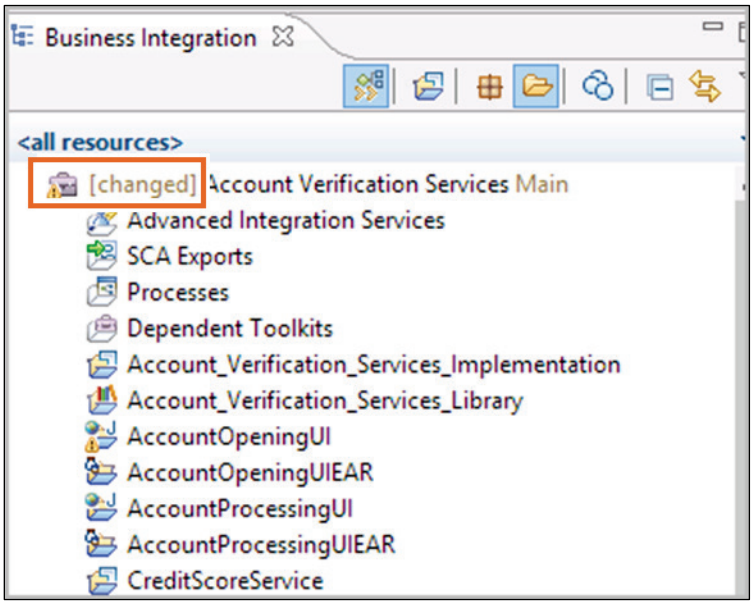
16. Expand **HumanTaskServices** and double-click **Assembly Diagram**.
17. Select the **Make operations visible to IBM Process Designer** for two export components in the assembly diagram: **RequestMoreDocumentationExport** and **FinalApplicationReviewExport**. The properties are in the **Binding** tab in the **Properties** view for the respective components.
18. Click **File > Save All** to save your changes.
19. Notice the new icon for the export.



20. Expand **FoundationServices** and double-click.

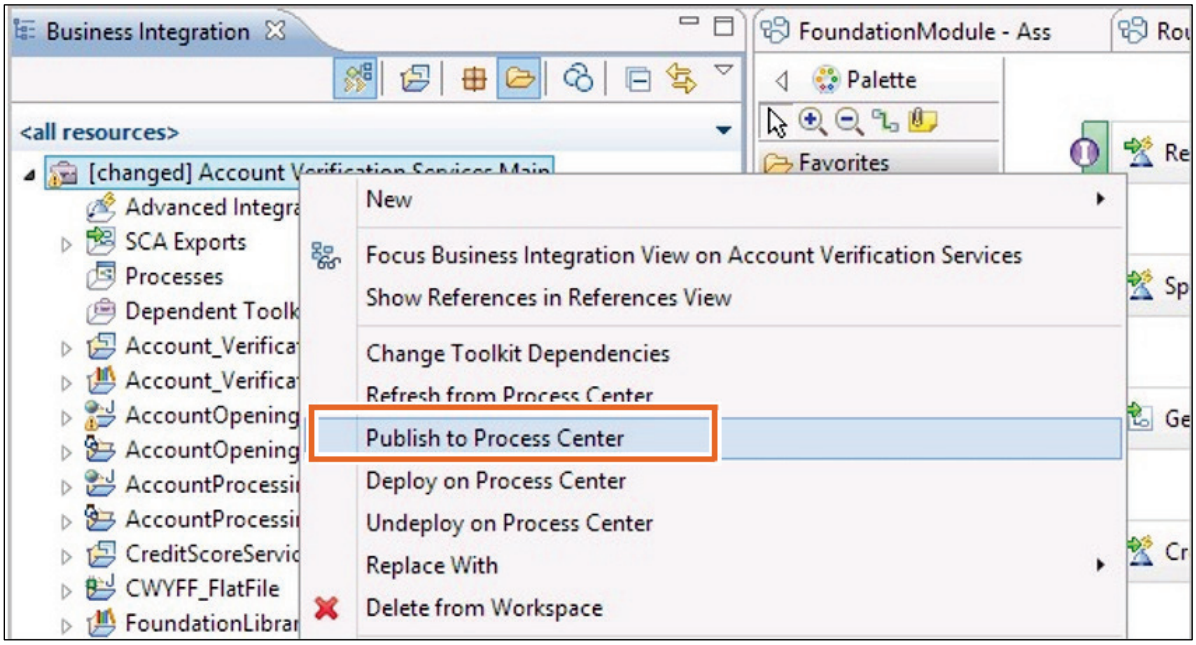
21. Select the **Make operations visible to IBM Process Designer** for three export components in the assembly diagram: **RecordIneligibleApplicationExport**, **SpecialDeclineExport**, and **CreditAssessmentExport**. The properties are in the Binding tab in the Properties view for the respective components.

    Click **File > Save All** to save your changes.
22. Notice the new icons for the three exports.
23. Examine the changed status of the **Account Verification Services Main** project. The newly added text [changed], which is to the left of the project, indicates the changed status.

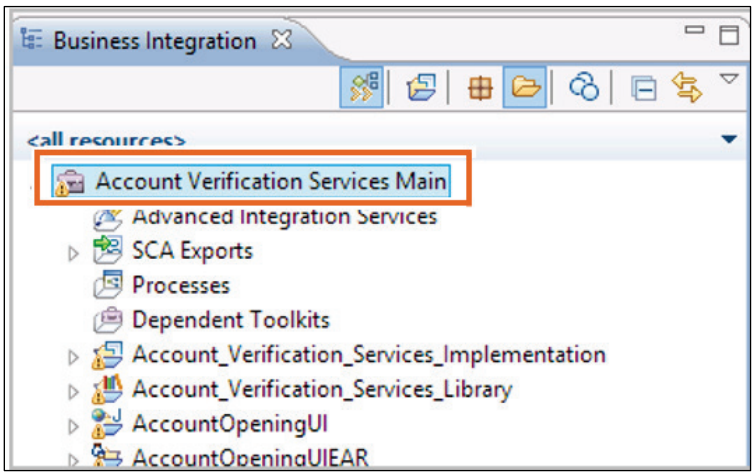


7. Publish Account Verification Services Main to the Process Center.

1. Right-click Account Verification Services Main and click Publish to Process Center.



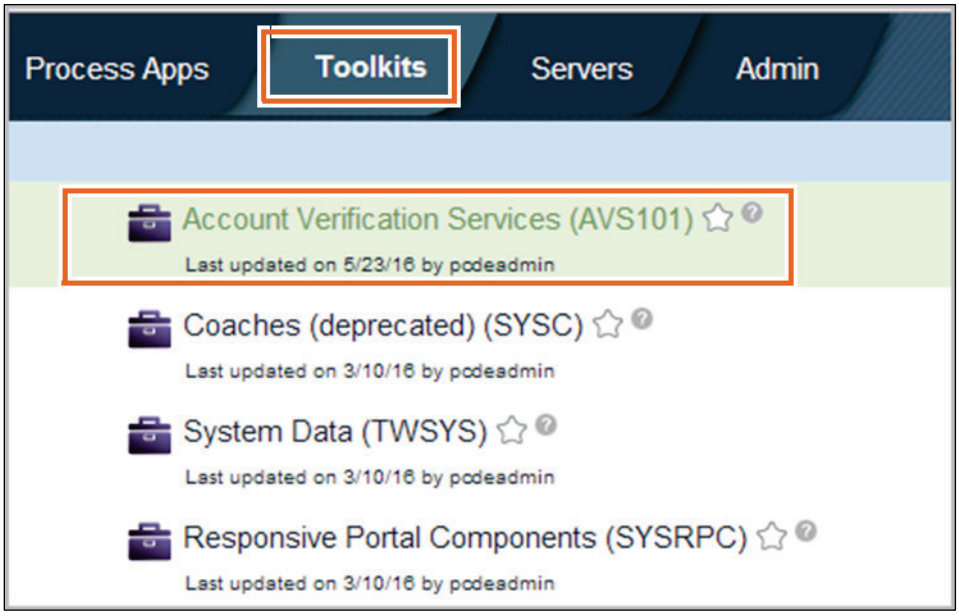
2. Click **OK** in the **Library Mirroring** window.
3. Wait for the update to complete. It takes a while for the status at the lower right to change. The update is complete when the status **changed** is no longer displayed next to **Account Verification Services Main** as it was before.



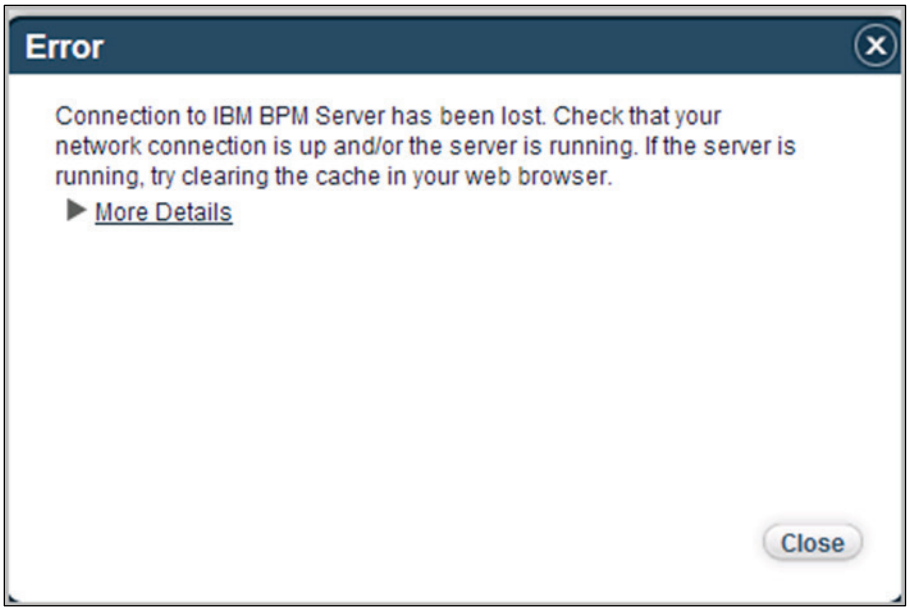
8. Create a snapshot of the published toolkit

Snapshots record the state of library items within a process application or track at a specific point in time. You can create snapshots in the Process Center console or in the IBM Integration Designer view. Snapshot management, such as deploying, exporting, and archiving, is done in the Process Center console.

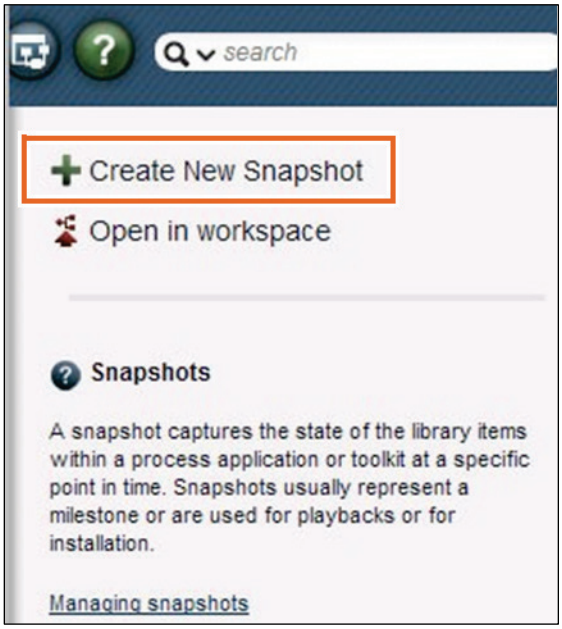
1. Switch to the Process Center perspective (click **Window > Switch to Process Center**) and verify that you are in the **Toolkits** tab. Click **Account Verification Services (AVS101)**.



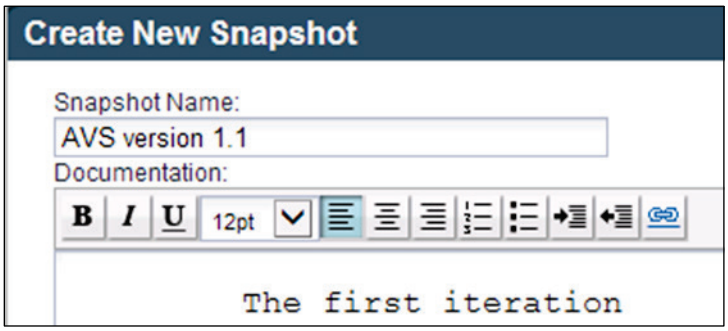
Important: At any time, while working with the Process Designer or Process Center, you might receive an IBM Business Process Manager error with a message that the connection was lost. If that occurs, do not be concerned. Click Close and continue with your lab exercise.



2. Click **Create New Snapshot** on the right.

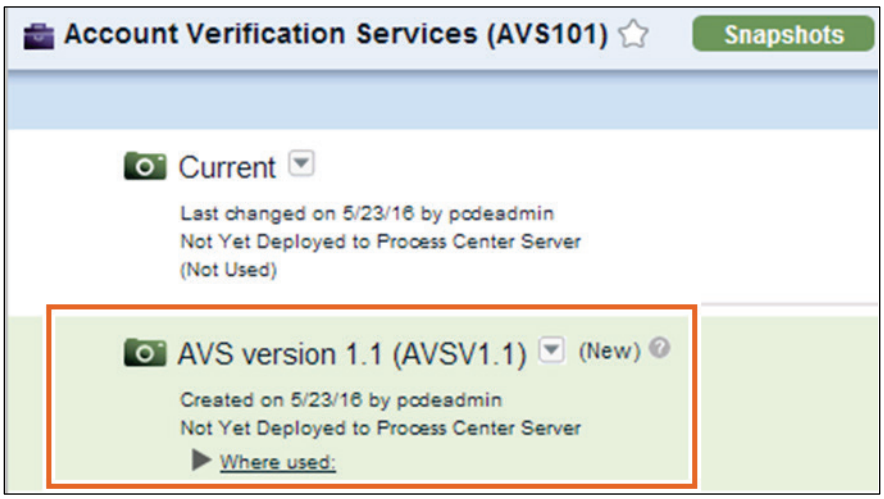


3. In the Create New Snapshot window, enter AVS `version 1.1` in the **Snapshot Name** field and enter `The first iteration` as the **Description**.



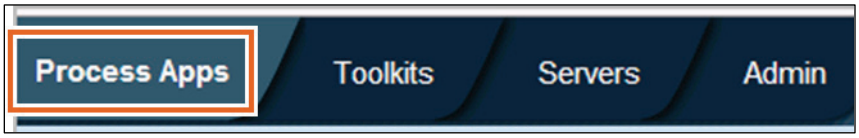
4. Click **Create**. It is not necessary to click **Create** twice. When it is created, the **Create New Snapshot** window closes.

   The toolkit is now ready to be shared among process developers (refresh might be needed).

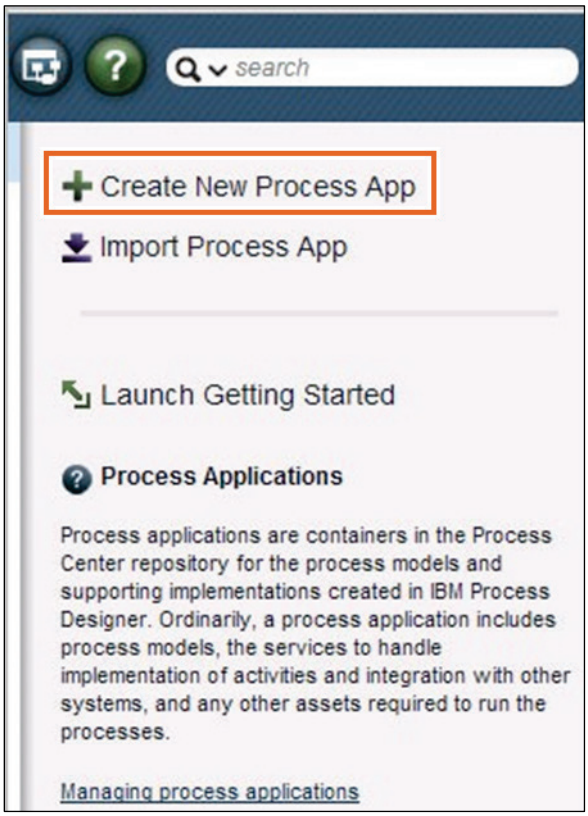


9. Generate access to process applications and toolkits.

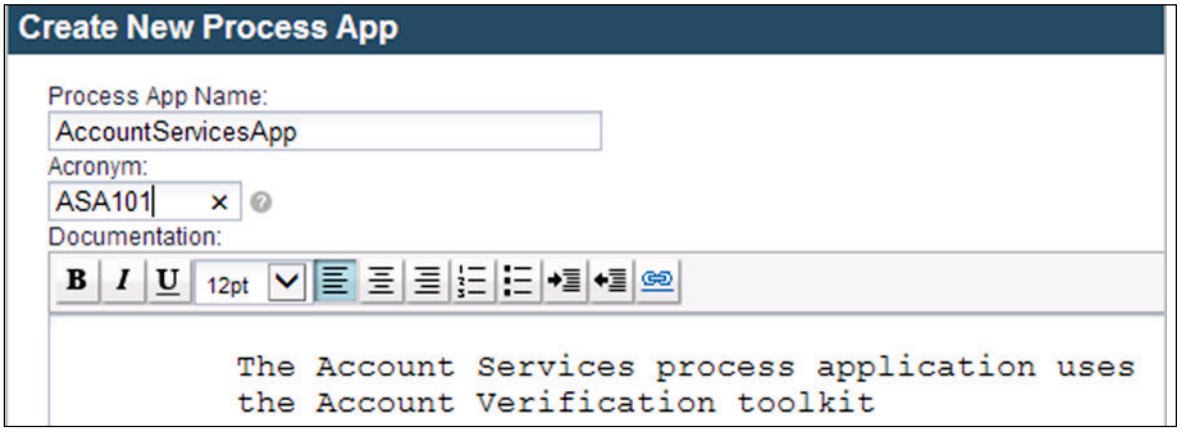
1. Click the **Process Apps** tab at the upper left of Process Center.



2. Click **Create New Process App** from the upper-right corner of the screen.

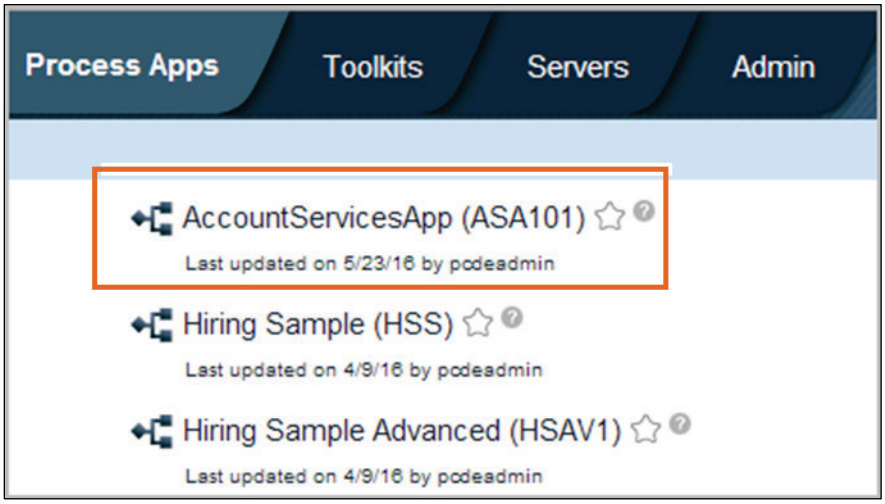


3. In the Create New Process App window, enter the following information:
   - Process App Name: `AccountServicesApp`
   - Acronym: `ASA101`
   - Description: `The Account Services process application uses the Account Verification toolkit`



Hint: The acronym must be unique in the repository. You can change this value in case this acronym is in use in the repository.

4. Click **Create**. The newly created process application is listed in the tab.

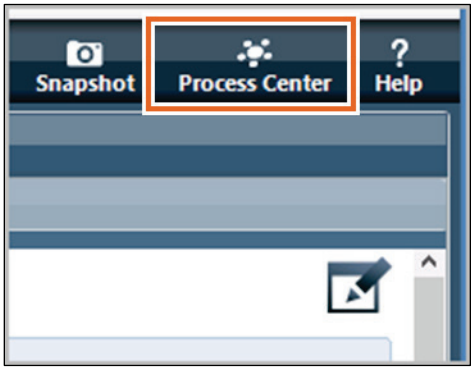


10. Set the dependency between the AccountServicesApp and the Account Verification Services toolkit.

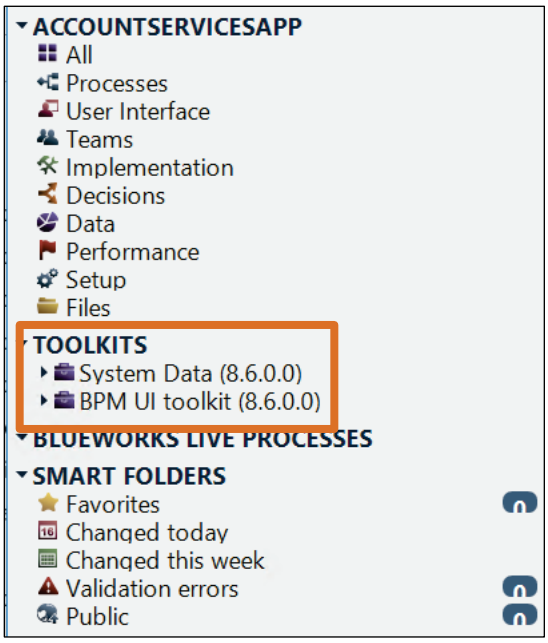
1. On the Windows desktop, locate the icon that is labeled **IBM Process Designer**.

2. Double-click the icon or press Enter to start IBM Process Designer.
3. To log in and connect to the IBM Process Center repository, enter `pcdeadmin` and `web1sphere` for the user name and password fields.
4. Click **Login**.

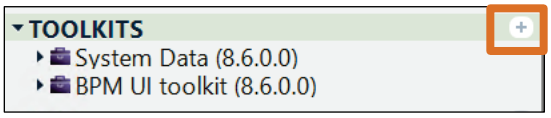
5. In the **Security Alert** dialog box, click **Yes**.
6. Click **Yes** one more time to continue. After a few moments, IBM Process Designer starts.
7. If you are not already on the Process Center page, switch to the **Process Center** tab by clicking **Process Center** at the upper right.



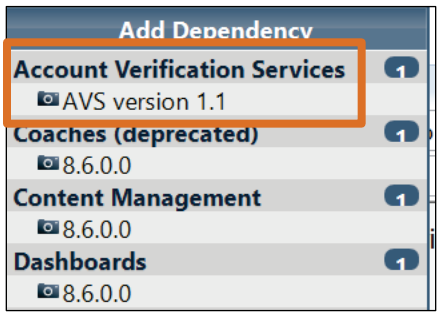
8. Click **Open in Designer** next to **AccountServicesApp (ASA101)**.
9. Expand **Toolkits**, unless it is already expanded. The default **System Data** and **BPM UI** toolkits are listed.



