Chapter 8

Configuring and testing dynamic assembly

The last chapter showed how to use Dynamic Assembler components in place of hard-wired imports. This chapter will complete the configuration of these components and test your policies using the Business Services Perspective in WebSphere Integration Developer.

Composition Studio is an extension of WebSphere Integration Developer that contributes the Business Service Perspective. The tools are designed to allow several users to collaboratively edit configuration that is maintained as Fabric projects on the server. There are a few things to keep in mind when working with Fabric projects:

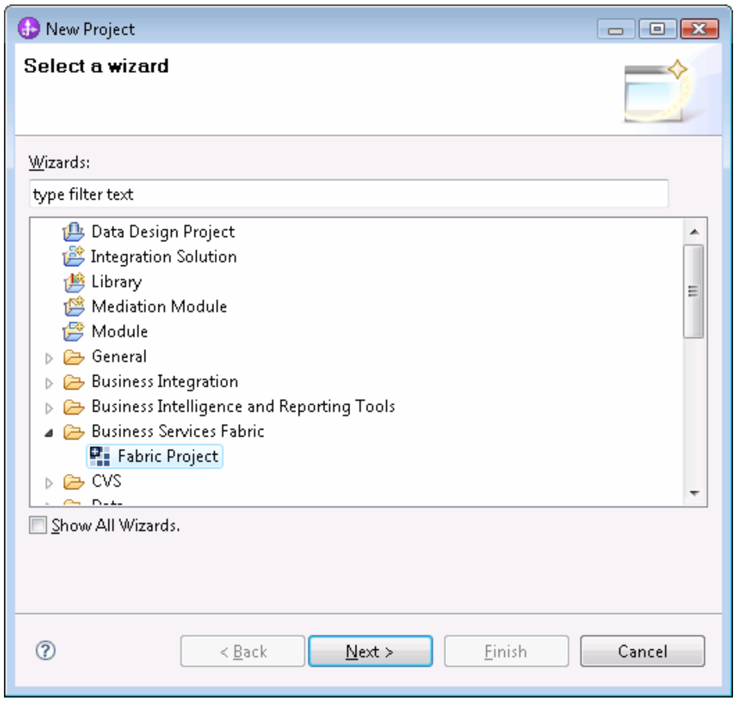
- A Fabric Server needs to be running to create, update, or submit changes to a project.
- Since a local Fabric project represents only a user’s connection and local changes to a specific server, it is not useful to export or share a Fabric project. Submit your changes and export the data you want to share from the server environment instead.
- A workspace can only connect to one Fabric Server at a time. Fabric projects will not work if imported into a workspace other than the one they were created in.
- If not working with a Fabric server unit test environment, changes need to be approved by an administrator in Business Space before the changes can be seen by the runtime.

Creating a Fabric project

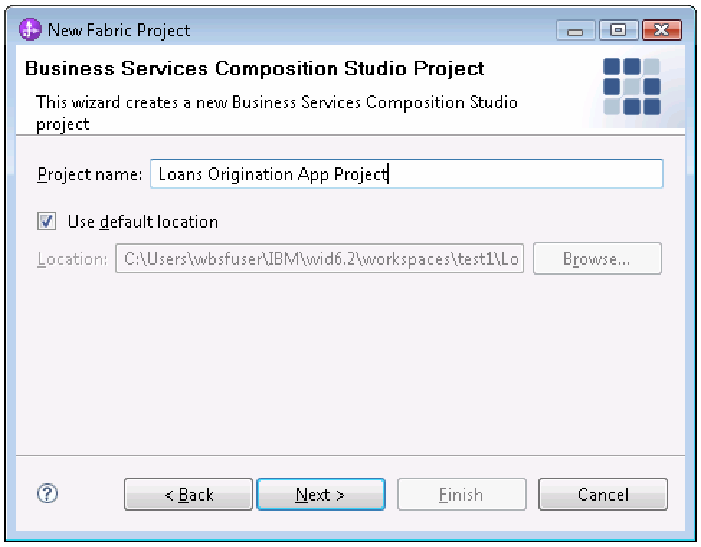
The authoring performed earlier using Business Space caused Fabric projects to be created in Business Space. A Fabric project created for an application automatically imports each of the business services used by that application. You will connect to the “Loans Origination” fabric project.

Note: The first time a user creates a Fabric project, it is necessary to log in to the Fabric web console at http://<server>:<port>/fabric to establish a Fabric user account.
If this step is not performed, the create project wizard will show an empty list of available projects.

1. File -> New -> Project … Select "Fabric Project", click Next.



2. Provide a name for the project. Click Next.



This wizard creates a new Business Services Composition Studio project

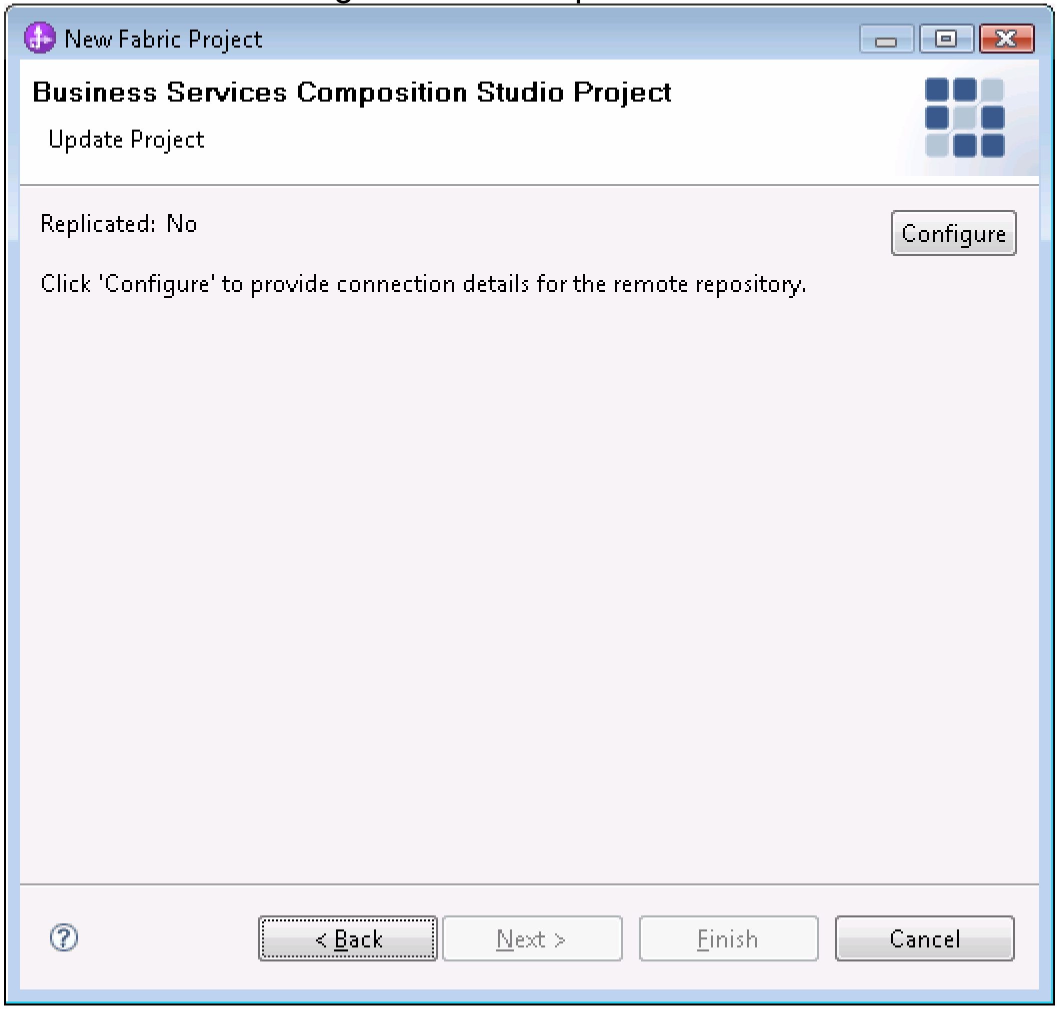
Project name: Loans Origination App Project

[✓] Use default location

Location: C:\Users\wbsfuser\IBM\wid6.2\workspaces\test1\Lo

Browse...