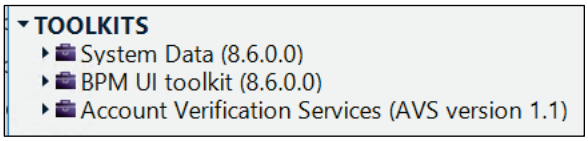
10. Click the plus sign (**+**) to the right of **Toolkits**. The plus sign (**+**) is not visible by default. You must hover over **Toolkits** to see the icon.



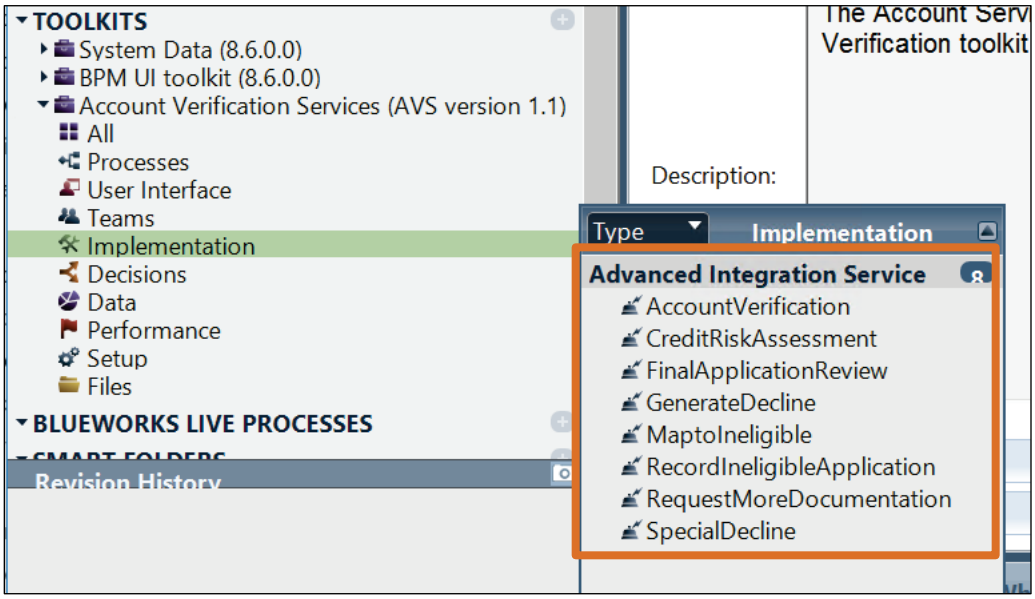
11. Click **AVS Version 1.1** to add the dependency to the **AccountServicesApp** process application.



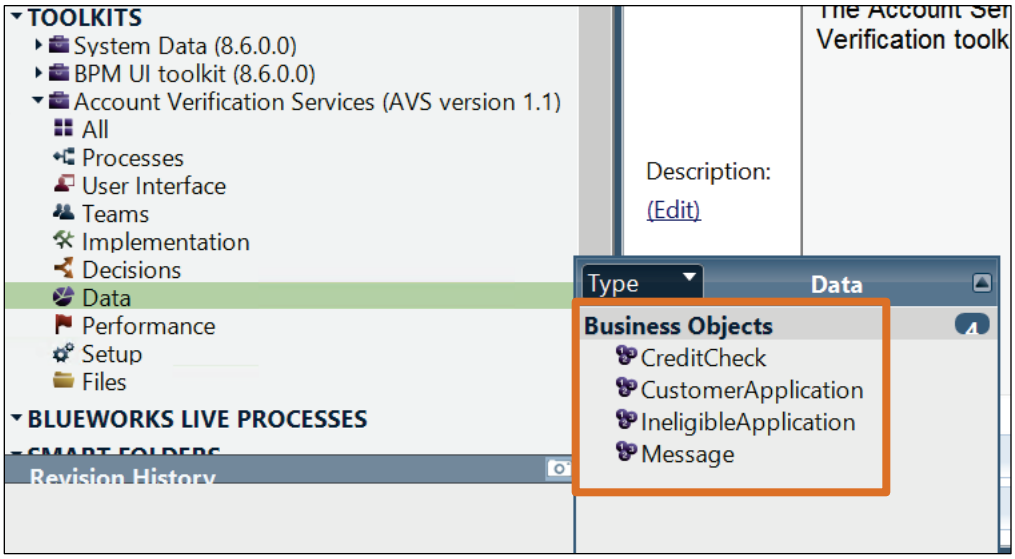
12. Verify that the newly added dependency is listed under **Toolkits**.



13. Expand **Account Verification Services (AVS version 1.1)** and click **Implementation** to view its contents.

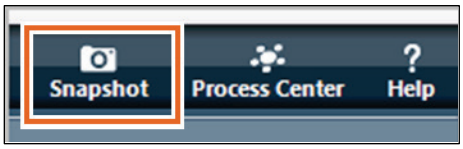


14. Click **Data** to see a list of business objects from the SCA module.



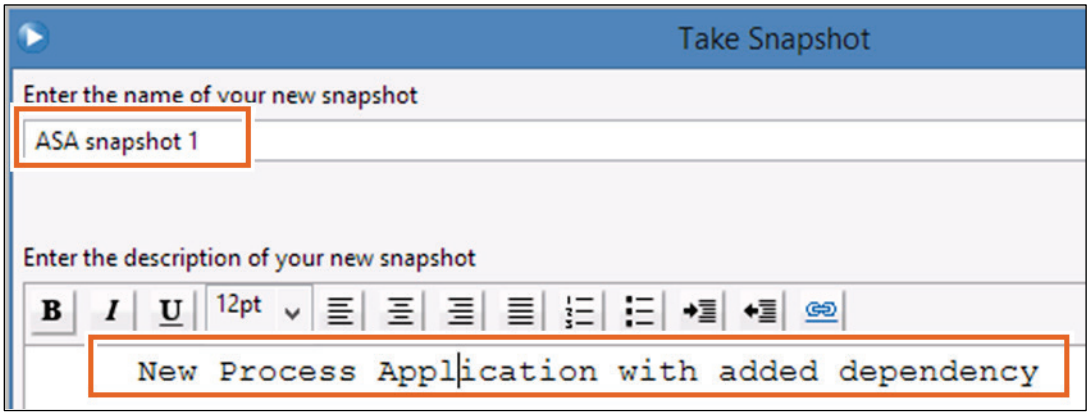
11. Deploying the Account Services process application

1. Before you deploy the process application, you must take its snapshot. In the Process Designer, click **Snapshot**.



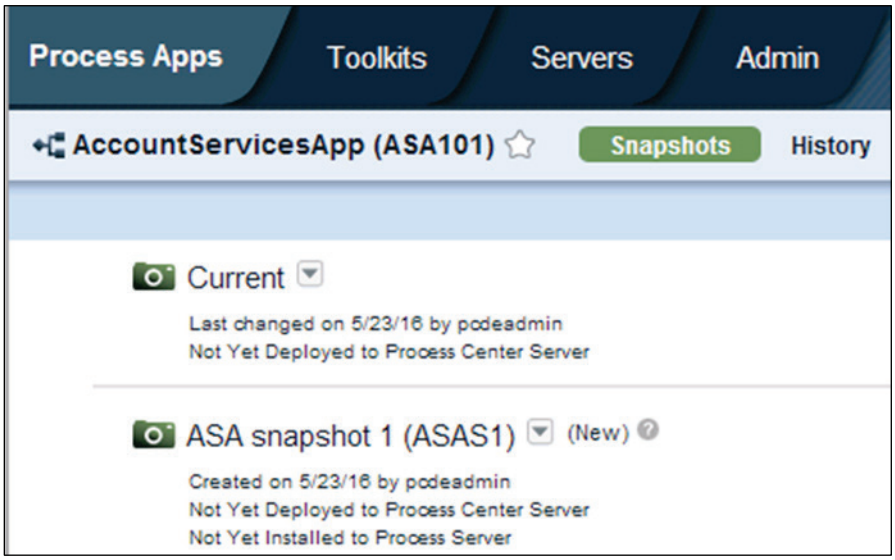
2. In the Take Snapshot window, enter ASA `snapshot 1` in the **Enter the name of your new snapshot** field.

3. Enter `New Process Application with added dependency` in the **Enter the description of your new snapshot** field. Several rich text formatting options are available here.



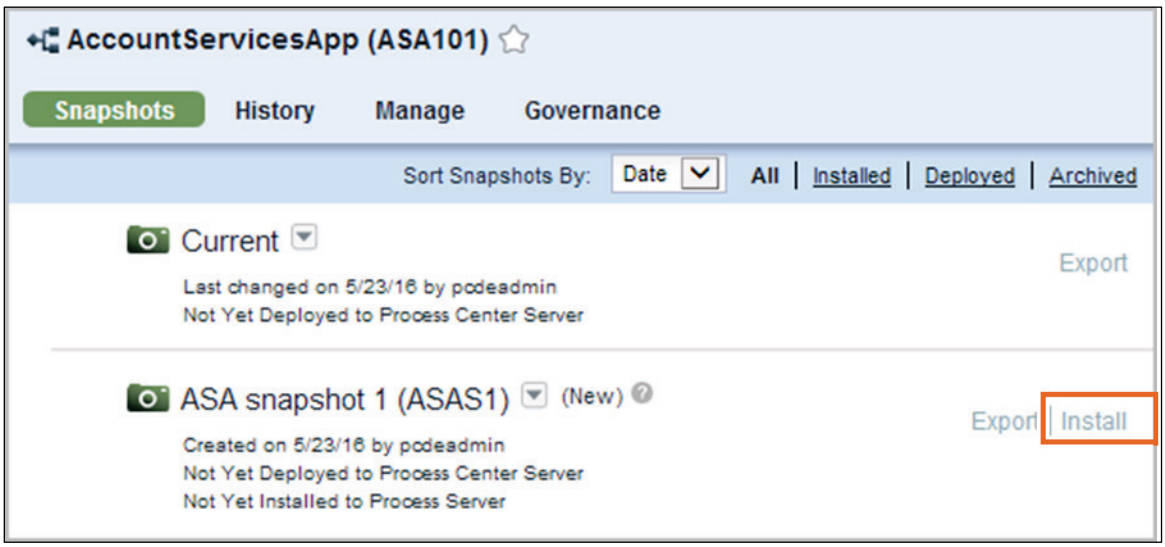
4. Click **OK**.
5. Verify that the new snapshot is listed following the **Revision History** section.
6. Minimize the Process Designer window as you are going to work with it again. If system resources are running low, you can also choose to close the IBM Process Designer.
7. Switch to the IBM Integration Designer window. Verify that you are in the Process Center perspective and click the **Process Apps** tab.

8. Click **AccountServicesApp (ASA101)**. The newly created snapshot is displayed under the **Snapshots** tab.

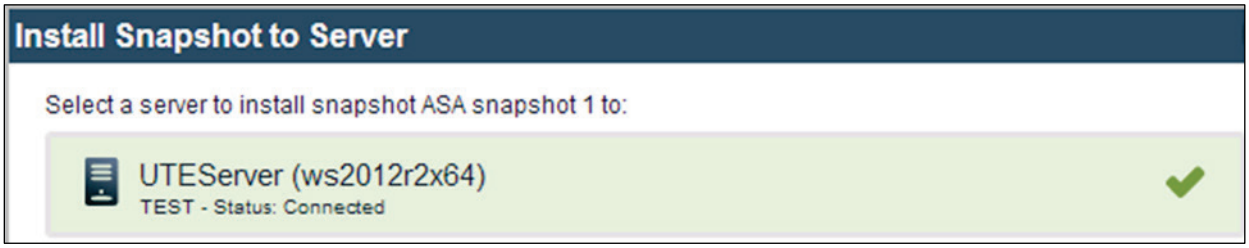


The message that follows the snapshot gives the deployment status. Currently, the process application is not deployed.

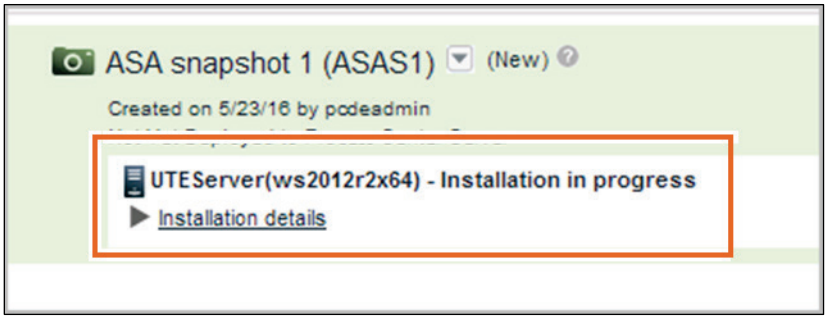
9. If IBM Process Server v8.6 is not started already, go ahead and start it by using the desktop shortcut.
10. Return to the **Process Apps** tab and click the **AccountServicesApp (ASA101)** link.
11. To the right of **ASA snapshot 1**, click **Install**.



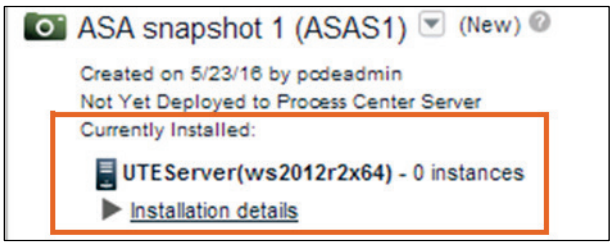
12. In the “Install Snapshot to Server” window, select **UTEServer**. A check mark is displayed to the right.



13. Click **Install** at the lower right of the window to begin deployment. It takes several minutes for the process application to install on the Process Server test environment. In the meantime, the status changes to **Installation in progress**.

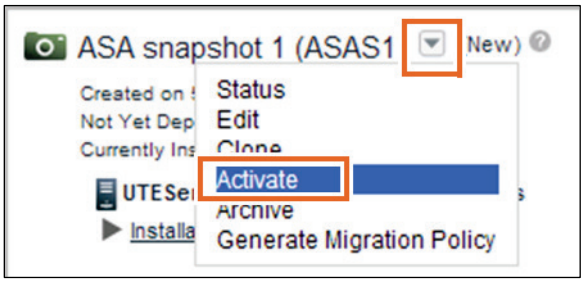


14. Do not click other options in the Process Center. When the snapshot is installed, the status that is displayed is updated to **Currently Installed**.

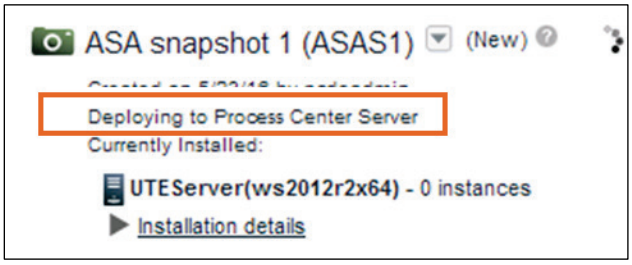


The process application is not yet deployed to the Process Center. Note the status: `Not yet Deployed to Process Center Server`

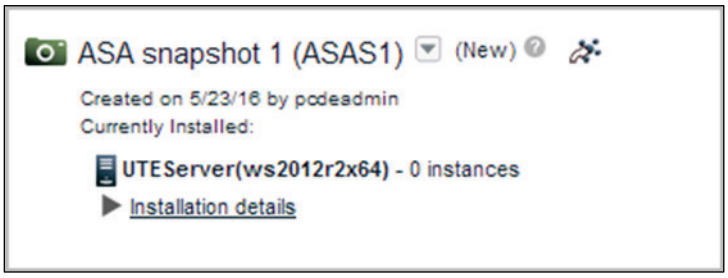
15. Click the down arrow next to the snapshot and select **Activate**.



16. It takes several minutes for process application to deploy. In the meantime, the status changes to “Deploying to Process Center Server”.

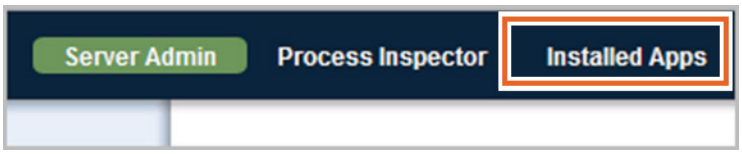


17. When the deployment is done, verify that the “Deploying to Process Center Server” message is no longer displayed.

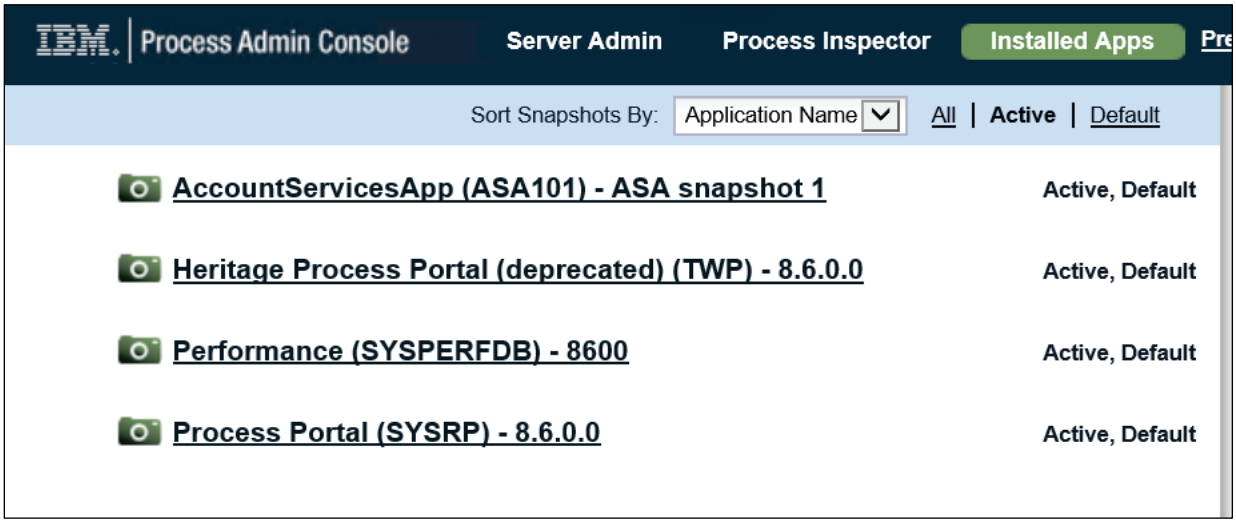


18. Switch to the IBM Process Designer window. If you closed it earlier, you can start it by double-clicking the shortcut icon on the desktop.
19. If prompted, log in using `pcdeadmin` and `web1sphere` for the **User Name** and **Password** fields.
20. Verify that you are in the Process Center perspective and click the **Servers** tab.
21. Click the **Configure Server** link.

22. Click the down arrow to expand **I Understand the Risks** and click **Add Exception**.
23. On the next screen, accept the defaults and click **Confirm Security Exception**.
24. Log in using `admin` and `web1sphere` for the **User Name** and **Password** fields.
25. Click **Installed Apps** at the top.



26. Verify that **AccountServicesApp** is listed among the deployed applications.

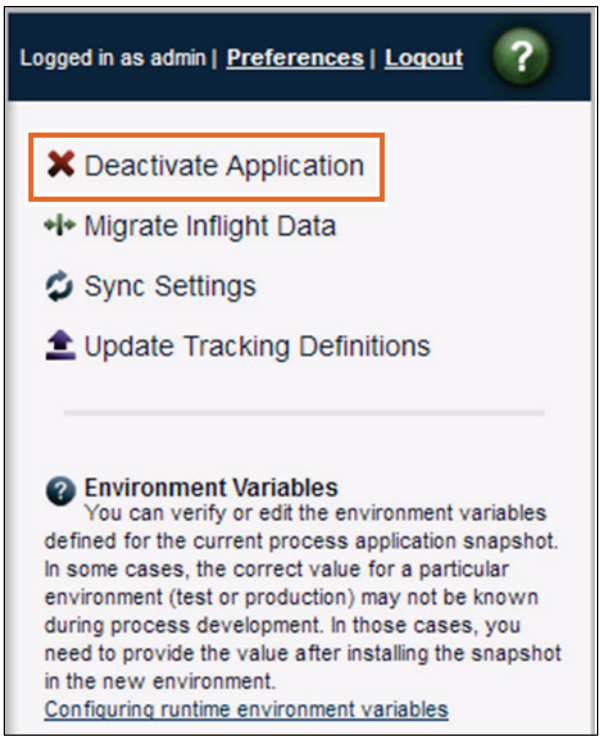


12. Removing the Account Services process application.

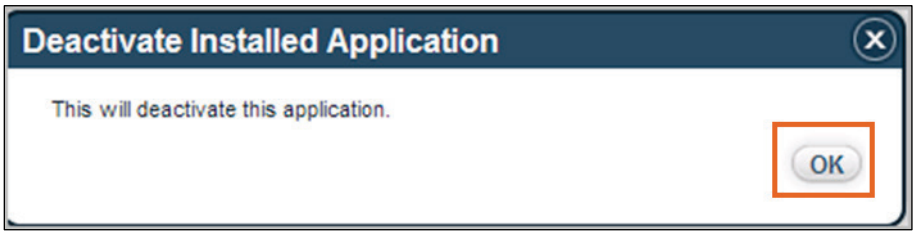
In this section, you learn how to stop and undeploy process applications so that you can control what applications are available to users.

1. In the **Installed Apps** pane of the console, click **AccountServicesApp (ASA101) - ASA snapshot 1**.

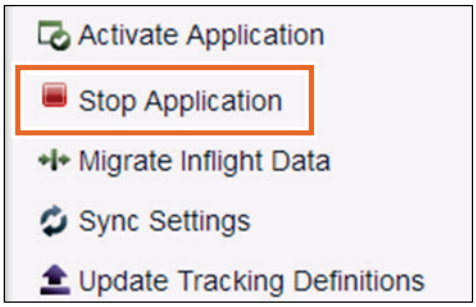
2. Click **Deactivate Application** in the upper-right corner.



3. In the Deactivate Installed Application dialog box, click **OK** to confirm the deactivation.

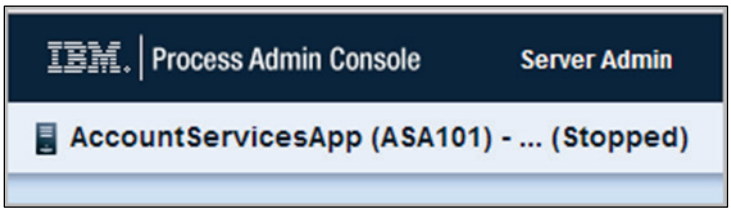


4. When the application is deactivated, a new **Stop Application** link is listed on the right. Click that link.

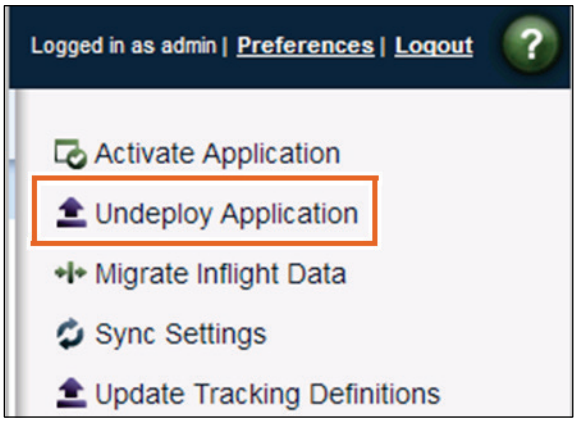


5. In the Stop Installed Application dialog box, click **OK**.

6. It takes a few minutes for the application to stop. When it stops, the dialog box closes and a stopped status is displayed at the top. If you receive an error that the connection is lost, close that message box and check the status of the application. If it is stopped, then you can continue, or else click the **Stop Application** link again.