3. Click the configure button to provide server connection details.


New Fabric Project

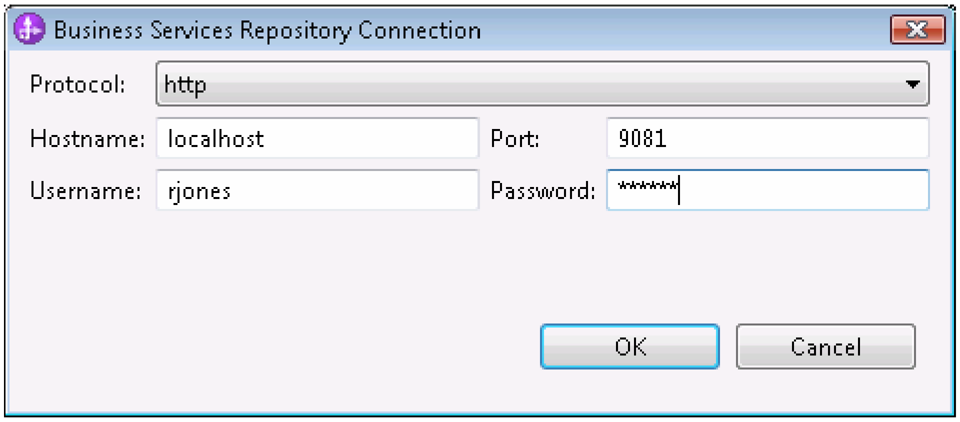
Business Services Composition Studio Project

Update Project

Replicated: No

Click 'Configure' to provide connection details for the remote repository.

4. Provide the server, port, username, and password. Click OK.


Business Services Repository Connection

Protocol: http

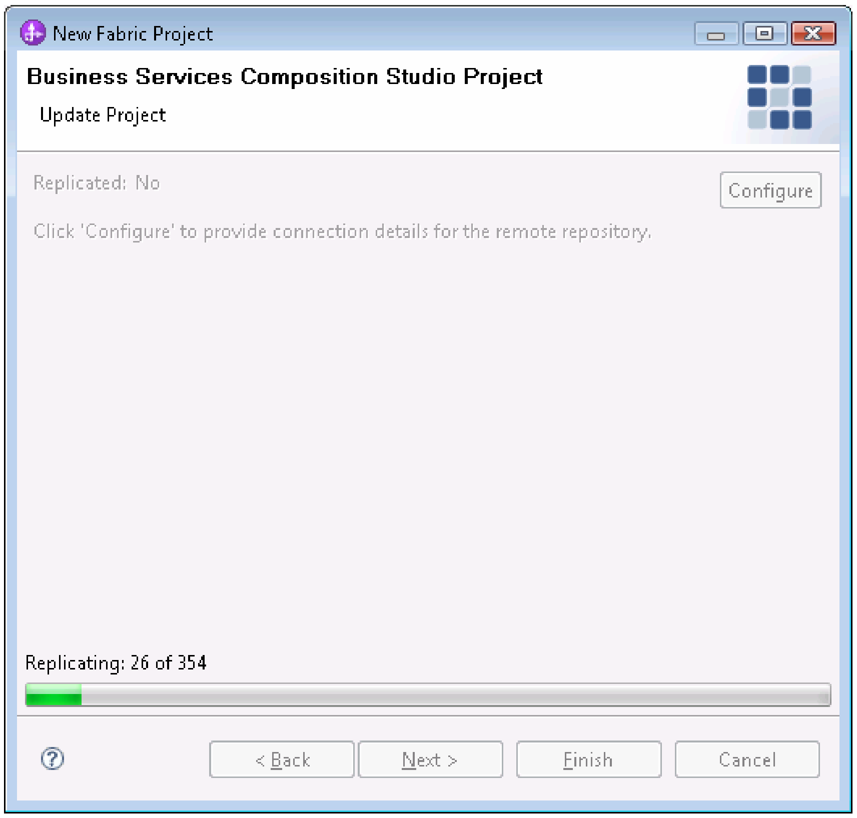
Hostname: localhost

Port: 9081

Username: rjones

Password: *******

5. Wait. After several moments a progress bar will indicate the progress of replication. This operation might take a long time to complete.



Business Services Composition Studio Project

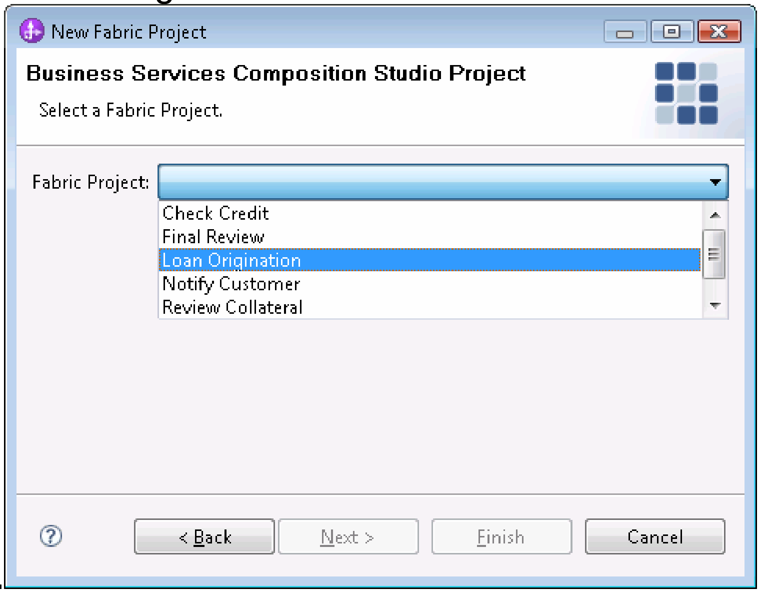
Update Project

Replicated: No

Click 'Configure' to provide connection details for the remote repository.

Replicating: 26 of 354

6. Choose "Loan Origination" from the list of available Fabric Projects.



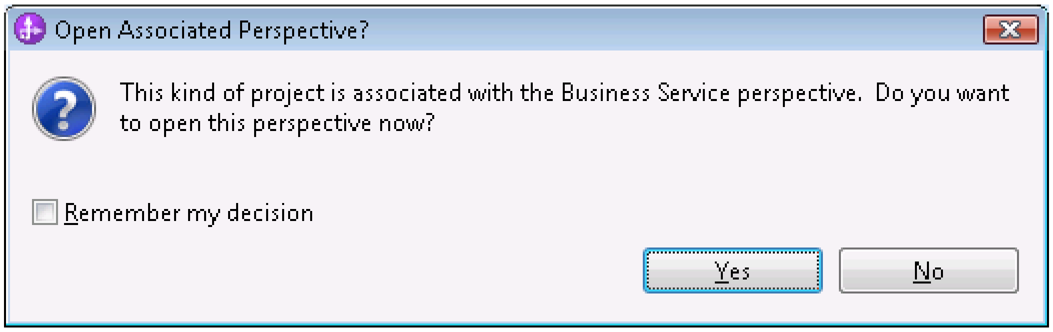
Business Services Composition Studio Project

Select a Fabric Project.

Fabric Project:

Check Credit
Final Review
Loan Origination
Notify Customer
Review Collateral

7. You will see this dialog if you are not already in the Business Service perspective. Click Yes.

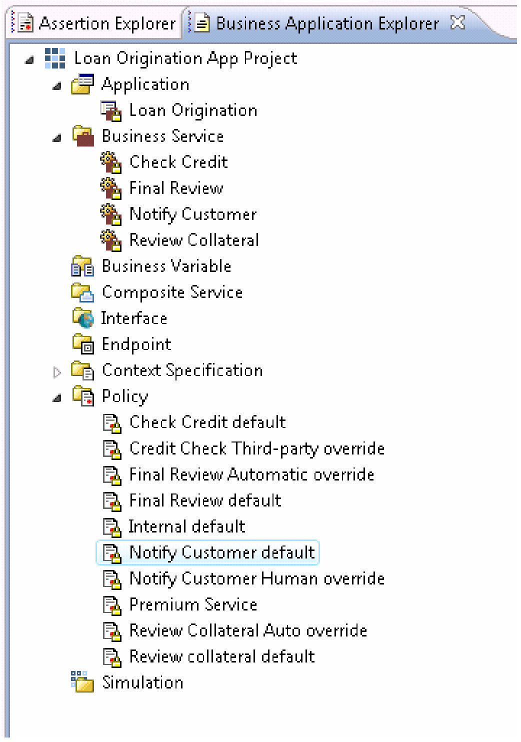

Open Associated Perspective?

This kind of project is associated with the Business Service perspective. Do you want to open this perspective now?

Remember my decision

8. By default, the Business Service Perspective has a few views that are only useful if you are working with projects from WebSphere Business Services Fabric 6.1.2 or earlier releases. Go ahead and close these views if empty:
- Close the Subscriber Explorer view.
- Close the Business Services Explorer view.