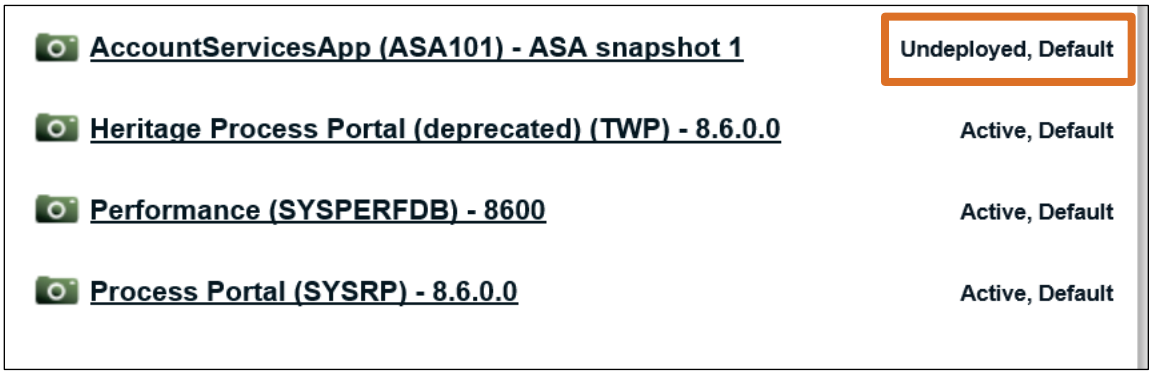


7. When the application stops, a new **Undeploy Application** link is listed on the right. Click that link.



8. In the Undeploy Application dialog box, click **OK**. Do not click **OK** multiple times as it takes a few minutes for the application to uninstall, and the dialog box stays open. When the application is uninstalled, the dialog box closes.

9. Click **All** to verify that the application has a status of **Undeployed**.

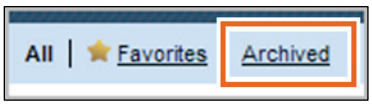


10. Log out and close the Process Admin Console.

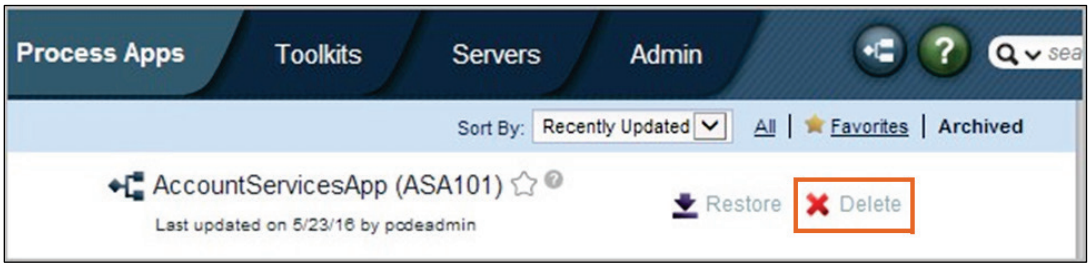
13. Deleting the Account Services process application snapshot.

1. Switch to the Process Center perspective window.
2. In the Process Apps tab, click **AccountServicesApp**.
3. Click **Manage**.
4. Click **Archive Process App**.
5. Click **Archive** in the dialog box. It takes a couple of minutes until the application is archived. No further confirmation is displayed.

6. Wait a few minutes and then click the **Process Apps** tab. If AccountServicesApp is listed, then it is still being archived. Eventually it goes away from the list. Click the **Archived** link.



7. Only one application is listed in the Archived category. The application is now ready to be deleted. Click **Delete**.



8. Click **Delete** again in the dialog box. The application is now permanently deleted.
9. Close Process Designer.
10. If you are continuing to the next exercise, do not stop the servers.
11. Close IBM Integration Designer.
12. Close any other open windows.

Results:
In this exercise, you used the IBM Process Center console to explore IBM Process Center. You deployed a process application to the Process Server test environment and then archived and deleted that application.

IBM Training

Advanced Integration services

IBM Business Process manager V8.6

IBM Training

Unit objectives

- Explain Advanced Integration services (AIS)
- Describe the business value of AIS
- Describe the steps to create an AIS

Unit objectives

IBM Training

Topics

- AIS introduction
- Approaches in creation of AIS

Topics

IBM Training

AIS introduction

AIS introduction

IBM Training

What is an Advanced Integration service?

- An Advanced Integration service is a collaboration between a business user who is working with IBM Process Designer and an integration developer who is working with IBM Integration Designer
- An Advanced Integration service is used to call a service that is implemented in IBM Integration Designer from a business process definition (BPD)
- SCA modules encapsulate Advanced Integration services
  - You can changes services without affecting users of the service
- Advanced Integration services are available only with IBM Business Process Manager Advanced

Advanced Integration services

What is an Advanced Integration service?

An Advanced Integration service is used to call a service implemented in IBM Integration Designer from a business process definition (BPD) (through a system task) or another service (through a nested service). An Advanced Integration service is a collaboration between a business user who is working with IBM Process Designer and an integration developer who is working with IBM Integration Designer.

When using IBM Integration Designer and IBM Process Designer together, collaborate before defining your Advanced Integration service.

IBM Training

Business value of an Advanced Integration service

- Many business processes require a combination of human tasks and integrations with back-end systems
- Integrations might be complex in nature and require the orchestration of multiple service invocations
- Such complex integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL)
- Used by a process developer without the need to know much technical detail about the implementation of the technical service invoked
  - Process developer does not need to know Service Component Architecture (SCA), Java programming, transactional qualifiers, JDBC, and other implementation details
  - The only thing that the process developer needs to know is the I/O interface of the AIS and a bit of functional specification about its behavior

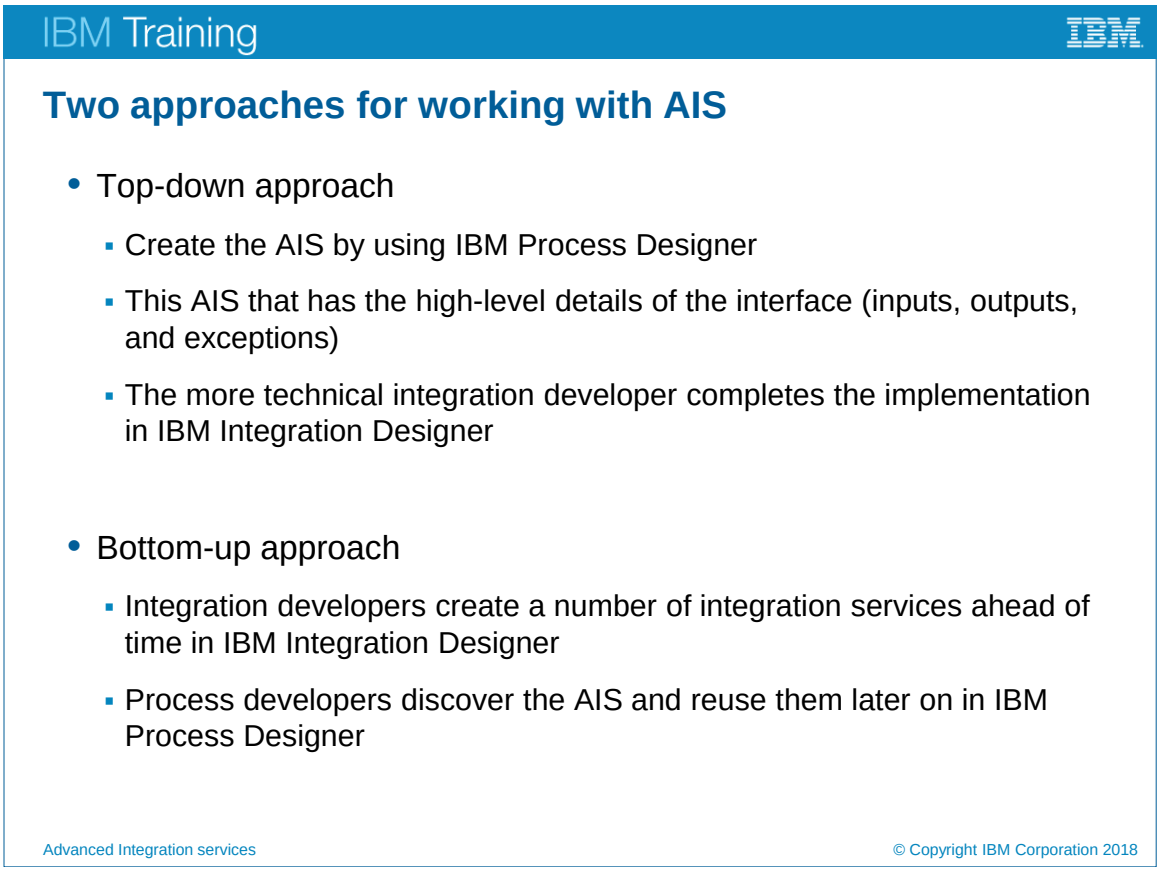
Advanced Integration services

Business value of an Advanced Integration service

Most business processes require a combination of human tasks and integrations with back-end systems. Certain integrations might be complex in nature and require the orchestration of multiple service invocations, and a higher level of transactional integrity and compensation. Such integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL). Process Designer has Advanced Integration services (AIS) and you can use it to invoke those IBM Integration Designer implementations.

For example, your business process might need a list of computer parts in your warehouses in Canada. Checking with an integration developer, you realize that a service is being built in Integration Designer to query the Canadian warehouses and return an inventory list of the computer parts available. You can create an Advanced Integration service that would use this Integration Designer service as an activity in your business process.

Approaches in creation of AIS



Two approaches for working with AIS

A business process can take advantage of integration services in two fundamental ways:

•The business process designer, by using Process Designer, can create an AIS that has the high-level details of the interface (inputs, outputs, and exceptions), and have the more technical integration developer complete the implementation in IBM Integration Designer. It is a top-down approach.

•Conversely, integration developers can create a number of integration services ahead of time, and the process developer can discover them and reuse them later on, for example, by packaging them in a Toolkit. It is called the bottom-up approach.

IBM Training

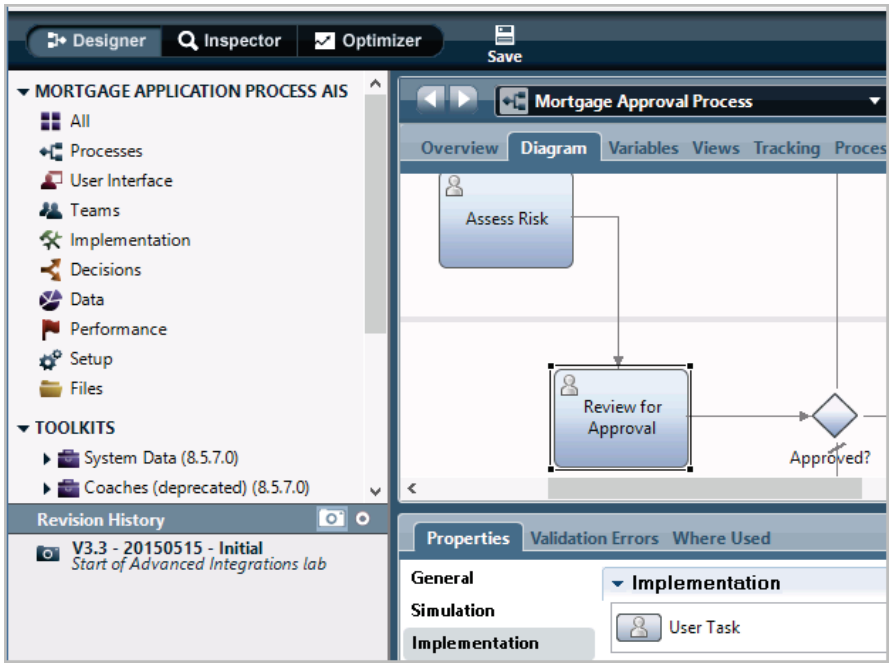
Top-down approach

- In a top-down approach, the Process Developer defines the inputs and outputs of the AIS and sends this information to the Process Center repository
- The integration developer receives the unimplemented Advanced Integration service and provides the implementation to be delivered back to the Process Center
- The Application Developer retrieves the implementation back in Process Designer for testing and playback
- The top-down approach works well when the service does not exist

Top-down approach

IBM Training

Step 1: Create process application in Process Designer

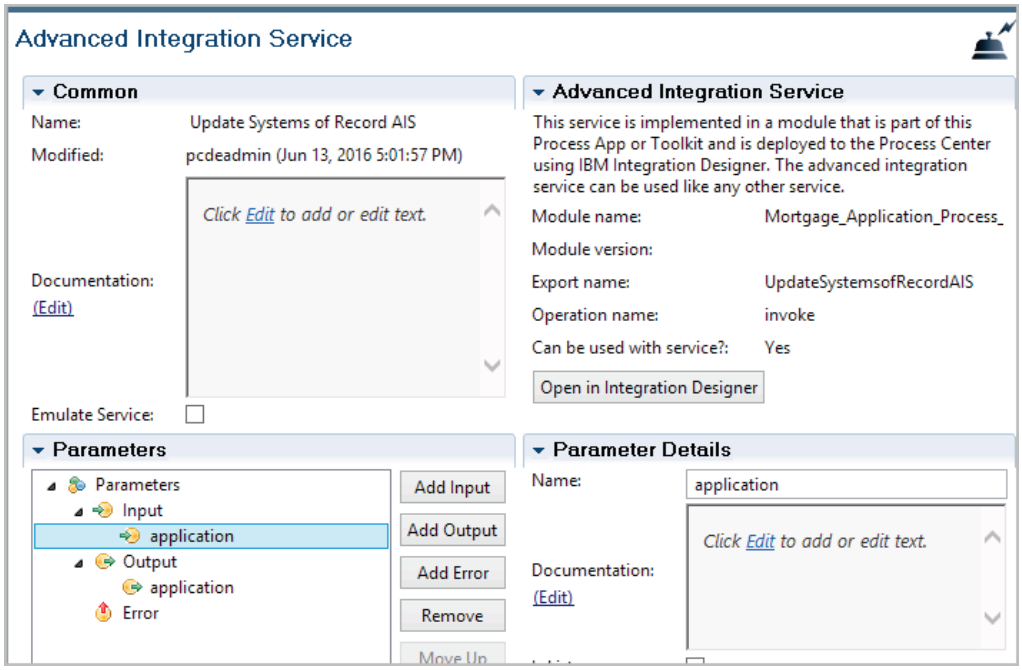


Advanced Integration services

Step 1: Create process application in Process Designer

IBM Training

Step 2: Define inputs and outputs in AIS

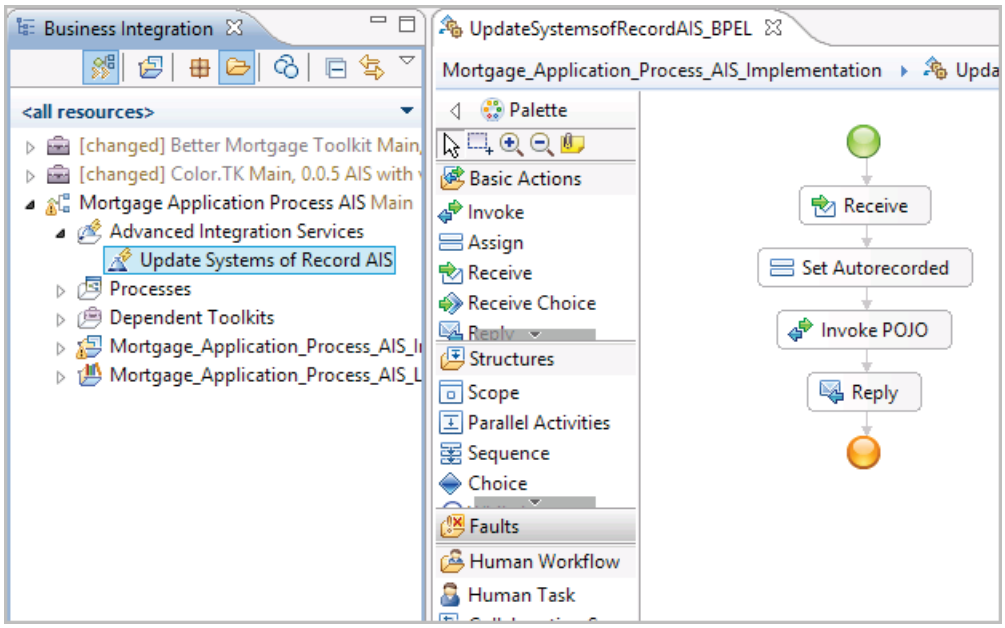


Advanced Integration services

Step 2: Define inputs and outputs in AIS

IBM Training

Step 3: Implement the AIS

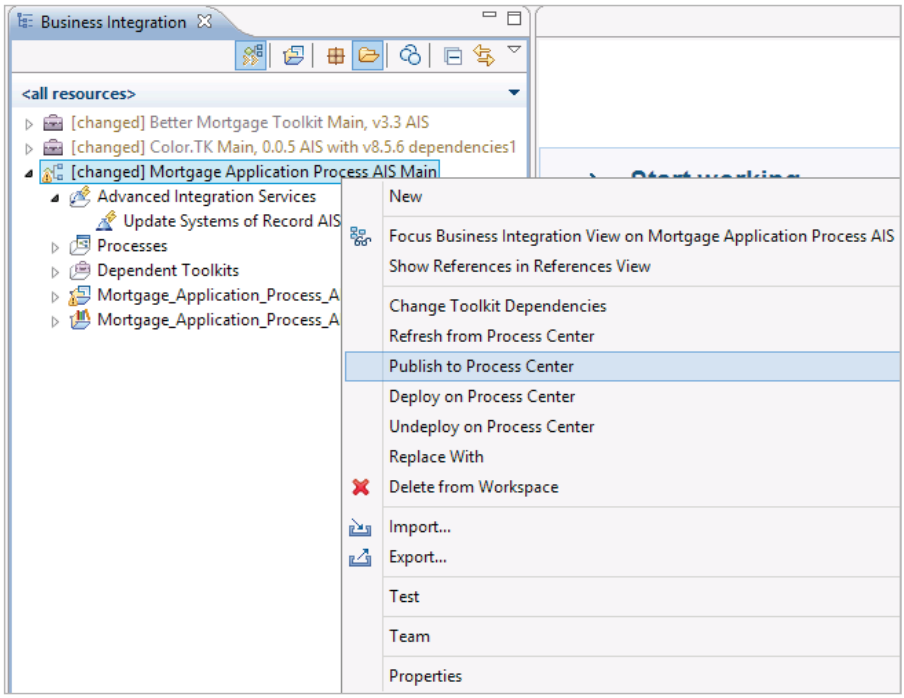


Advanced Integration services

Step 3: Implement the AIS

IBM Training

Step 4: Refresh and publish changes to Process Center



Advanced Integration services

Step 4: Refresh and publish changes to Process Center

IBM Training

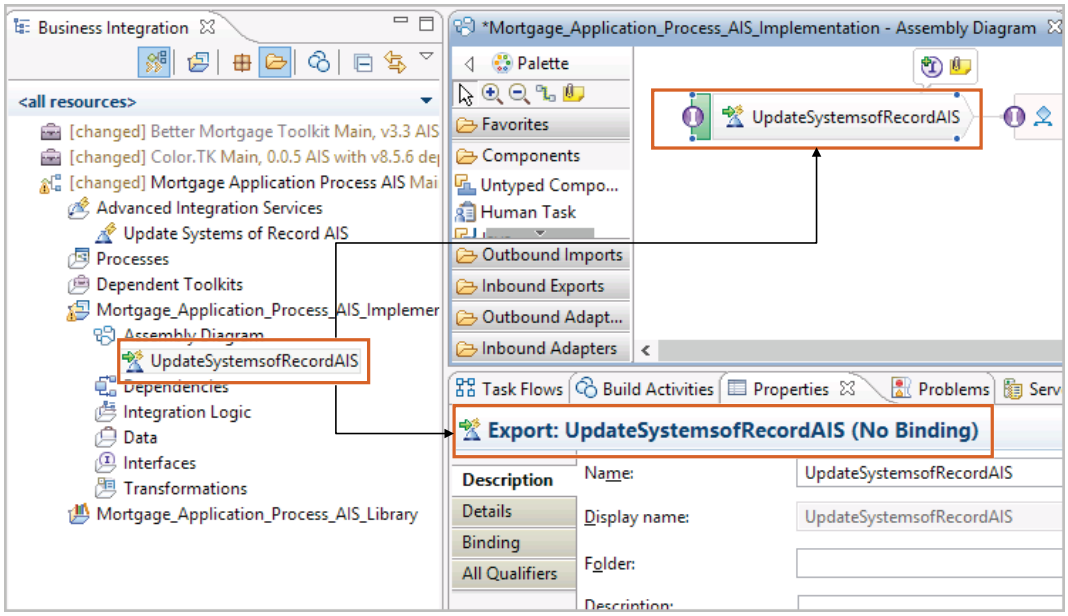
Bottom-up approach

- Design and develop the Advanced Integration service in Integration Designer
- Deliver the interface and the implementation to IBM Process Designer
- Define a BPD that uses the Advanced Integration service
- The bottom-up approach works well when a mature SOA architecture with reusable services exists

Bottom-up approach

IBM Training

Bottom-up approach, Step 1: Create export in Integration Designer



Advanced Integration services

Bottom-up approach, Step 1: Create export in Integration Designer

IBM Training

Bottom-up approach: Steps 2, 3, 4

- Step 2: Expose the export as AIS in Process Designer
- Step 3: Playback
- Step 4: Deploy the process application

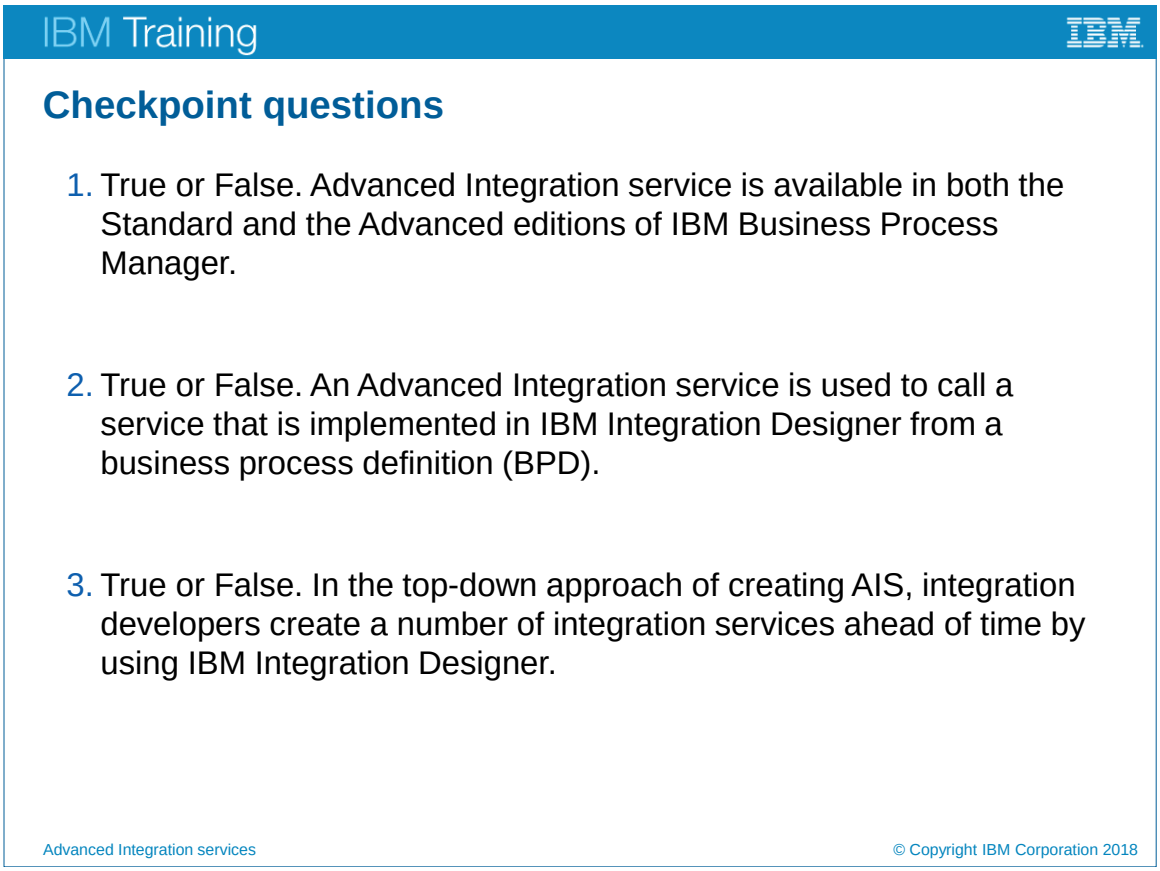
Bottom-up approach: Steps 2, 3, 4

IBM Training

Unit summary

- Explain Advanced Integration services (AIS)
- Describe the business value of AIS
- Describe the steps to create an AIS

Unit summary



Checkpoint questions

Write your answers here:

1.
2.
3.

IBM Training

Checkpoint answers

1. False. AIS is available only in IBM BPM Advanced.
2. True.
3. False. In the bottom-up approach, integration developers create a number of integration services ahead of time by using IBM Integration Designer.

Checkpoint answers

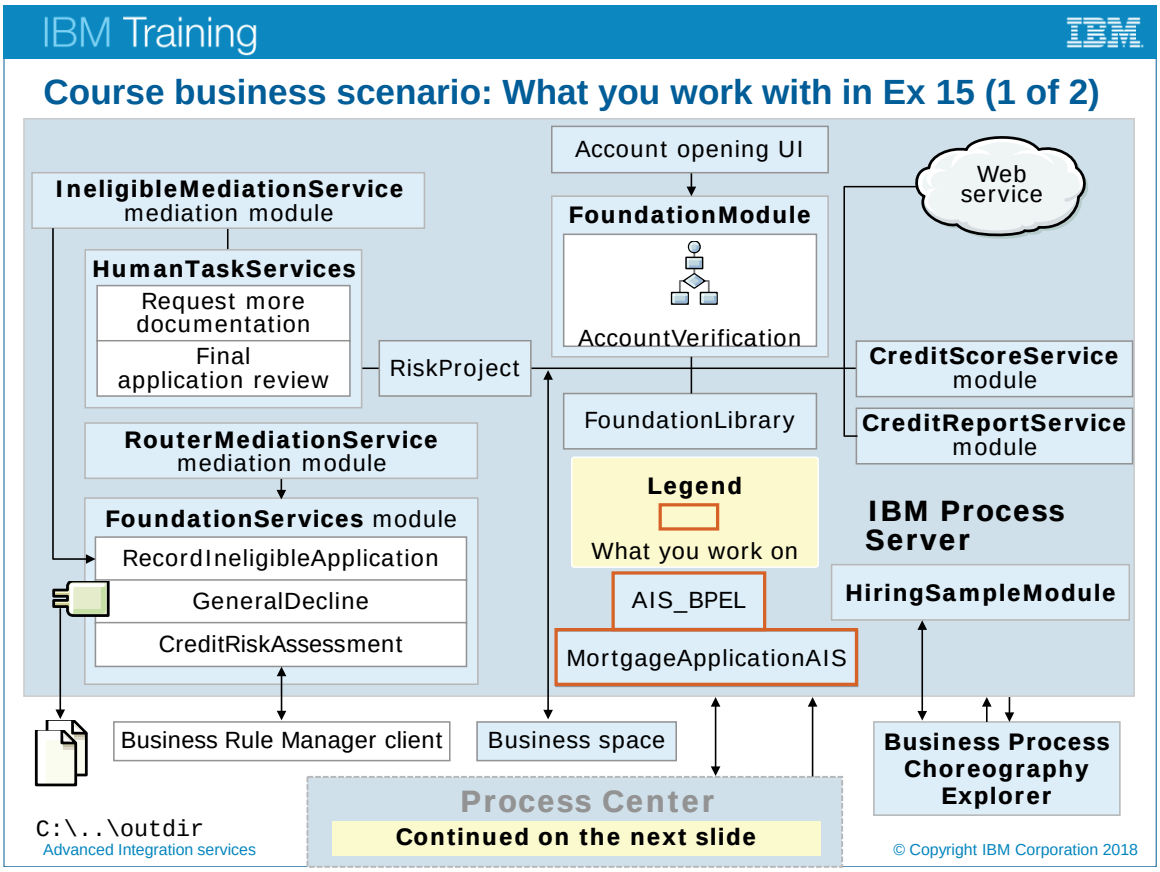
IBM Training

Exercise 15:Implementing Advanced Integration services

After completing this exercise, you should be able to:

- Create an AIS in IBM Process Designer
- Implement an AIS in IBM Integration Designer
- Unit test an AIS in IBM Integration Designer
- Invoke a BPEL process from a coach and BPD

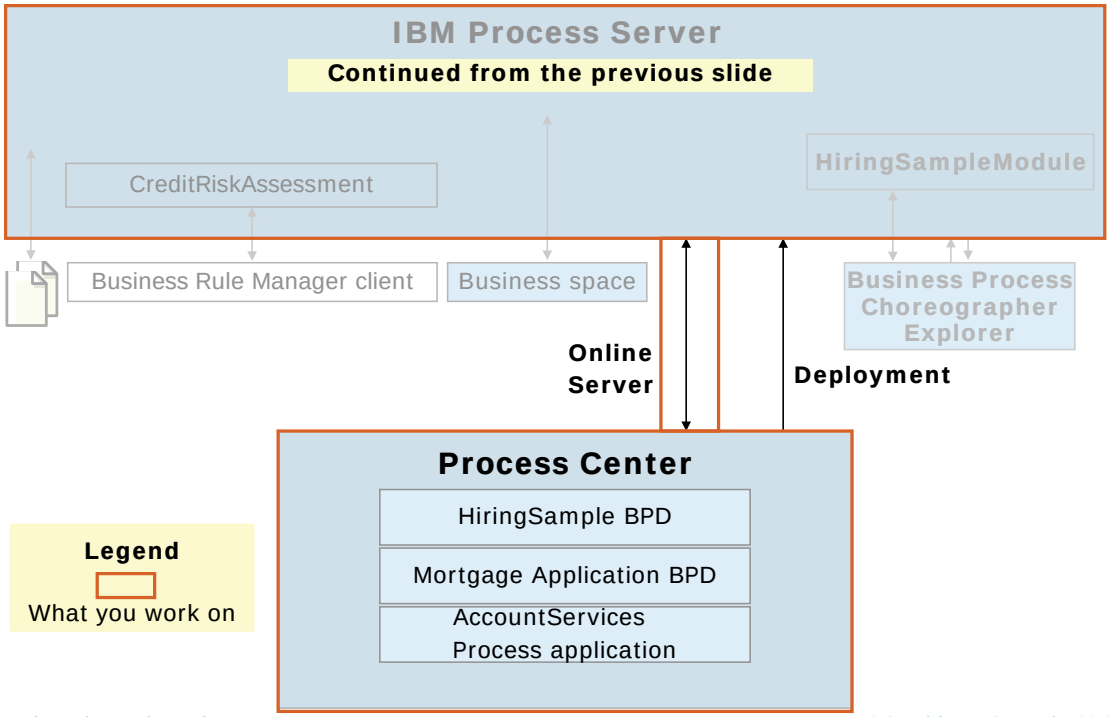
Exercise 15: Implementing Advanced Integration services



Course business scenario: What you work with in Ex 15

IBM Training

Course business scenario: What you work with in Ex 15 (2 of 2)



Advanced Integration services

Exercise 15: Implementing Advanced Integration services

Purpose:
In this exercise, you create an Advanced Integration service that is used by IBM Process Designer to invoke IBM Integration Designer implementations.

Most IBM BPM implementations require a combination of human tasks and integrations with back-end systems. Certain integrations might be complex in nature and require the orchestration of multiple service invocations, and a higher level of transactional integrity and compensation. Such integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL). IBM Process Designer has Advanced Integration services (AIS) which can invoke those IBM Integration Designer implementations.

This exercise shows you how Process Designer and Integration Designer can work in concert to create a fully integrated end-to-end IBM Business Process Manager solution.

A business process can take advantage of integration services in two fundamental ways:

- The business process designer, by using Process Designer, can create an AIS that has the high-level details of the interface (inputs, outputs, and exceptions). The more technical integration developer then completes the implementation in IBM Integration Designer. This approach is a top-down approach.
- Conversely, integration developers can create a number of integration services ahead of time. Then, the process developer can discover them and reuse them later on, for example, by packaging them in a Toolkit. This approach is called the bottom-up approach.

This exercise captures the steps to create a simple Advanced Integration service in IBM Business Process Manager by using the top-down approach

Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Center, and the IBM Process Server test environment.

Part 1. Start the IBM Process Center environment

If the Process Center environment is running from the previous exercise, then go to Part 2: Create an AIS by using IBM Process Designer. Otherwise, complete the following steps to start the environment:

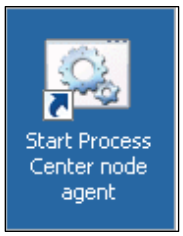
1. Start the deployment manager

1. On your Windows desktop in your lab environment, select the Start Process Center deployment manager shortcut. Double-click the shortcut or press Enter to start the server.

2. When the deployment manager starts, you are prompted to press any key to continue. Press any key to close the command window.

2. Start the node agent

1. On your Windows desktop, select the Start Process Center node agent shortcut. Double-click the shortcut or press Enter to start the server.



2. A DOS command window is displayed; press any key to continue when prompted.

3. Start the process center single cluster.

1. On your Windows desktop, select the Start Process Center Cluster shortcut. Double-click the shortcut or press Enter to start the cluster member.

2. A DOS command window is displayed, and the IBM Process Center server instance starts. Press any key to continue when prompted.

Part 2. Create an Advanced Integration service (AIS) by using IBM Process Designer

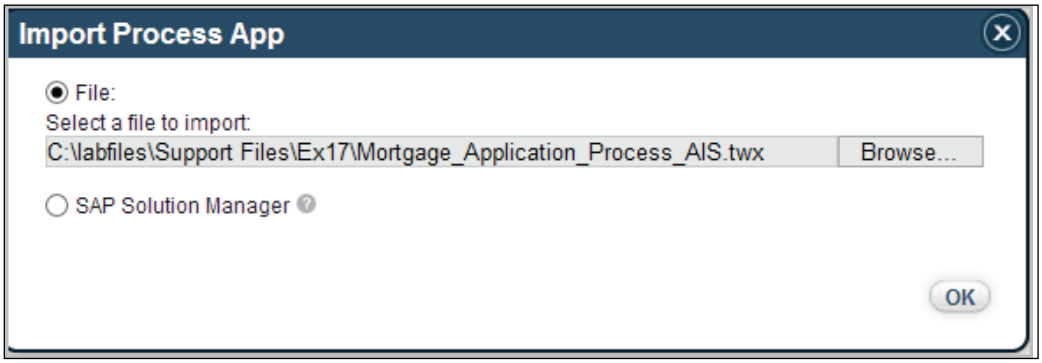
1. Start the desktop IBM Process Designer.

1. Double-click the Start IBM Process Designer icon on the desktop.
2. Enter `pcdeadmin` in the User name field and `web1sphere` in the Password field, and click Log In.
3. When the Security Alert dialog box is displayed, click **Yes** to proceed.
4. After a few moments, click **Yes** one more time.
5. Make sure that you are in the Process Center view. If you are in the Designer view, then click the Process Center icon at the upper right of the Designer view.

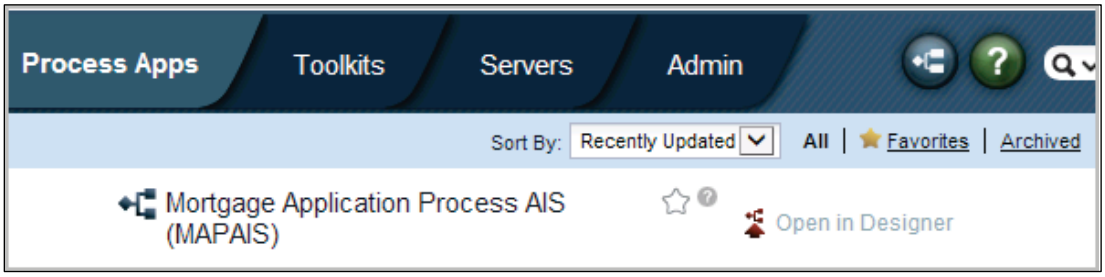
2. Import a Mortgage Application Process application in the Process Center.

After importing this process app, you then augment it with a new Advanced Integration service.

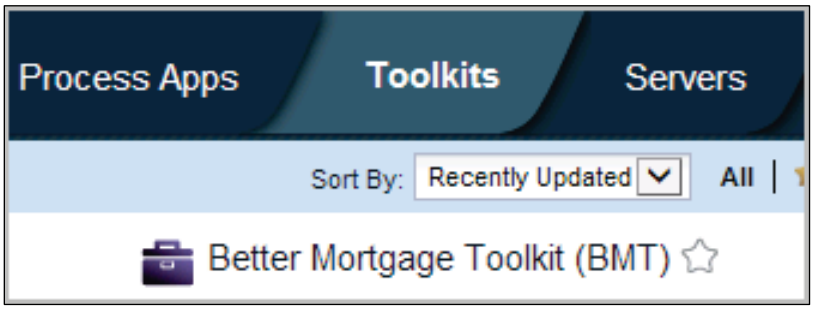
1. In the Process Apps tab, click **Import Process App** on the right.
2. In the Import Process App window, click **Browse**.
3. Go to `C:\labfiles\Support Files\Ex15` and select Mortgage_Application_Process_AIS.twx.
4. Click Open.
5. Verify that the selected application is listed in the Select the file to import field, and click OK.



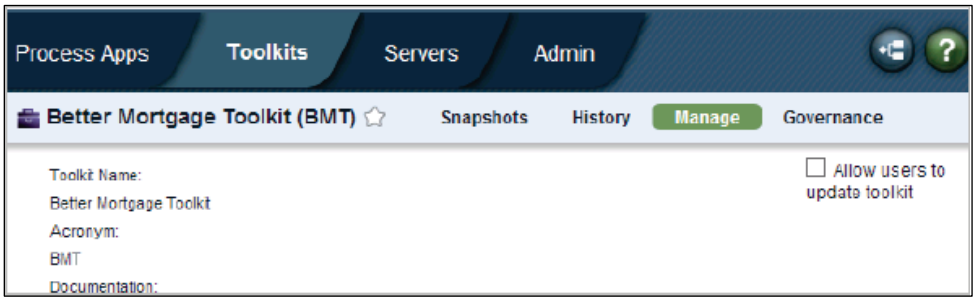
6. In the Import Process App window, click **Import**. It takes few minutes for the process app to import.
7. Verify that the process app was imported successfully by making sure that Mortgage Application Process AIS is listed in the Process Apps tab.



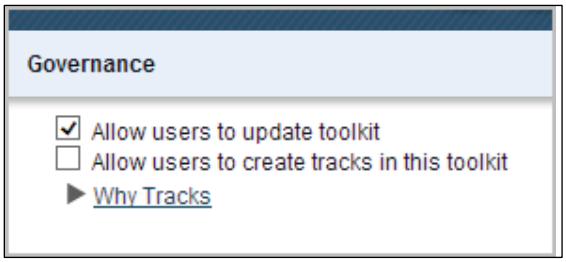
8. Switch to the **Toolkits** tab and click **Better Mortgage Toolkit**.



9. For the Integration Designer to access the toolkit that is used by the process app you are working with in this exercise, you need to make sure that it can be updated. Click **Manage** and then select the **Allow users to update toolkit** check box.



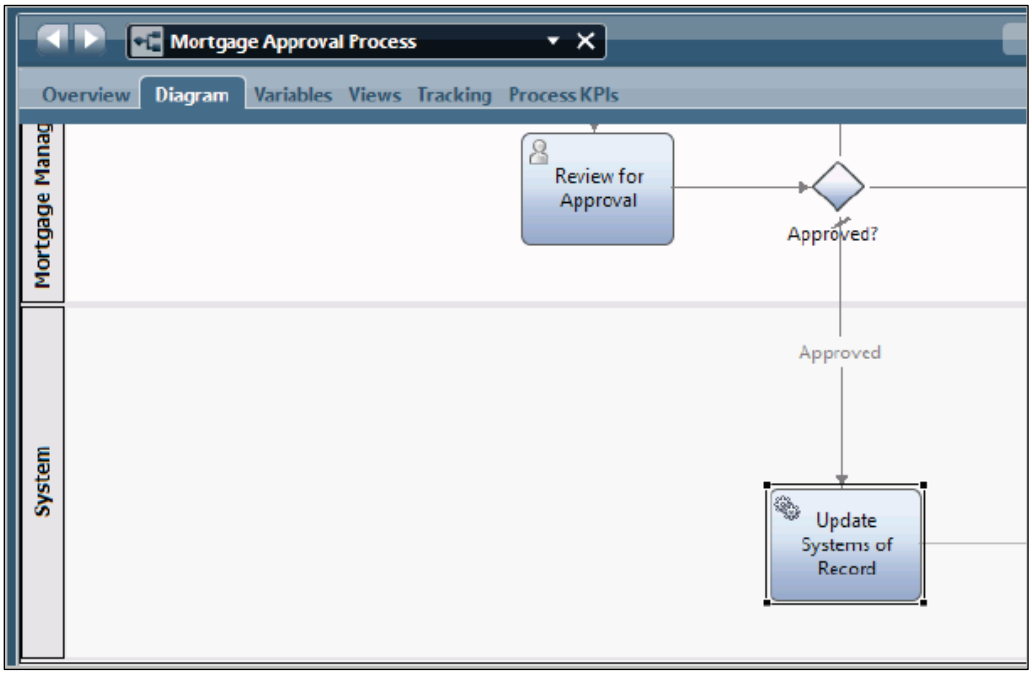
10. After selection, verify that the view looks similar to the following image:



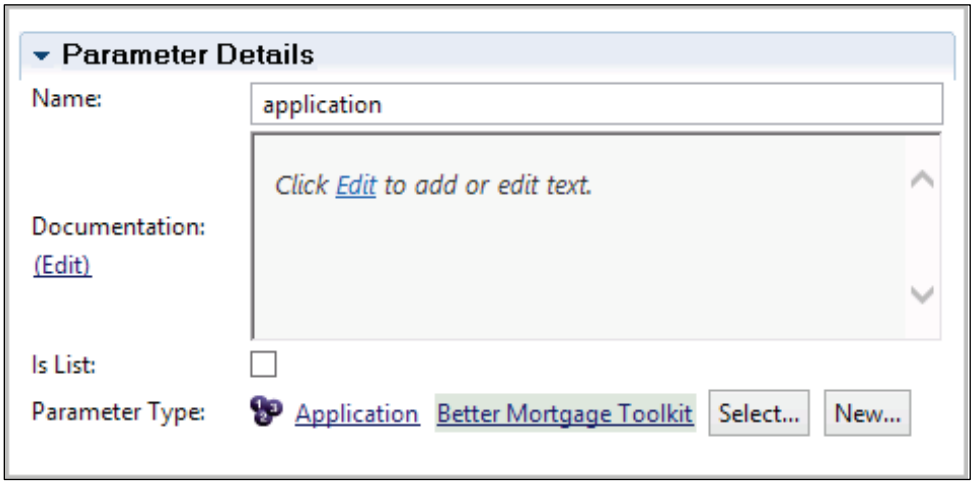
11. Click the **Toolkits** tab, and then click **Color TK**. This toolkit is another that the process app uses.
12. For the Integration Designer to access the Toolkit that is used by the process app you are working with in this exercise, you need to make sure that it can be updated. Click **Manage** and then select the **Allow users to update toolkit** check box.

3. Create an AIS in the business process.

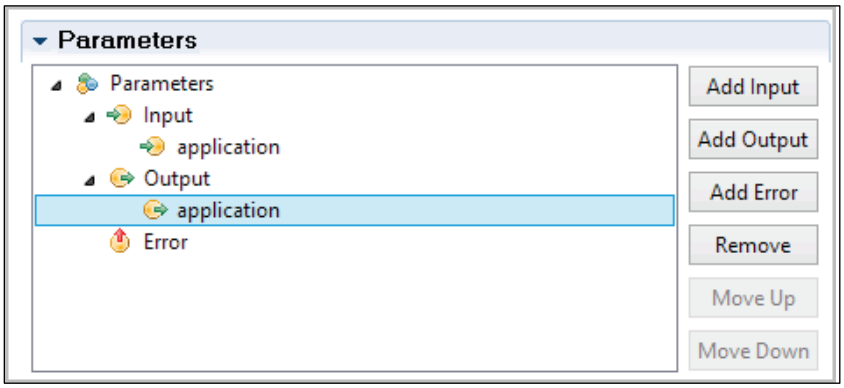
1. Switch to the Process Apps tab and then click **Open in Designer** to the right of Mortgage Application Process AIS.
2. In the process library, click Processes and then double-click the Mortgage Approval Process under Business Process Definitions.
3. The process diagram opens in the main editor pane. Scroll down to the **System** lane. In the next step, you add an Advanced Integration service to replace the current **Update Systems of Record** Activity with an Advanced Integration service.



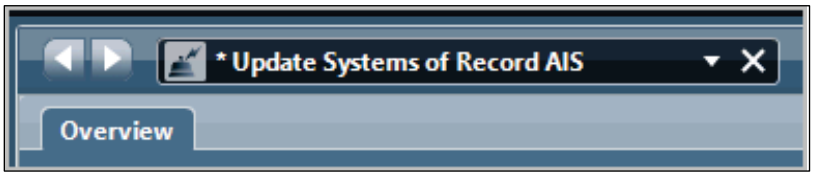
4. In the process library, next to **Implementation** click the + icon. Then, in the **New** window, click Advanced Integration Service.
5. In the New Advanced Integration Service window, enter **Update Systems of Record AIS** for the Name and then click **Finish**.
6. The Advanced Integration service editor opens. You define the input and output of this new service. In the Parameters section, click **Add Input**.
7. In the Parameter Details section, enter `application` for the **Name** and then set the Parameter Type to **Application**.