Expand the Business Application Explorer. Note that the application, business services, and policies authored in business space show up here with a lock icon indicating that they are read only. These objects can only be modified using the business space.

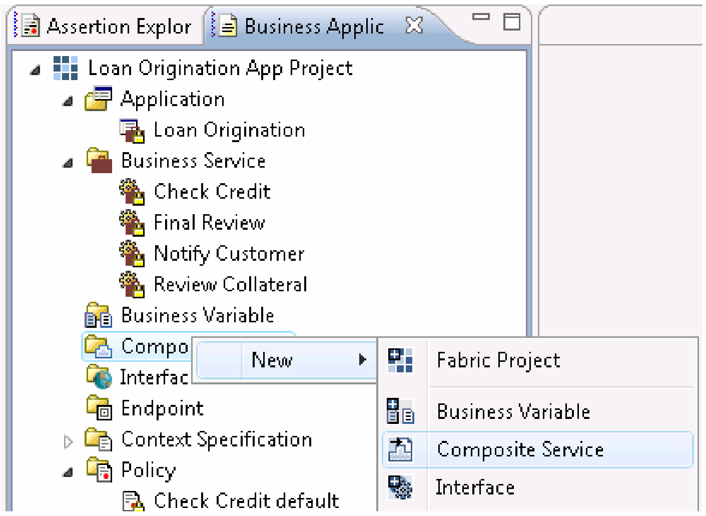


```
Loan Origination App Project
  Application
    Loan Origination
  Business Service
    Check Credit
    Final Review
    Notify Customer
    Review Collateral
  Business Variable
  Composite Service
  Interface
  Endpoint
  Context Specification
  Policy
    Check Credit default
    Credit Check Third-party override
    Final Review Automatic override
    Final Review default
    Internal default
    Notify Customer default
    Notify Customer Human override
    Premium Service
    Review Collateral Auto override
    Review collateral default
  Simulation
```

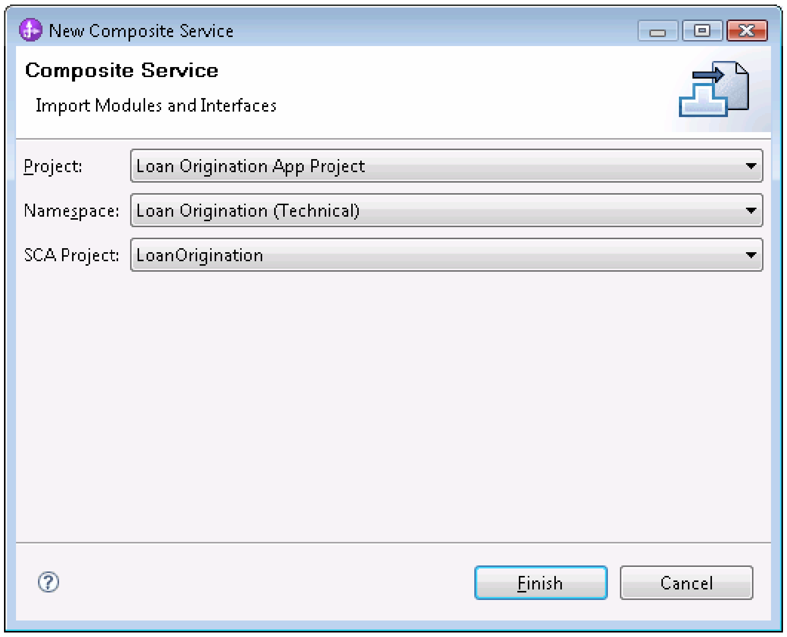
Import SCA modules to create composite service definitions

It is necessary to create composite service definitions to feed the dynamic assembler information about the interfaces, modules, and dynamic assembly components that are going to be deployed. This is done by importing SCA modules that exist in the workspace into Composition Studio.