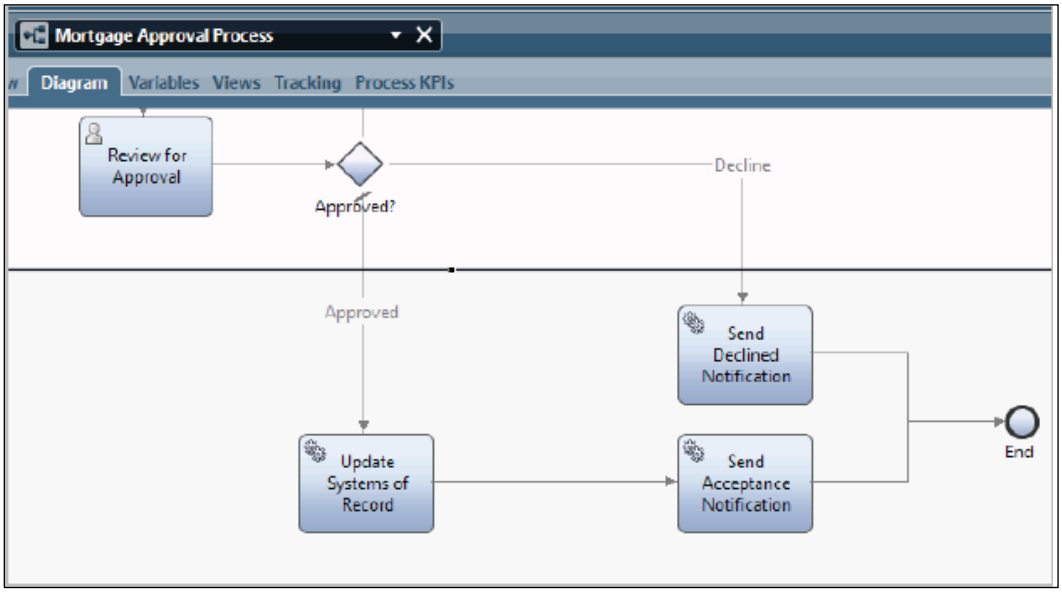
8. Add an output parameter by clicking **Add Output** in the **Parameters** section. In the Parameter Details section, enter `application` for the **Name** and then set the Parameter Type to **Application**.
9. The Parameters section looks similar to the following image:



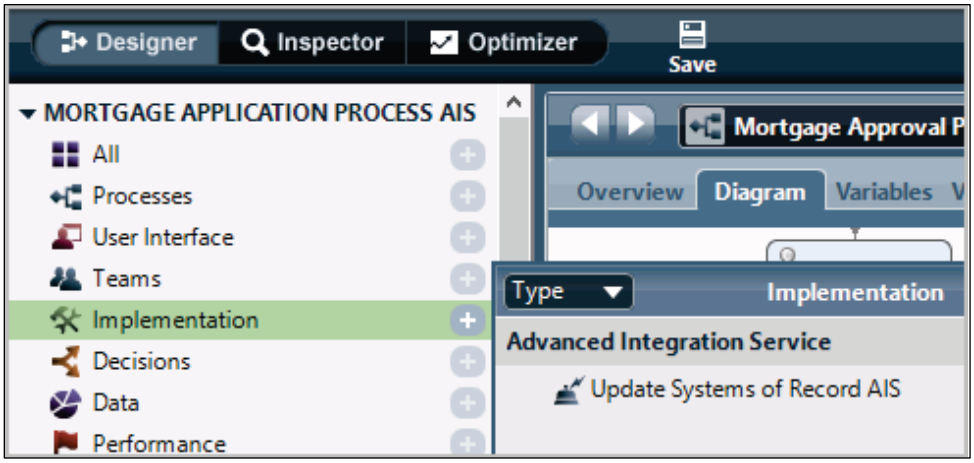
10. Save (Ctrl-S) your work and close the Advanced Integration service editor by clicking X.



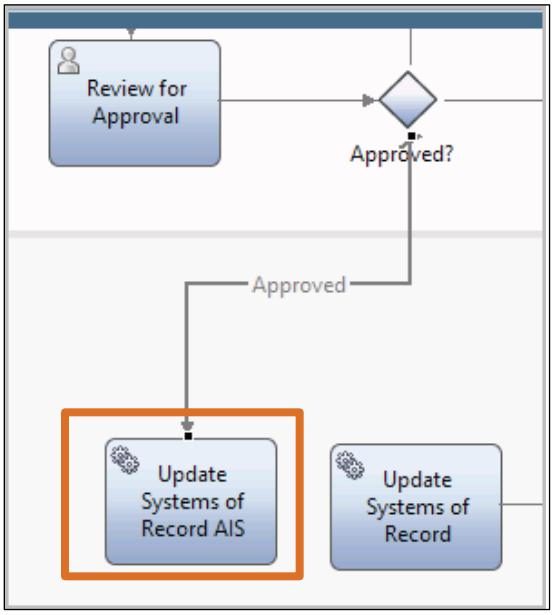
11. You now replace the Activity Update Systems of Record in the process with the recently created Update Systems of Record AIS Advanced Integration service. Make sure that the Mortgage Approval Process is displayed in the Diagram tab.



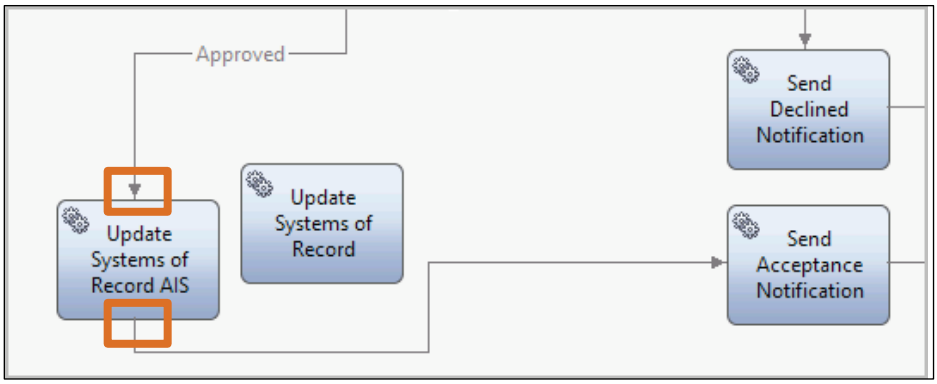
12. In the process library, click Implementation, and under **Advanced Integration Service** drag **Update Systems of Record AIS** to the **System** swimlane next to the existing Update Systems of Record activity.



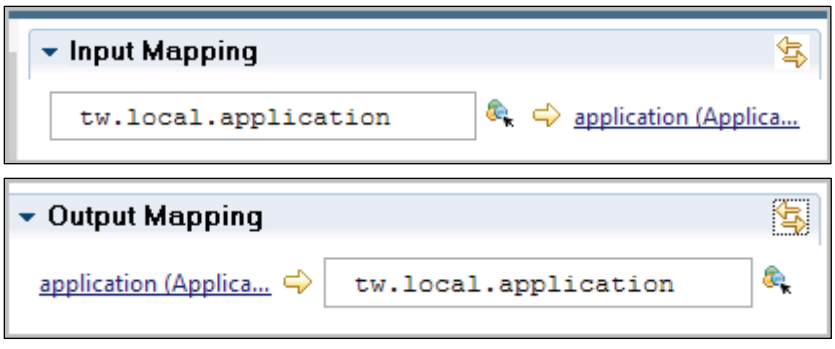
13. Click the tip of the sequence flow arrow that goes into **Update Systems of Record**, and drag it to the input of **Update Systems of Record AIS**.



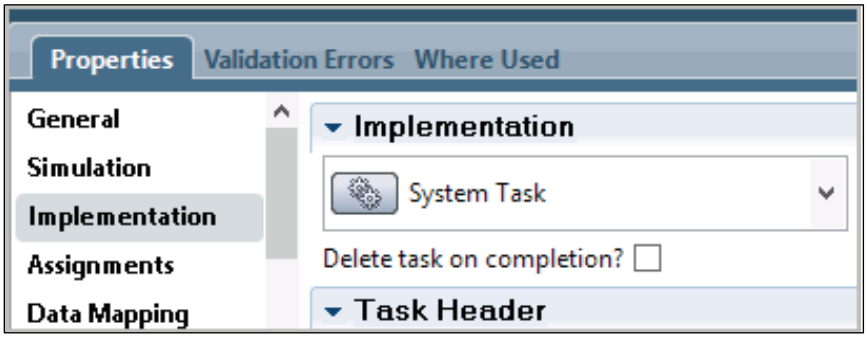
14. Drag the starting end of the arrow that flows out of **Update Systems of Record** to the corresponding spot in Update Systems of Record AIS. The diagram looks similar to the following image:



15. Right-click Update Systems of Record then click **Edit > Delete**.
16. Select **Update Systems of Record AIS** and click the **Data Mapping** tab.
17. Click the auto mapping icons for both input and output mapping. The two fields are automatically populated with tw.local.application:



18. Click **Implementation** and clear **Delete task on completion** check box.



19. Save your work (Ctrl-S).
20. Close Mortgage Approval Process by clicking X.
21. Click Process Center at the upper right corner to switch to the Process Center view.
22. Minimize IBM Process Designer.

    Your work as a process author is done for now. You created the definition of an Advanced Integration service, which a more technical user implements by using the IBM Integration Designer.

Part 3. Implement the AIS in IBM Integration Designer

This section, you take the unimplemented Advanced Integration service and complete its implementation in IBM Integration Designer.

1. Create the AIS workspace.

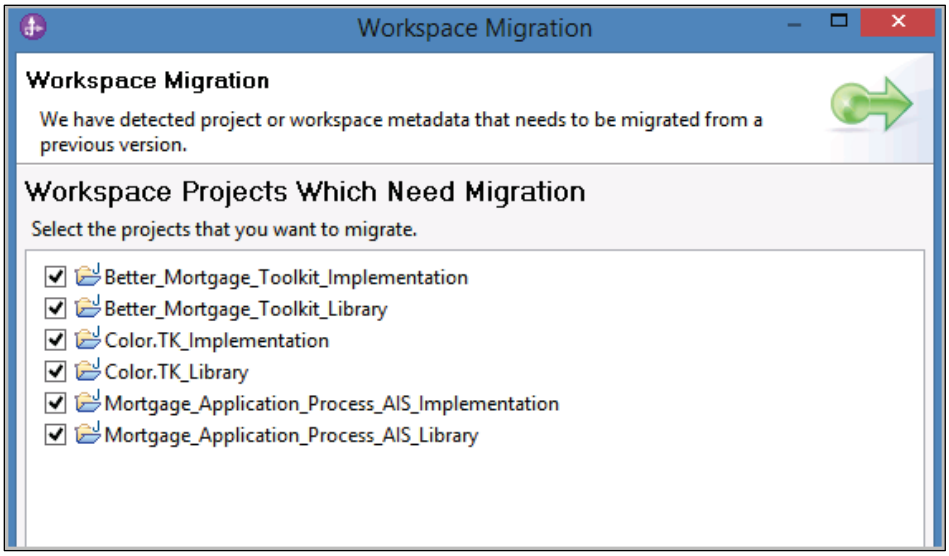
1. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
2. Double-click the shortcut that is labeled **Exercise 15**. Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of the Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears. If you get a message that the server is already set to publish, then click **OK**.
3. Close the **Getting Started** tab.
4. Click **Window > Switch to Process Center**.
5. In the Process Center Login window, enter the following credentials:

   •**Process Center URI**: `http://localhost:9081/ProcessCenter`

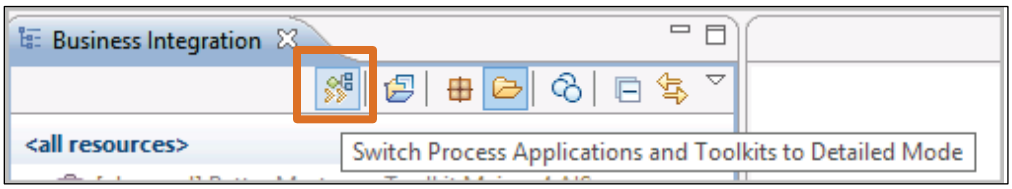
   •**User Name**: `pcdeadmin`

   •**Password**: `web1sphere`
6. Click Login.
7. When the Secure Storage window opens, click Cancel.
8. If a Security Alert window is displayed asking to proceed, then click Yes each time.

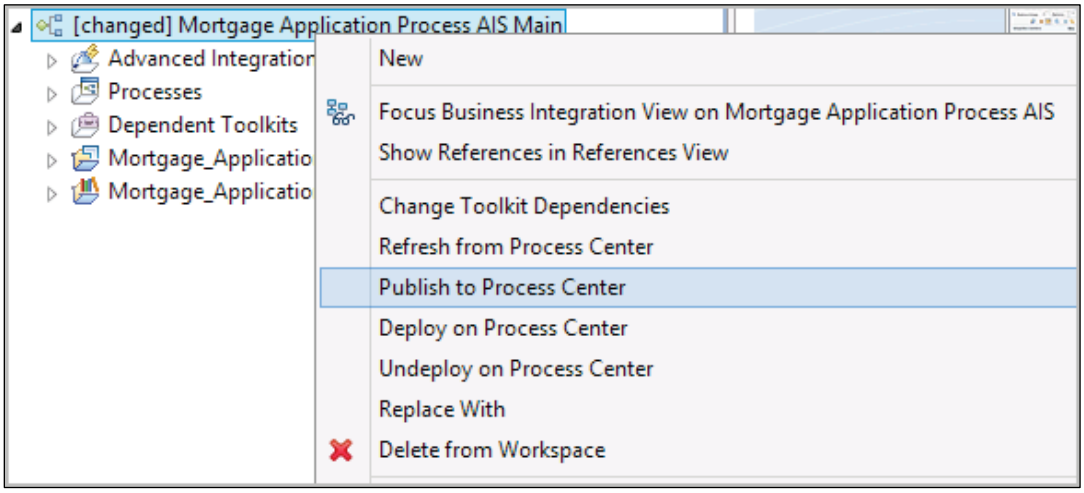
9. If the Secure Storage window opens again, click Cancel.
10. When the Secure Storage Warning window opens, click OK.
11. Close the “Getting Started with IBM Process Center 8.6.0.0” welcome screen, by clicking the X at the upper right corner of the window.
12. Next to **Mortgage Application Process AIS (MAPAIS),** click **Open in workspace**.
13. In the Workspace Migration window make sure that all the projects are selected and click **Next**.



14. Click **Next** one more time.
15. Click **Finish** to complete the migration.
16. When the Migration Validation window is displayed confirming that the migration validation completed successfully, click OK.
17. Close the Migration Results view.
18. Wait until the workspace is built. It takes few minutes for the project to build. A progress bar is on the lower right corner of the Integration Designer.
19. If a window is displayed confirming that changes are detected, then click OK to close that window.
20. Switch to the detail mode by clicking the Switch process Applications and Toolkits to Detailed Mode icon.

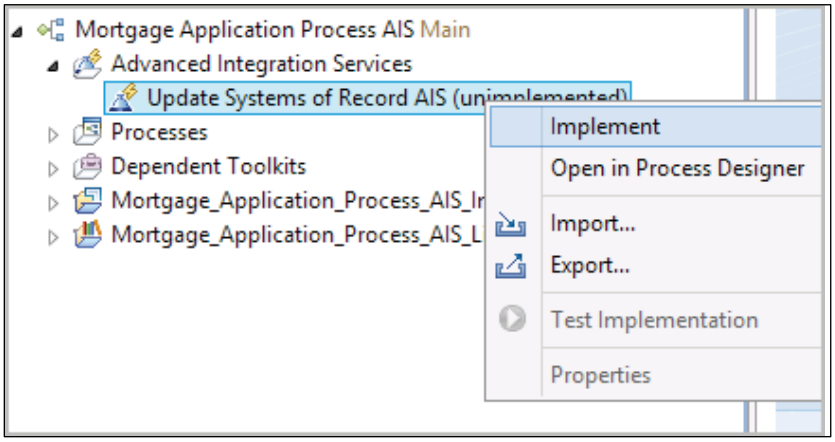


21. Notice that Mortgage Approval Process has [changed] next to it. Right-click **[changed] Mortgage Approval Process Main**, and click **Publish to Process Center**.

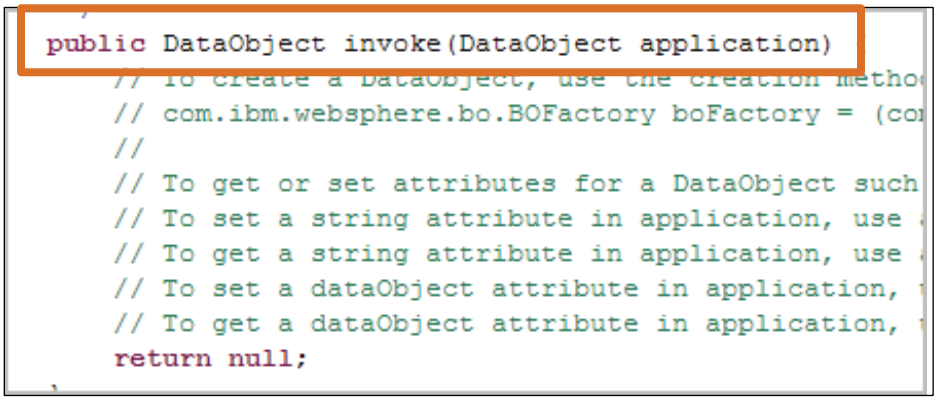


2. Implement the Update Systems of Record AIS service.

1. In the project tree of the Business Integration perspective, expand **Advanced Integration Services** under **Mortgage Application Process. AIS Main**. Notice that Update Systems of Record AIS is labeled as unimplemented. Right-click it and click Implement.



2. In the **Implement Advanced Integration Service** window, select **Java component**, and click **Finish**. This action opens a Java editor with predefined functions and methods.
3. In the Java editor, scroll down to the bottom, and locate the invoke method.



You write some logic that sets the mortgage application status to “RECORDED FROM AIS” and print some information to the log files, for initial debugging purposes. Delete return null at the end of the invoke method.

4. Open Windows Explorer and browse to: C:\labfiles\Support Files\Ex15
5. Open AIS_Recorded.txt in a text editor such as Notepad.
6. Copy the text in AIS_Recorded.txt and paste the content in the Invoke method.

```
//-----------------------------------------------------------------
//  Invoke
//-----------------------------------------------------------------
application.setString("status","RECORDED FROM AIS");
System.out.println("********<<Advanced Integration Service Java
Implementation successfully executed>>********");
return application;
```

7. Alternatively, you can manually replace the code with the following text:

```
application.setString("status","RECORDED FROM AIS");
System.out.println("********<<Advanced Integration Service
Java Implementation successfully executed>>********");
return application;
```

8. Remember to delete return null from the invoke method.
9. Save your changes and close the Java editor. You should have no errors.

Part 4. Unit Test the Update Systems of Record AIS Advanced Integration service in IBM Integration Designer

1. Publish your changes to Process Center.

1. Verify that Mortgage Approval Process has [changed] next to it.
2. Right-click [changed] Mortgage Approval Process Main, and click Publish to Process Center.

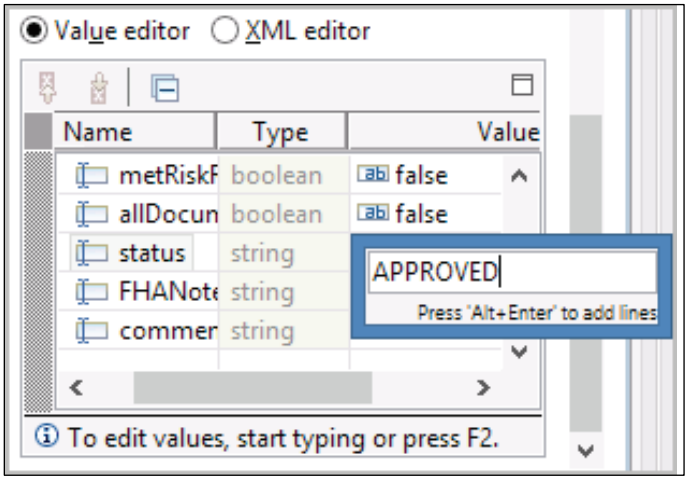
2. Start the UTE server (if it is not already running).

1. If the UTE Process Server is not running then double-click the desktop shortcut and start the server. Wait for the startup process to complete before continuing.

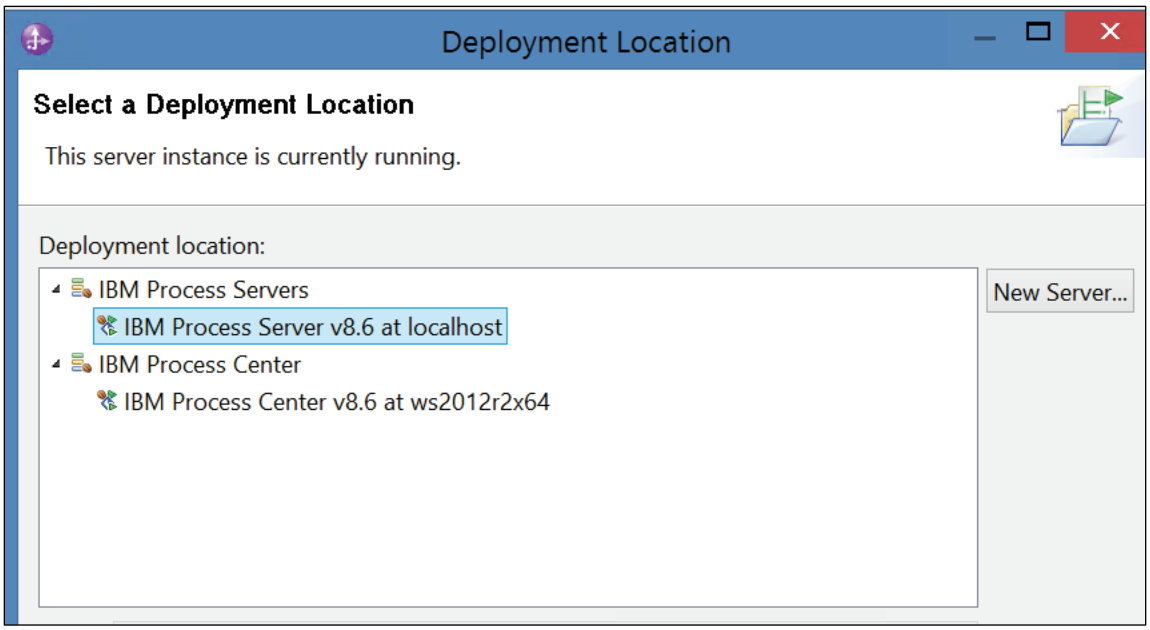
3. Run the AIS test.

1. In Business Integration, expand **Mortgage Application Process AIS Main > Advanced Integration Services**, then right-click **Update Systems of Record AIS**, and select **Test Implementation**.

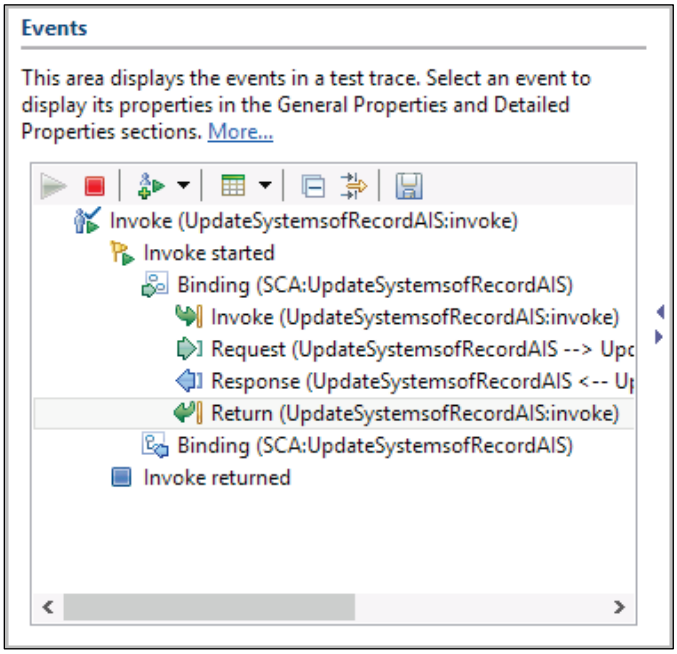
2. The Integration Test Client starts. In the Initial request parameters to the right, scroll through the value editor and locate the status field under the Name column.
3. Enter APPROVED in the Value column for **status**.



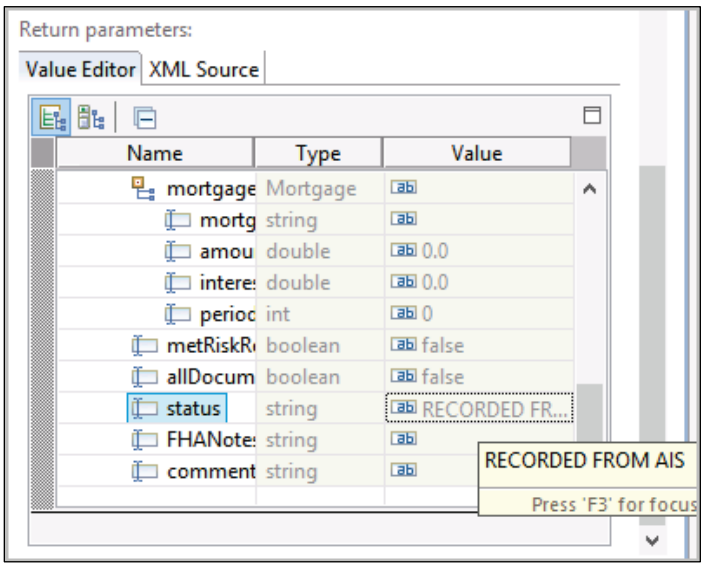
4. Under Events, click the Continue icon to start the Test.
5. In the Deployment Location window, expand IBM Process Servers and select **IBM Process Server v8.6.**. Click **Finish**.



6. In the User Login - Default Module Test window, accept the default values and click OK.
7. It takes few minutes after which the project is deployed to the test server and runs the AIS implementation. You see a trace of all the components that are being run, and a result. In the Events section, click **Return**.



8. In the return parameters section to the right, notice the value of **status**. It was earlier set to APPROVED, and it is now updated to RECORDED FROM AIS.



9. You did AIS unit testing successfully. Next, you use it in the BPD in IBM Process Designer.

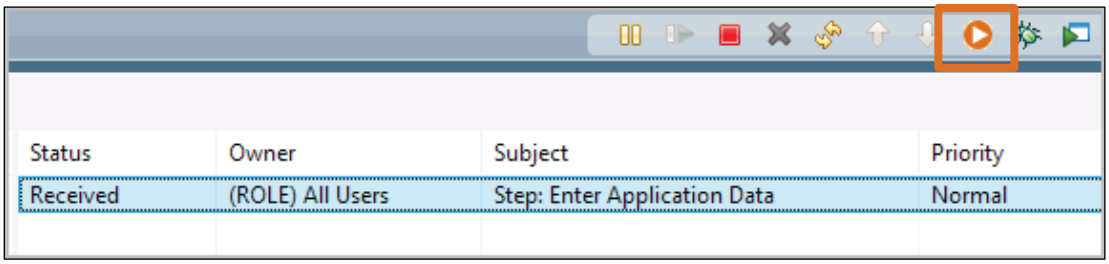
Part 5. Run the BPD in IBM Process Designer and execute the AIS

In the previous section, you tested and confirmed that the Advanced Integration service is working as expected. In this section, you take the completed AIS and execute from the BPD. You use the Process Inspector to walk through the business process and execute the Advanced Integration service.

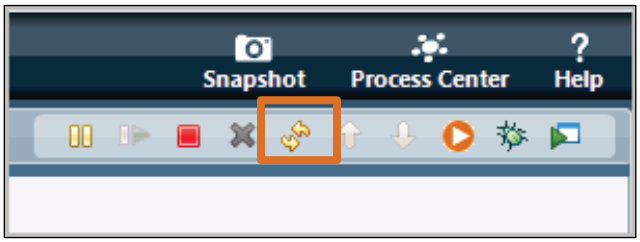
1. Test the end-to-end business process (including the AIS) in

Process Designer.

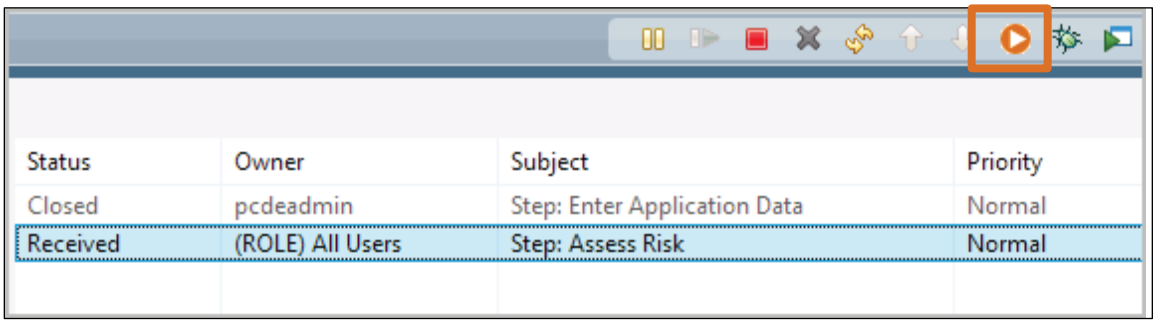
1. Minimize Integration Designer, and switch back to IBM Process Designer.
2. If needed, go to the Process Center view by clicking the Process Center icon at the upper right.
3. In the **Process Apps** tab, click **Open in Designer** next to **Mortgage Application Process AIS**.
4. Click **Processes** and then double-click **Mortgage Approval Process**.
5. Click the **Run Process** icon in the upper right corner to run the **Mortgage Approval Process**:
6. If prompted, click Yes to switch to the Inspector view to see the progress of the process instance.
7. The Inspector opens and the focus is on the running process instance. Select the task on the upper right pane (Enter Application Data), and click **Runs the selected task**.



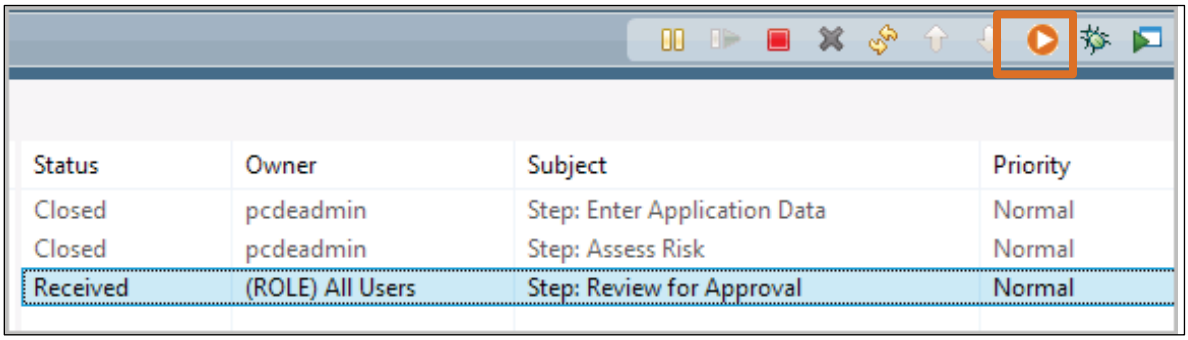
8. In the Select User window, select pcdeadmin and click OK.
9. If prompted with an untrusted connection, click I Understand the Risks and then click Add Exception. Click Confirm security exception.
10. After few minutes, the browser opens with the first coach of the mortgage approval process. All fields are optional for this exercise and you can leave them empty. Click Submit to complete the task.
11. Close the browser window.
12. Go back to Process Designer and click the refresh icon.



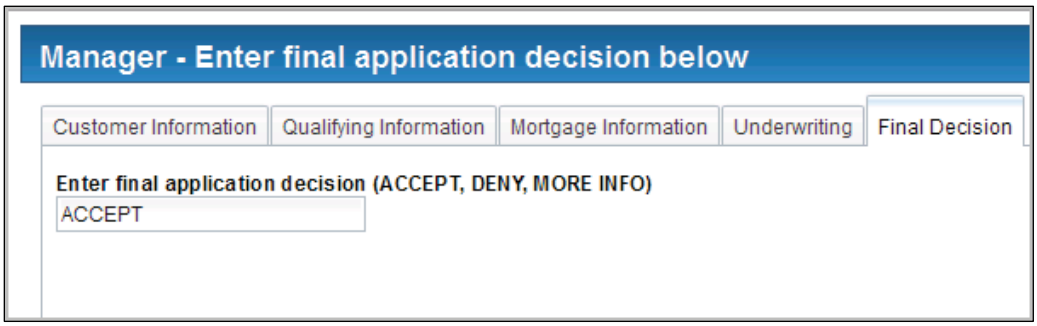
13. Find the second step on the stack (Assess Risk). Select it and click **Runs the selected task**. If you do not see it listed after some time, then click the instance on the left.



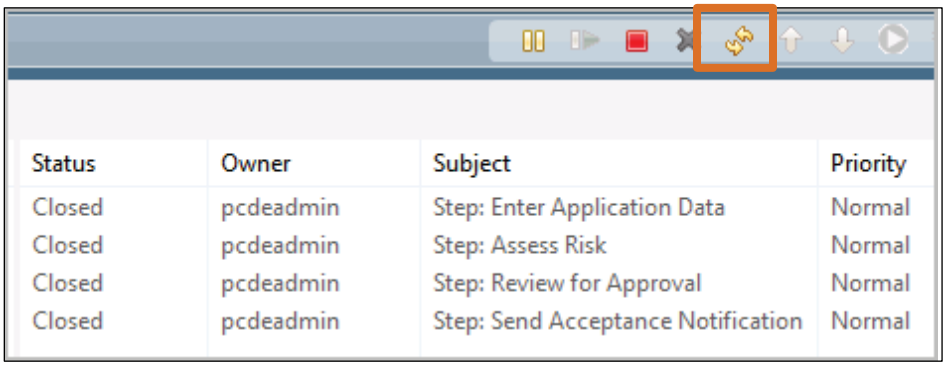
14. In the Select User window, select pcdeadmin and click **OK**.
15. When the coach opens, click **Submit** and then close the browser window.
16. Refresh the process state again as you did before. A third task is **Review for Approval**.
17. Select it and click **Runs the selected task**.



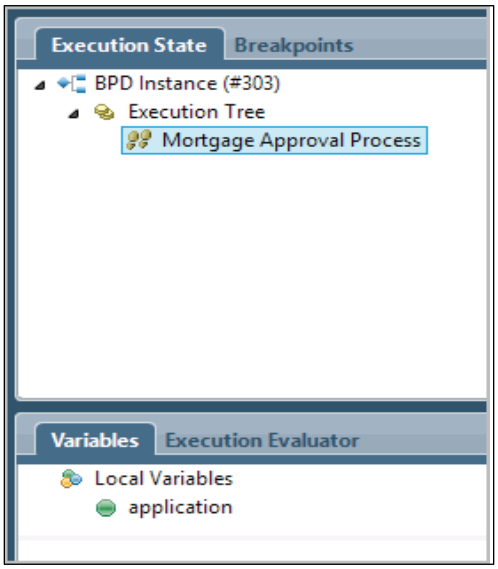
18. In the Select User window, select pcdeadmin and click **OK**.
19. In the coach, this time, make sure that you switch to the Final Decision tab and enter ACCEPT.



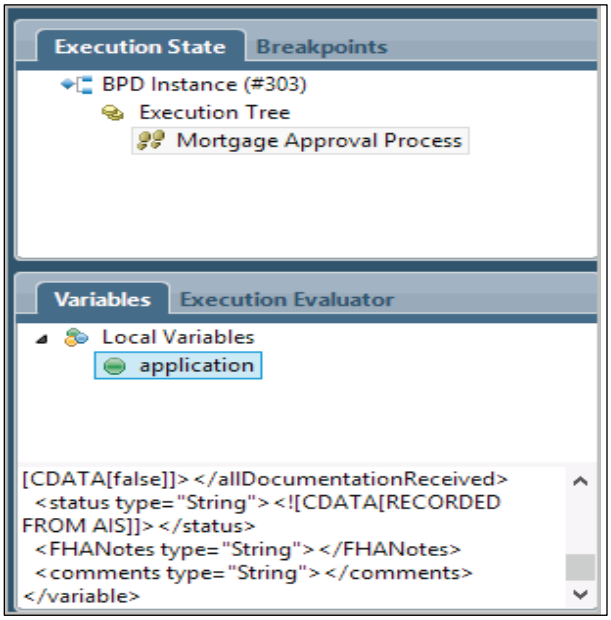
20. Click **Submit** and close the browser.
21. In the Inspector, click the refresh icon a few times until the process completes. The stack should look similar to the one that follows (if you clicked refresh several times and the status of all the tasks is closed, it is OK if your screen does not match exactly):



22. Under Execution State, select **Mortgage Approval Process** and under the **Variables** tab, double-click **application**.



23. The inspector displays the XML representation of the final application business object: Scroll down in the XML pane. You should notice that although the status was set to “ACCEPT” in the Review for Approval, the Advanced Integration service changed it to “RECORDED FROM AIS”:



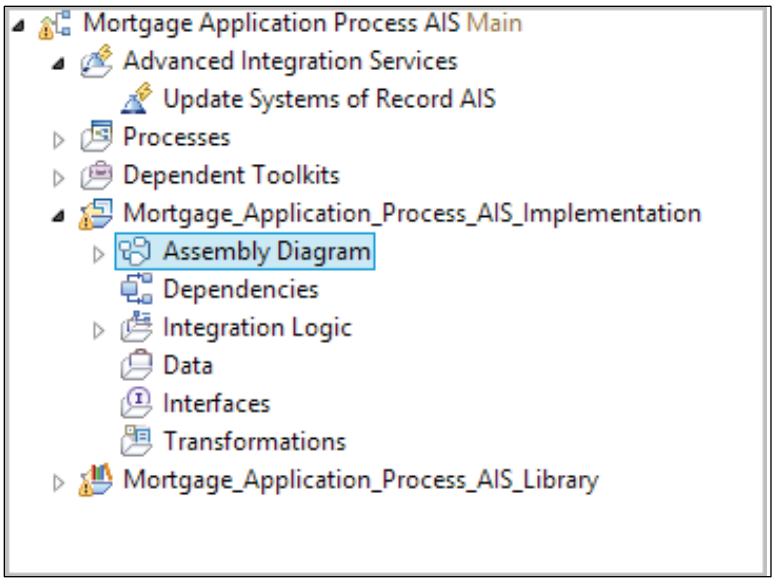
24. Close Mortgage Approval Process by clicking X.
25. Click Process Center at the upper right corner to switch to the Process Center view.
26. This portion of the test is successfully concluded, as it shows the invocation of AIS from a running BPD and a coach in IBM Process Designer.

Part 6. Implement the revised AIS in IBM Integration Designer

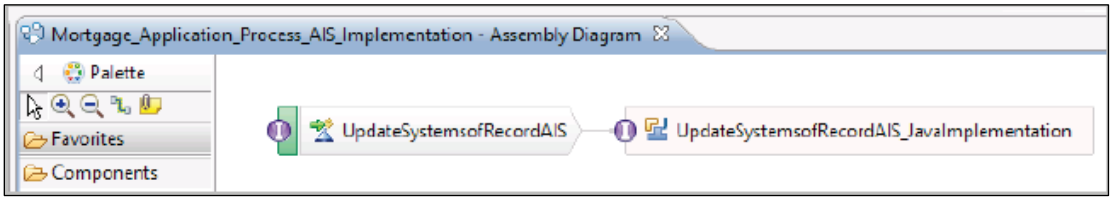
In the previous section, you created a dummy implementation of an AIS to prove the concept of being able to invoke integration artifacts from BPMN processes. The integration artifacts can be complex service orchestrations, with mappings and adapters. In this section, you take the implemented Advanced Integration service and reimplement it as a BPEL business process. As soon as you understand how this solution, including coach, BPD, AIS, and BPEL is implemented, you can use this scenario to customize and create several different complex integrations.

1. Use IBM Integration Designer to reimplement the AIS.

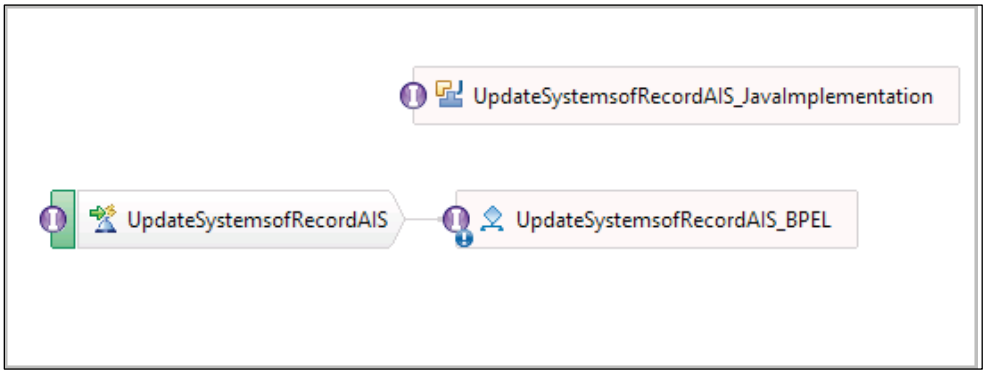
1. Minimize IBM Process Designer.
2. Maximize the IBM Integration Designer workspace that you worked on in the previous section.
3. In Business Integration, expand **Mortgage_Application_Process_AIS_Implementation** and double-click **Assembly Diagram**.



4. The Assembly Diagram shows two components: An export (that the outside world uses to call the integration) – UpdateSystemsofRecordAIS, and the Java Implementation that you created in earlier portion of this exercise - UpdateSystemsofRecordAIS_JavaImplementation.



5. You replace the Java implementation with a BPEL process. To create a BPEL Process skeleton and implement it, drag the Process component from the palette to the canvas. Rename the Component to **UpdateSystemsofRecordAIS_BPEL**.
6. Disconnect the export from the Java Implementation and connect it to the **UpdateSystemsofRecordAIS_BPEL** process.
7. Click **OK** when presented with the option to create a matching interface on the target BPEL process. It associates the same interface (business object application and operation invoke) that is used by the Java implementation.
8. Notice an exclamation point next to the BPEL process, which means that it is not implemented yet.



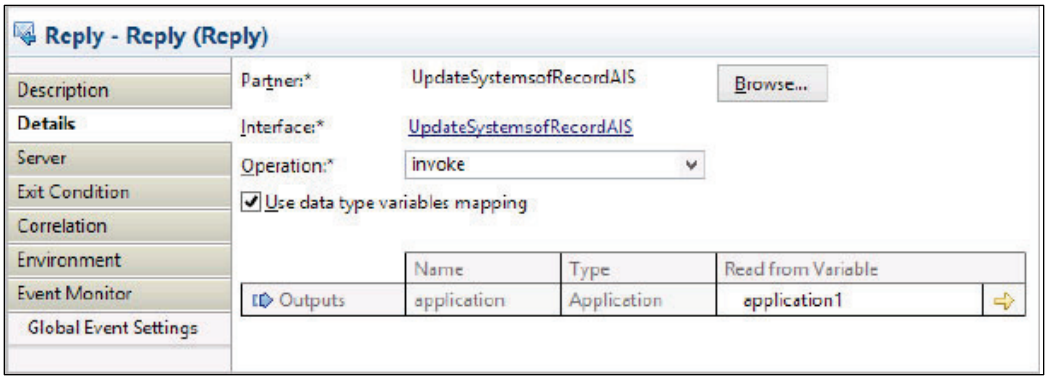
9. Double-click the BPEL process to open it in the BPEL editor. Click **Yes** at the Open window.
10. In the Generate Implementation window, click New Folder and specify com/ibm/ais for the target folder.
11. Click OK to confirm the new folder, and then OK to complete the operation.

2. Continue with the implementation of the BPEL Process.

1. Wait until the workspace is fully built. The BPEL Process diagram is displayed when this workspace build is completed. The BPEL Process diagram is displayed top to down (while the BPMN in the Process Designer is displayed left to right), and starts off with a Receive and Reply activities.
2. Right-click the green circle at the top of the process diagram, and then click **Show in > Properties View**.
3. Click **Details** to open the general properties for this process at the bottom of the screen. Notice that, by default, this process is configured to be a long-running process.

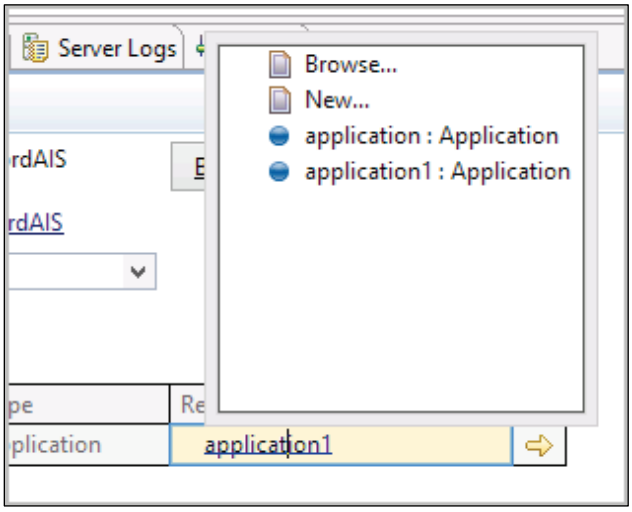
Long-running processes are useful when process state needs to be preserved for a longer time than the span of a single transaction. In this exercise, because the process needs to record the transaction to one or more systems of record, this state does not need to be preserved. You can make this process a microflow, which has better performance characteristics than a long running process.