1. Using the Business Application Explorer, select New -> Composite Service.



2. Select the SCA module to be imported and click Finish. You can start with the module named “LoanOrigination”.


New Composite Service

Composite Service

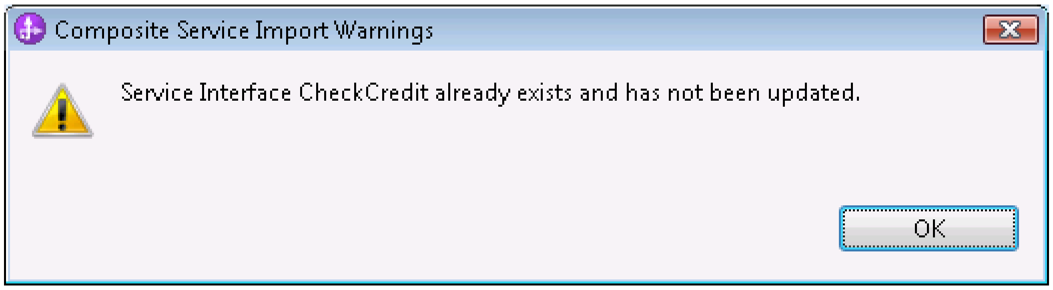
Import Modules and Interfaces

Project: Loan Origination App Project

Namespace: Loan Origination (Technical)

SCA Project: LoanOrigination

The import operation creates a Composite Service object that will show up in the Business Application explorer. It will also create Service Interface definitions for interfaces used by the module. Endpoints are created for SOAP/HTTP or SCA Exports in the modules. Even though multiple modules might use a given service interface, the system does not create duplicates. Instead, it provides warning messages like these:


Composite Service Import Warnings

Service Interface CheckCredit already exists and has not been updated.

Once a composite service is created, changes in the original to the original SCA module can be performed by carrying out the import again. It is important to import a module again if there are changes to its imports, exports, or DA Components.

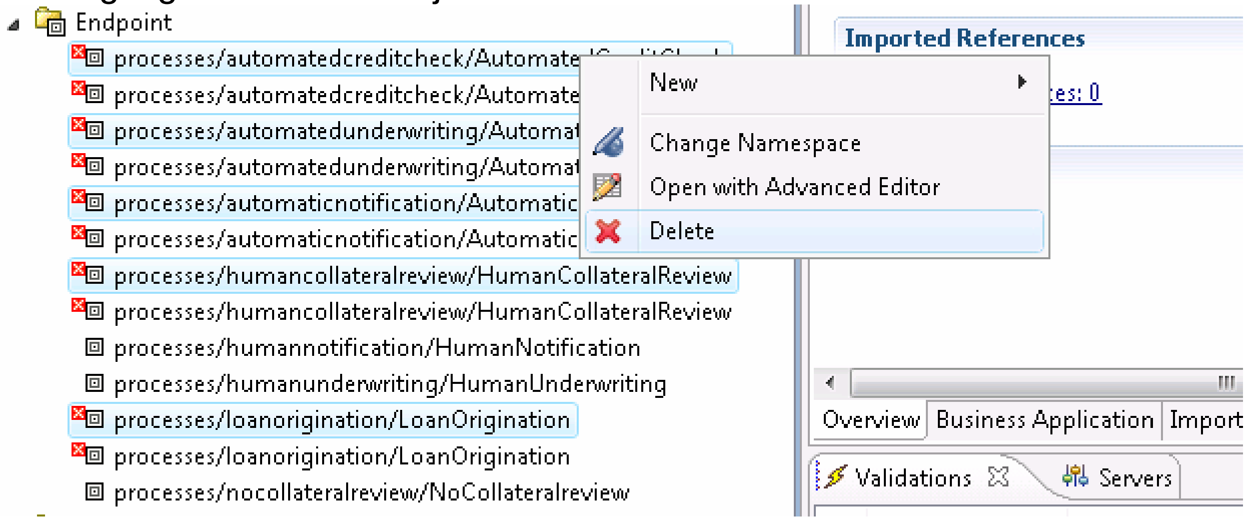
3. For each of the remaining modules, carry out the same import steps.