4. Click **Refactor to microflow**.
5. If you are prompted to save the outstanding changes, click **OK**.
6. In the Change BPEL process run mode window, click **Refactor**.
7. Notice that the process type is now **Microflow**.
8. In the process editor, select the **Receive activity**.
9. In the **Properties** view that follows the diagram, switch to the **Details** tab.
10. Notice the invoke operation and input parameter (application) that was specified in the earlier part of this exercise.
11. Notice the name of the BPEL variable, which maps to the input business object (application).
12. Click the Reply activity.
13. In the **Properties** view that follows the diagram, click the **Details** tab.
14. Notice at the operation and output parameter (application) that was specified in the earlier part of this exercise. The Output that is named application receives the data from the variable application1, which is also a local variable.



The reason why the input and output variables are different is because in many cases – you might want to preserve the original state of the input while you are working on the output. But in this exercise it is not needed. You can change the input variable directly, and return it as output.

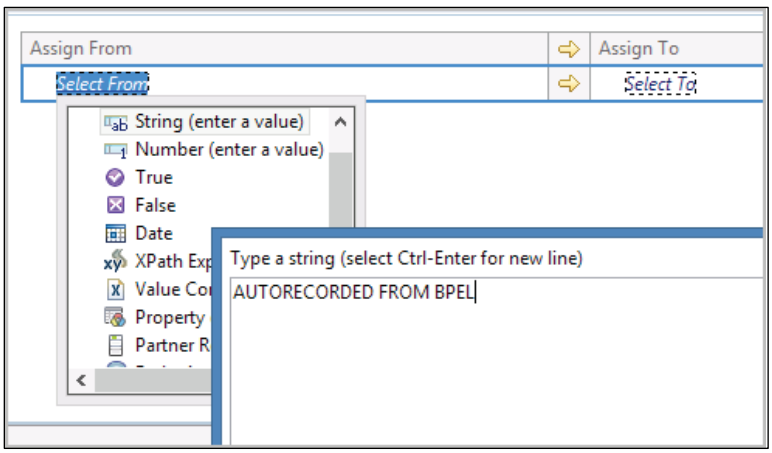
15. Click **application1** and then select application from the list.



16. Verify that the Read from Variable value changes to application.
17. You can eliminate the extra BPEL variable application1. On the right side of the main editor pane, locate the Variables drawer, select application1, and then click the red X to delete it:
18. Save your work (Ctrl-S) and wait for the build to finish.

3. Add an assign activity that sets the mortgage application status.

1. Place an Assign activity between Receive and Reply:
2. Change the display name of the activity to: **Set Autorecorded**
3. Select **Set Autorecorded** and then in the Properties view that follows, switch to the Details tab.
4. Click Select From and then select String (enter a value). Type AUTORECORDED FROM BPEL in the Type a string field.



5. Press the Enter key to complete and then click **Select To**.
6. Expand the application and then select status.
7. Verify that the assigned values are correct.

8. Save your work.

4. Add an invoke activity that does the recording. Normally, BPEL processes invoke a number of external services, by using web services, JMS, adapters, and other protocols.

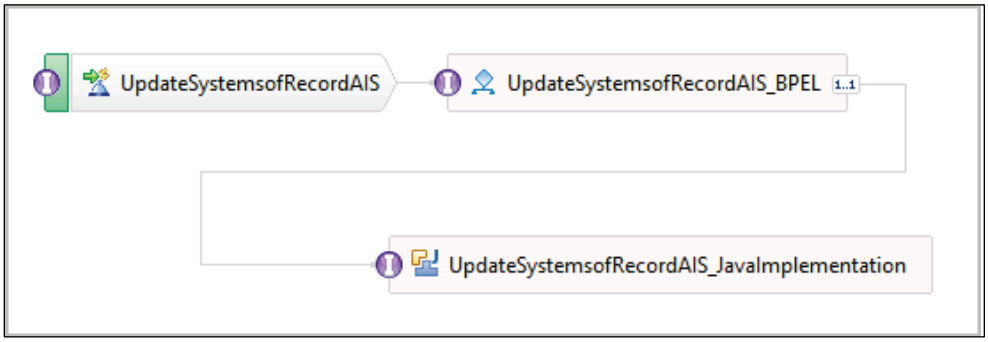
In this exercise, you are going to invoke a single service. For the service, you implement the POJO component.

1. In the process editor, add an invoke activity just above the Reply activity.
2. Change the display name of the new invoke activity to: **Invoke POJO**
3. Invoke operations need a “Reference Partner” to invoke. The reference partner represents the external service that is being invoked. On the right side, click the plus sign next to Reference Partners. For Name, enter POJO and then select the UpdateSystemsofRecordAIS interface under Matching Interfaces and click OK.

   The partner gets created.
4. Click the Invoke POJO operation that you just added.
5. In the Properties view, click the **Details** tab. Then, click Browse next to Partner. Select POJO and click **OK**.
6. For the Inputs, click none and then select application.
7. For the Outputs, click none and then select application.
8. The BPEL process is now complete. This process sets the mortgage status to AUTORECORDED FROM BPEL and then calls a service that does the actual recording.
9. Save your work. (Ctrl+S), and close the BPEL editor.

5. Wire the BPEL process to the POJO service.

1. Go back to the Assembly Diagram. Notice the error indication by the BPEL process component.
2. The error exists because you manually added a partner reference, which is not yet reflected on the diagram. Right-click the BPEL process component and click Synchronize Interfaces and References > from Implementation.
3. Click Yes in the confirmation dialog. Notice that now the BPEL process has a reference icon and the error goes away.
4. You wire the reference to the target service, hover over the reference icon so that a connector appears. Then, click and drag it to the input interface of the POJO component. The connection is made, and the result looks similar to the following image:



5. Save your work.
6. Double-click the POJO component to open up the implementation. The Java editor opens up.
7. Scroll down to the bottom and locate the invoke object. You implemented Java code earlier in the exercise.
8. Comment out: application.setString("status", "RECORDED FROM AIS");
9. Change the text of the print statement by replacing Java with: BPEL
10. Now the Java POJO does not do anything other than writing to the log file. Save your work.
11. Save your changes and close the Java editor.

Part 7. Unit Test the updated AIS and the new BPEL component in IBM Integration Designer

1. Publish your changes to Process Center.

1. Verify that Mortgage Approval Process has [changed] next to it.
2. Right-click [changed] Mortgage Approval Process Main, and click Publish to Process Center.

2. Run the AIS test.

1. Make sure that the process server is running.
2. In Business Integration, expand **Mortgage Application Process AIS Main > Advanced Integration Services**, then right-click **Update Systems of Record AIS** and then select **Test Implementation**.
3. The Integration Test Client starts. In the Initial request parameters to the right, scroll through the value editor and locate the status field under the Name column.
4. Enter APPROVED in the value column for status.
5. Under Events, click the **Continue** icon to start the Test.

6. In the Deployment Location window, expand IBM Process Servers and select **IBM Process Server v8.6 at localhost**. Click **Finish**.
7. In the User Login - Default Module Test window, accept the default values and click OK.
8. It takes a few minutes, after which the project is deployed to the test server and runs the AIS implementation. You see a trace of all the components that are being executed, and a result. In the Events section, click Return.
9. In the return parameters section to the right, notice the value of status. It was earlier set to APPROVED, and it is now updated to AUTORECORDED FROM BPEL.
10. Review the events stack trace. It shows that the BPEL process is being executed too.
11. You tested the revised AIS successfully. You also successfully tested the BPEL process.

Part 8. Run the BPD in IBM Process Designer and execute the revised AIS and invoke the new BPEL component

In the previous section, you tested and confirmed that the Advanced Integration service is working as expected. In this section, you take the completed AIS and execute from the BPD that used the AIS to invoke a BPEL process. You use the Process Inspector to walk through the business process and execute the Advanced Integration service.

1. Test the end-to-end business process (including the AIS) in Process Designer.

1. Minimize Integration Designer, and switch back to IBM Process Designer.
2. If you are in the Process Designer view, then go to the Process Center view by clicking the Process Center icon at the upper right.
3. In the **Process Apps** tab, click **Open in Designer** next to **Mortgage Application Process AIS**.
4. Click **Processes** and then double-click **Mortgage Approval Process**.
5. Click the **Run Process** icon in the upper right corner to run the Mortgage Approval Process:
6. If prompted, click **Yes** to switch to the Inspector view to see the progress of the process instance.
7. The Inspector opens and the focus is on the running process instance. Select the task on the upper right pane (Enter Application Data), and click **Runs the selected task**.
8. In the Select User window, select pcdeadmin and click **OK**.

9. After few minutes, the browser opens with the first coach of the mortgage approval process. All fields are optional for this exercise and you can leave them empty. Click Submit to complete the task.
10. Close the browser window.
11. Go back to Process Designer and click the refresh icon.\
12. The second step on the stack is **Assess Risk**. Select it and click **Runs the selected task**.
13. In the Select User window, select pcdeadmin and click **OK**.
14. When the coach opens, click **Submit** and then close the browser window.
15. Refresh the process state again as you did before. The third task is Review for Approval.
16. Select it and click **Runs the selected task**.
17. In the Select User window, select pcdeadmin and click **OK**.
18. In the coach, this time, make sure that you switch to the **Final Decision** tab and enter: ACCEPT
19. Click **Submit** and close the browser.
20. In the Inspector, click the refresh icon a few times until the process completes. The stack should look similar to the following image:
21. Under **Execution State**, select **Mortgage Approval Process** and under the **Variables** tab, double-click **application**.
22. The inspector displays the XML representation of the final application business object: Scroll down in the XML pane as needed. You should notice that although the status was set to “ACCEPT” in the Review for Approval, the Advanced Integration service changed it to “RECORDED FROM BPEL”:
23. Click Process Center at the upper right corner to switch to the Process Center view.
24. This portion of the test is successfully concluded, as it shows the invocation of AIS and then BPEL from a running BPD and a coach in IBM Process Designer.

2. Close Process Designer.

3. Stop the IBM Process Center server.

1. Double-click the Stop Process Center cluster icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
2. Double-click the Stop Process Center node agent icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
3. Click the Stop Process Center deployment manager icon on the desktop. Wait until it stops successfully. Press any key to close the command window.
4. Stop the IBM Process Server unless it is already stopped.

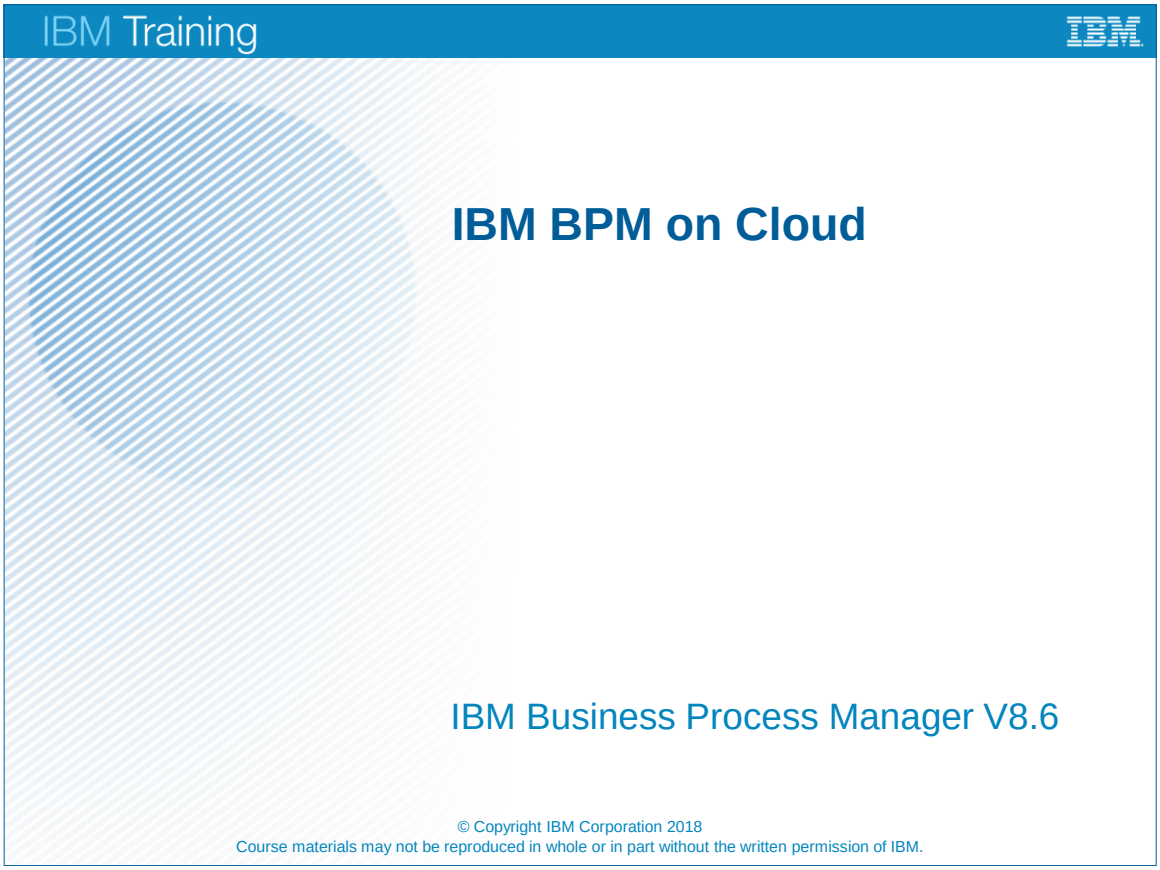
5. Close IBM Integration Designer.
6. Close any other open windows.

Exercise review and wrap-up

You successfully completed the Advanced Integration Lab. As you saw, the flow of a business process from the Process Designer to Advanced Integration in the Integration Designer is seamless. The capabilities of Advanced Integration provided by IBM Business Process Manager Advanced open a whole world of service orientation. These capabilities include access to process choreography through BPEL, adapters, easy incorporation of Java, and the tools that are needed to test every part of the solution.

Though the top-down approach is the most common approach, the other common approach is a bottom-up approach. This process is also straightforward, where an Integration Designer creates Advanced Integrations and publishes them to the Process Center, and they are made visible to the Process Designer to use in a Business Process Definition.

It is important to always test and take small incremental function steps. If you combine this practice with frequent playbacks, you find that it not only shortens the development lifecycle, but also the stakeholders and business users continue to be involved and invested in the application development process. This involvement allows for better communication and quick agreement on what the requirements are. If you have any more questions on any of these labs, contact your IBM Technical Professional.

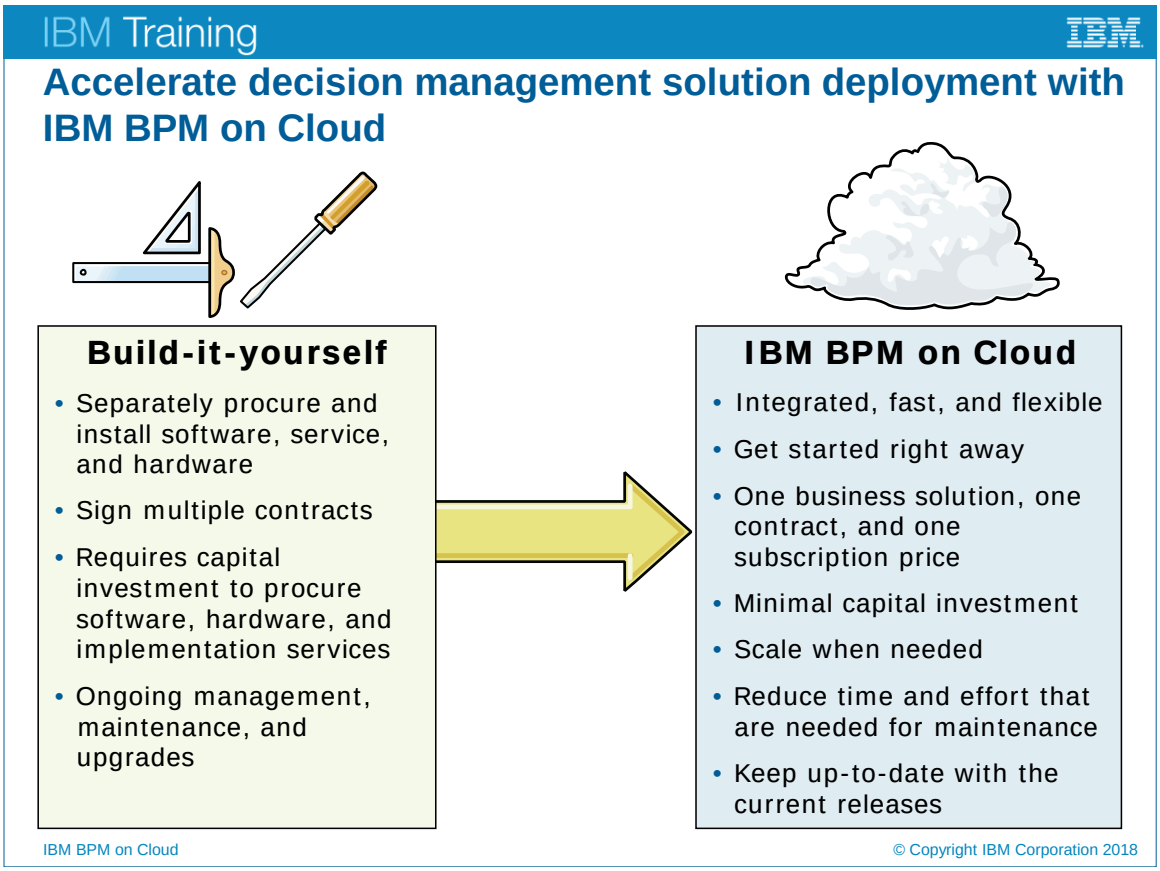

IBM Training
IBM
IBM BPM on Cloud
IBM Business Process Manager V8.6
© Copyright IBM Corporation 2018
Course materials may not be reproduced in whole or in part without the written permission of IBM.

IBM Training

Introduction to IBM BPM on Cloud

- Enterprise-grade BPM cloud service for development, testing, and production
- Cloud-based, collaborative, role-based environment
  - Capture, automate, and manage frequently occurring, repeatable rules-based business decisions
- Ready-to-use development, test, and production environments are available
- Monthly subscription plans
- Available exclusively on IBM Cloud infrastructure
  - As of 2015, over 25 data centers are available worldwide
- Managed by IBM
- Artifacts that are created with IBM BPM on Cloud are compatible with IBM BPM on-premises product

Introduction to IBM BPM on Cloud



Accelerate decision management solution deployment with IBM BPM on Cloud

IBM Training

BPM on Cloud customer focus: Manage and automate decisions

IBM manages:

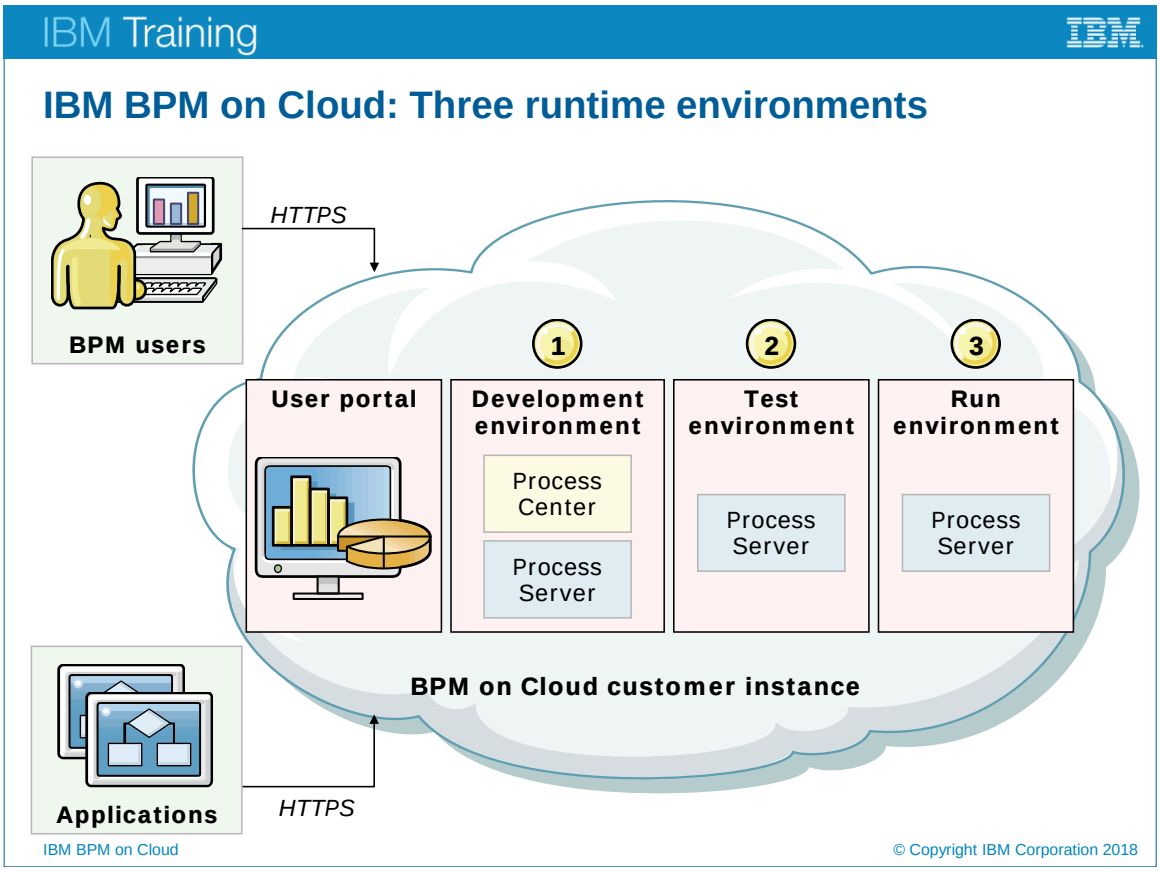
- Uptime
- Monitoring
- Backup
- High availability
- Disaster recovery
- Updates
- Maintenance

Customers manage:

- Application development
- Application integration
- Application support

IBM BPM on Cloud

BPM on Cloud customer focus: Manage and automate decisions



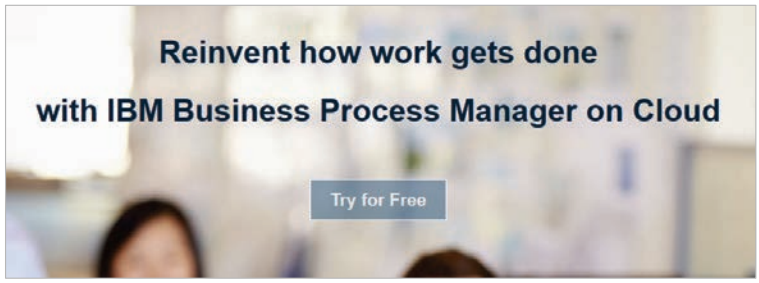
IBM BPM on Cloud: Three runtime environments