Module	Description
LoanOrigination	Module contains BPEL that captures

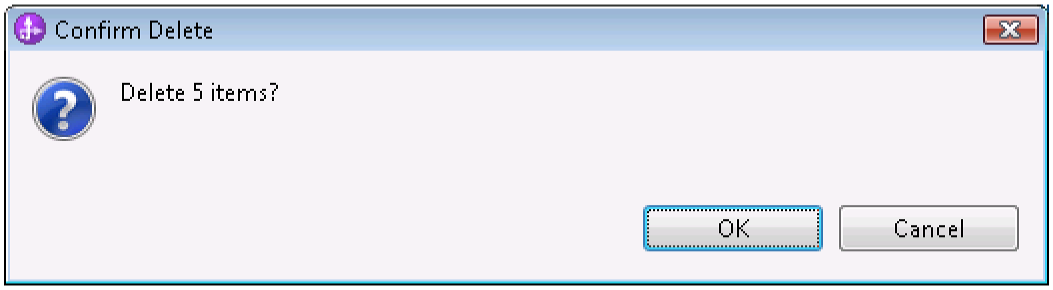
	the main process flow modeled in Business Space. Has several DA Components, with one for each business service used in the flow.
CheckCredit	Uses another dynamic assembler to select among three backend credit services.
ReviewCollateral	Provides implementation of pass-through and human collateral review process variations.
FinalReview	Provides implementations for automated and human underwriting.
NotifyCustomer	Provides implementations for automated and human implementations.

A problem with the import sometimes causes duplicate endpoints to be created. Validation errors are displayed next two endpoints that have a duplicate. The problem can be resolved by selecting one of each duplicate deleting them.

4. Highlight unwanted objects and use the delete menu action.


Endpoint
processes/automatedcreditcheck/AutomatedCreditCheck
processes/automatedcreditcheck/Automate
processes/automatedunderwriting/Automat
processes/automatedunderwriting/Automat
processes/automaticnotification/Automatic
processes/automaticnotification/Automatic
processes/humancollateralreview/HumanCollateralReview
processes/humancollateralreview/HumanCollateralReview
processes/humannotification/HumanNotification
processes/humanunderwriting/HumanUnderwriting
processes/loanorigination/LoanOrigination
processes/loanorigination/LoanOrigination
processes/nocollateralreview/NoCollateralreview

5. Confirm the delete.



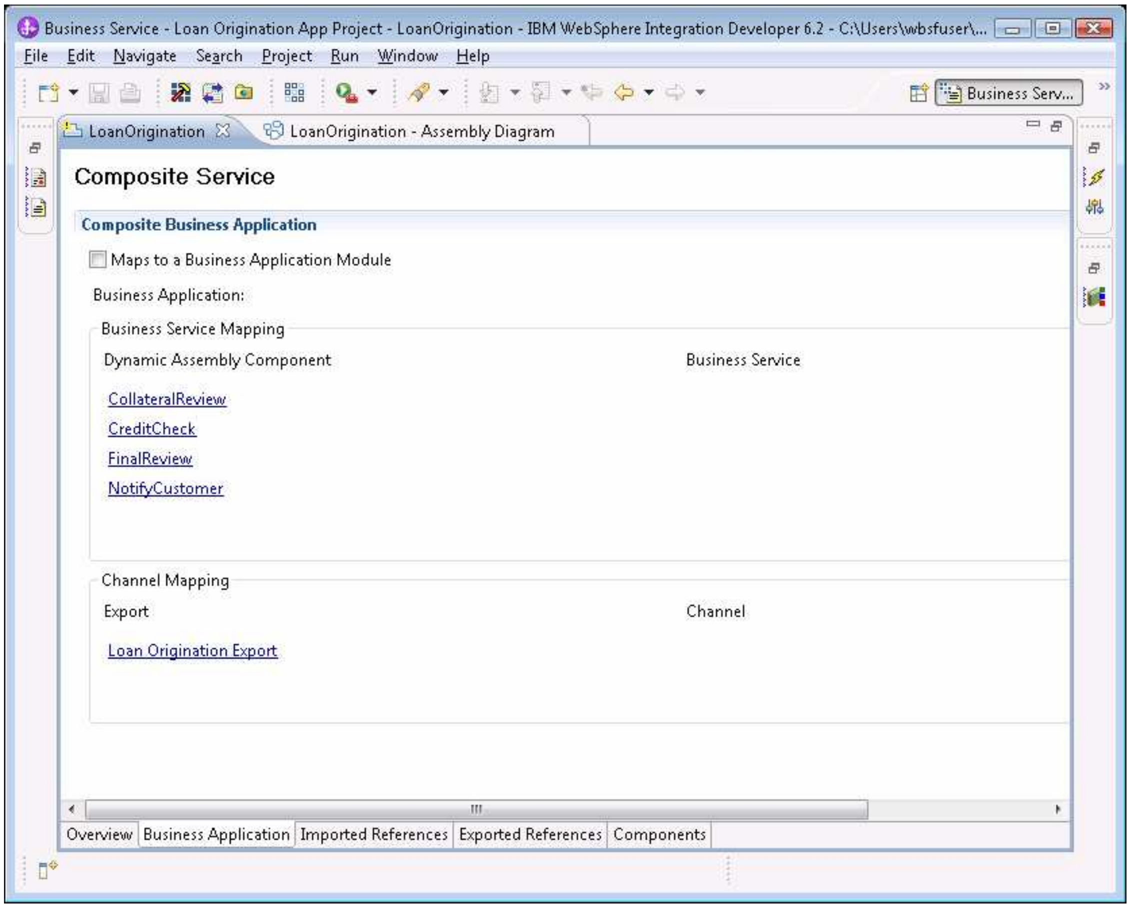
Delete 5 items?