IBM Training

IBM BPM on Cloud free trial

- Free 30-day trial for IBM BPM on Cloud is available
- Go to the following website and click **Try for free** to sign up:

  https://www.bpm.ibmcloud.com/#home



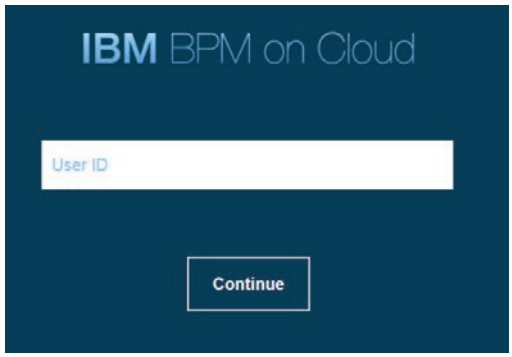
IBM BPM on Cloud

IBM BPM on Cloud free trial

IBM Training

Activating access and logging in to IBM BPM on Cloud

- Welcome email includes the following information:
  - Link to activate BPM on Cloud access
  - Link to BPM on Cloud instance
- Activation link is tied to a specific email
- After activating access, you can log in to your BPM on Cloud instance

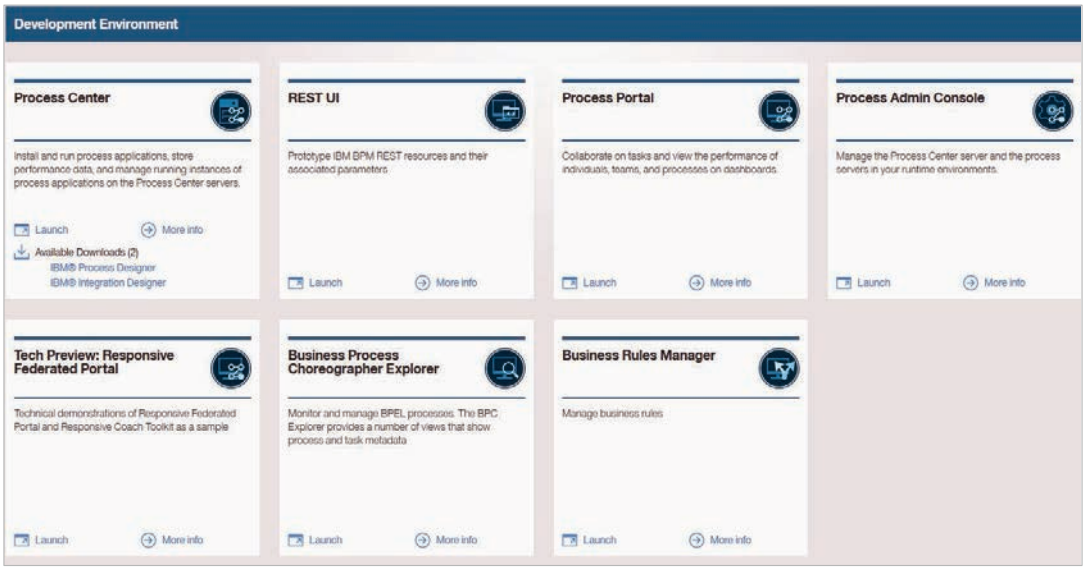


Activating access and logging in to IBM BPM on Cloud

IBM Training

IBM BPM on Cloud user portal (1 of 3)

- Access from home page to an array of tools in three environments:
  - Development



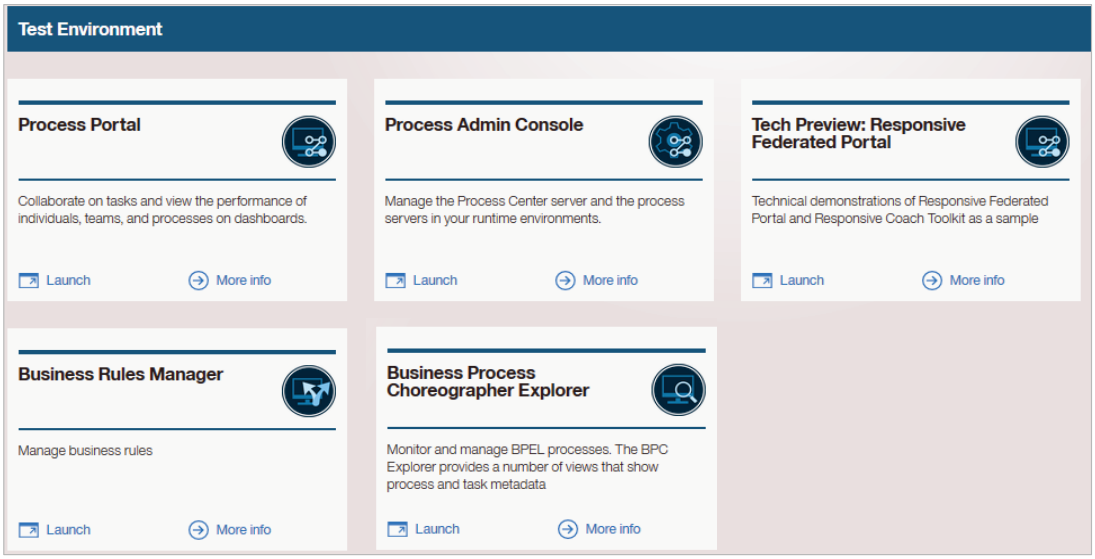
IBM BPM on Cloud

IBM BPM on Cloud user portal

IBM Training

IBM BPM on Cloud user portal (2 of 3)

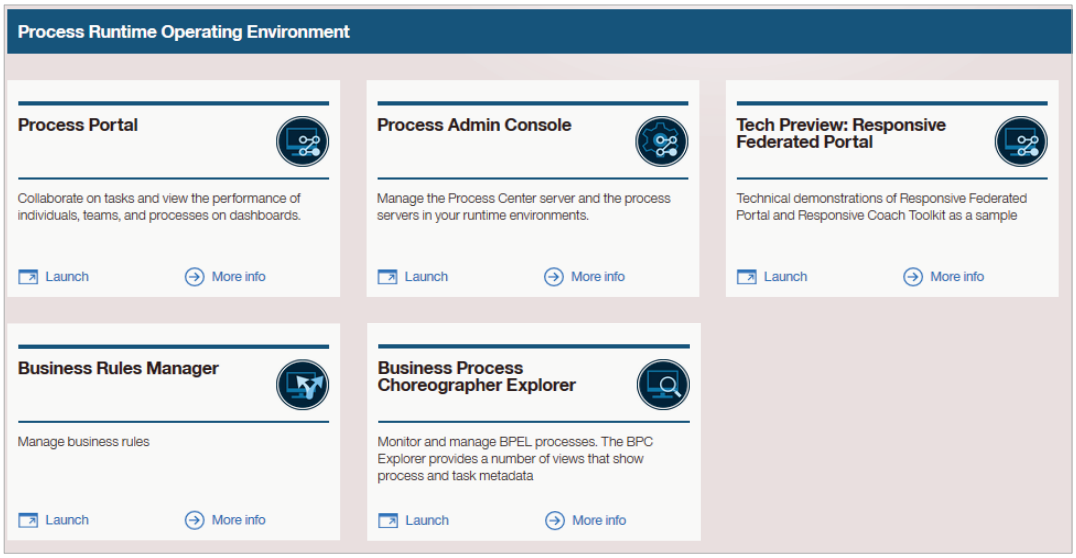
- Access from home page to an array of tools in three environments:
  - Test



IBM Training

IBM BPM on Cloud user portal (3 of 3)

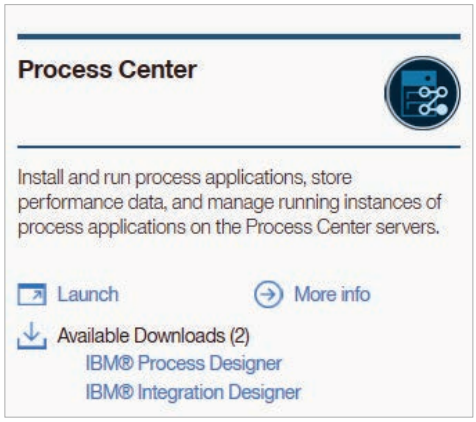
- Access from home page to an array of tools in three environments:
  - Production Runtime Operating



IBM Training

Using the IBM Process Designer (1 of 3)

- Download a version of Process Designer that is configured for use with IBM BPM on Cloud
- Start Process Designer by double-clicking `eclipse.exe`

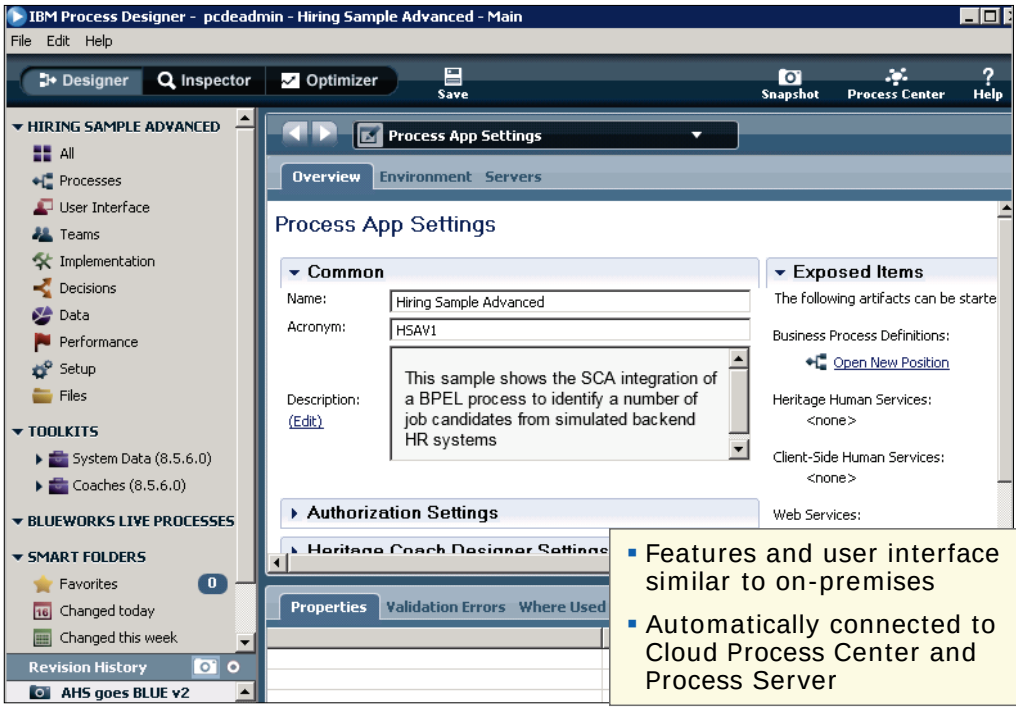


IBM BPM on Cloud

Using the IBM Process Designer

IBM Training

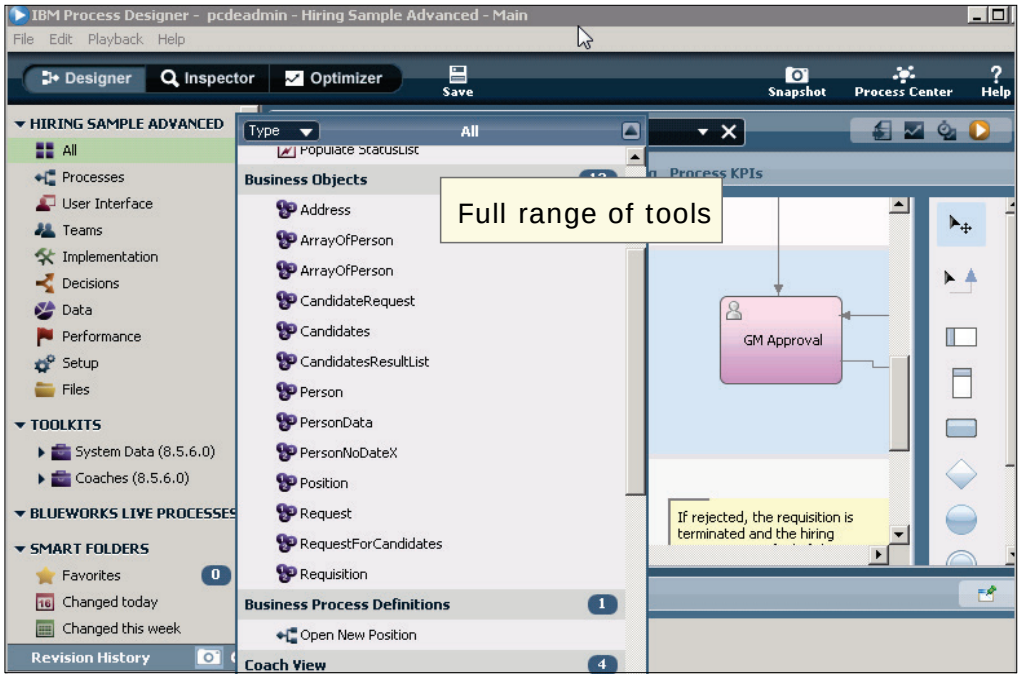
Using the IBM Process Designer (2 of 3)



- Features and user interface similar to on-premises
- Automatically connected to Cloud Process Center and Process Server

IBM Training

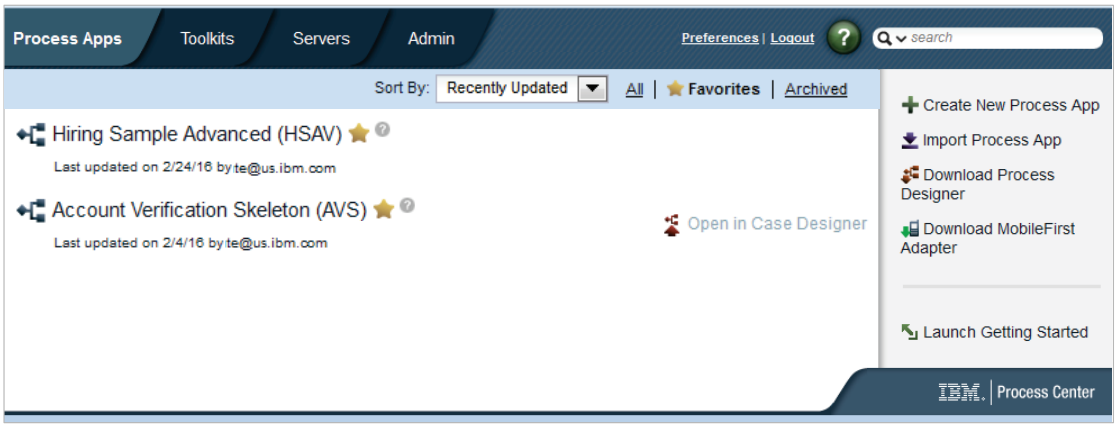
Using the IBM Process Designer (3 of 3)



IBM Training

Using the IBM Process Center (1 of 3)

- Familiar interface

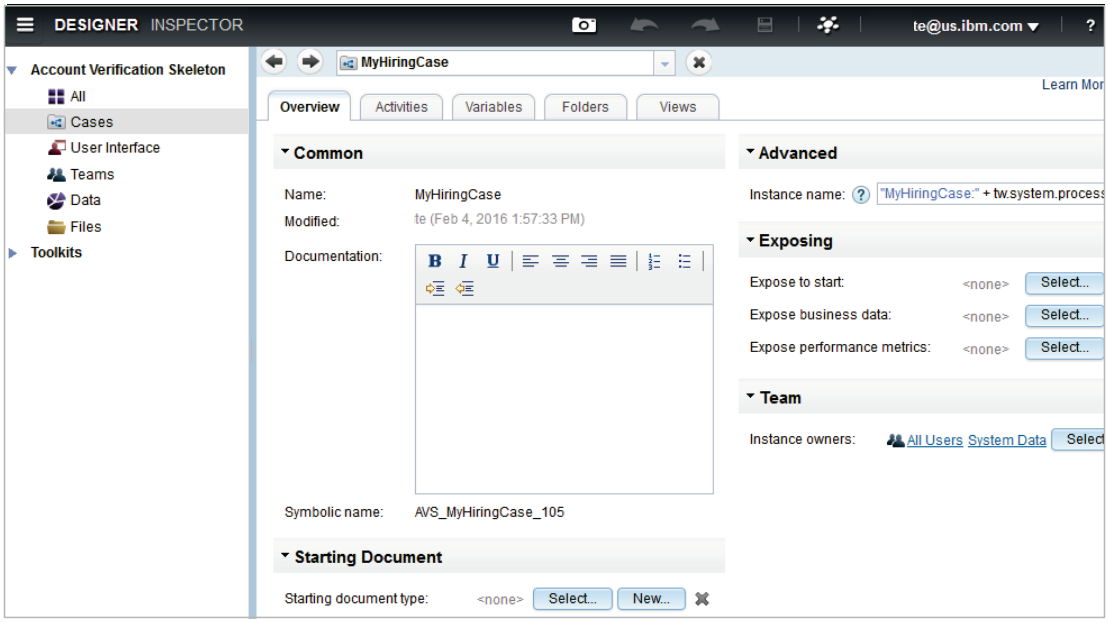


Using the IBM Process Center

IBM Training

Using the IBM Process Center (2 of 3)

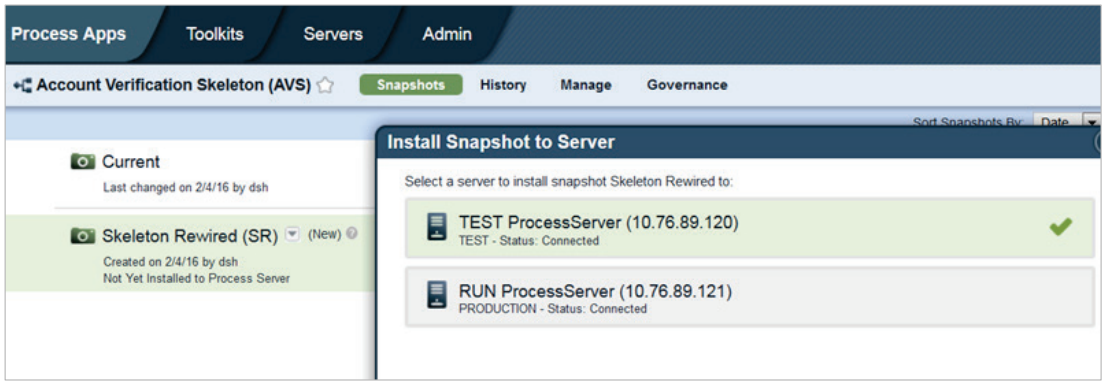
- Cloud-based Case Designer

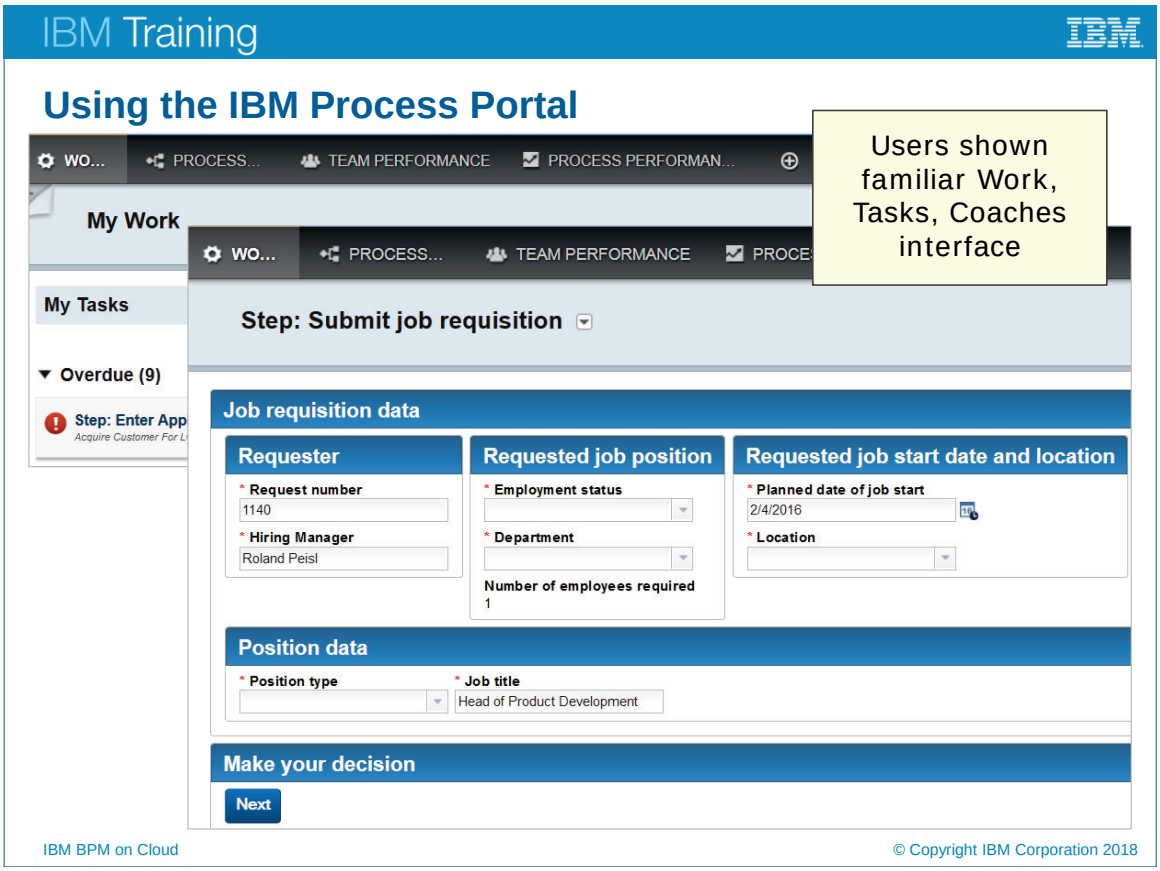


IBM Training

Using the IBM Process Center (3 of 3)

- Snapshots, export, install familiar



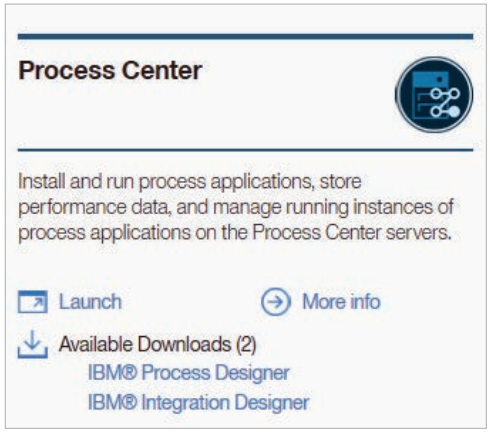


Using the IBM Process Portal

IBM Training

Using the IBM Integration Designer

- Download a version of the IBM Integration Designer from the BPM cloud:
  - URL for connecting to IBM BPM on Cloud provided
- Start IBM Integration Designer on local workstation as usual



IBM BPM on Cloud

Using the IBM Integration Designer

IBM Training

Finding help for IBM BPM on Cloud

- IBM Knowledge Center for IBM BPM on Cloud
  - http://www.ibm.com/support/knowledgecenter/SS964W/ditamaps/product_welcome_oncloud.html
  - Complete product documentation for IBM BPM on Cloud, including a “Getting Started” tutorial
  - IBM BPM on Cloud user portal also has direct links to the documentation
- IBM BPM Support Portal
  - https://www.ibm.com/support/entry/portal/product/websphere
  - Support Portal provides tools and resources for help with IBM Business Process Manager
  - Open service requests, view fix lists, access community resources, and more

Finding help for IBM BPM on Cloud

IBM Training