Configuring the Business Application module

When your business analyst authored the application and its business service, she had no exposure to the notion of a dynamic assembly implementation component. And up to this point, nothing was done to associate the SCA modules to the content authored in business space.

In this section, you will perform some mappings that configure the dynamic assembly components to process policies authored in Business Space.

1. Find the composite service named “LoanOrigination”, double click to open the editor, and go to the Business Application tab.


Business Service - Loan Origination App Project - LoanOrigination - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\...

File Edit Navigate Search Project Run Window Help

LoanOrigination | LoanOrigination - Assembly Diagram

Composite Service

Composite Business Application

☐ Maps to a Business Application Module

Business Application:

Business Service Mapping

Dynamic Assembly Component	Business Service
CollateralReview	
CreditCheck	
FinalReview	
NotifyCustomer	

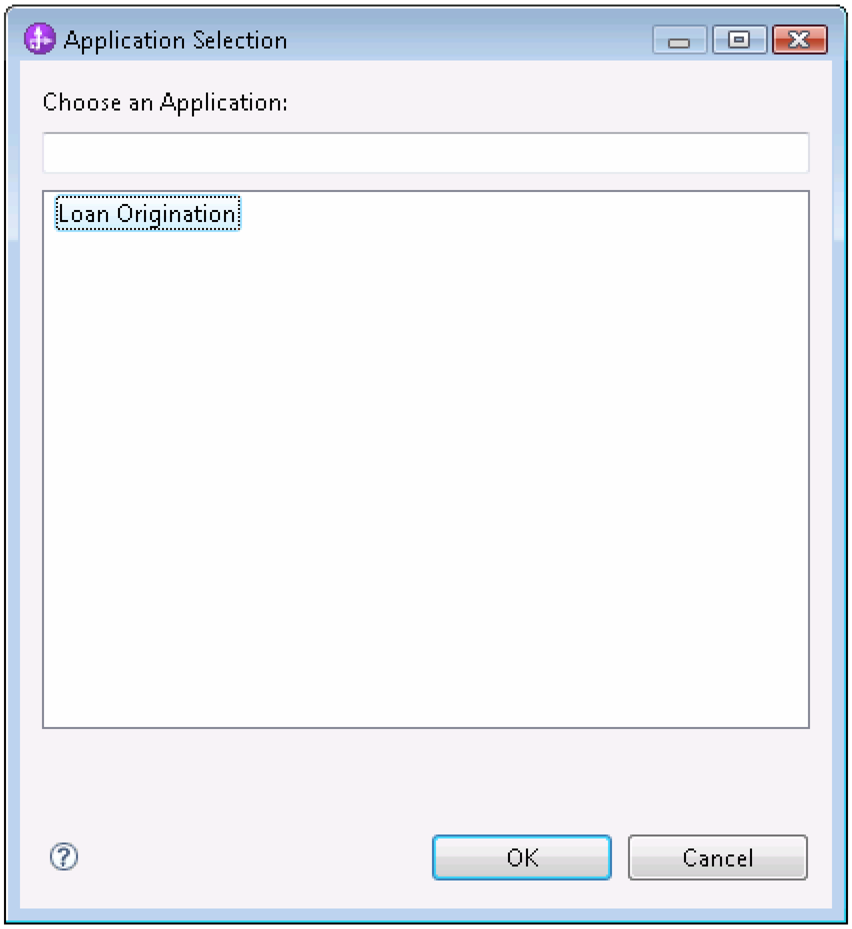
Channel Mapping

Export	Channel
Loan Origination Export	

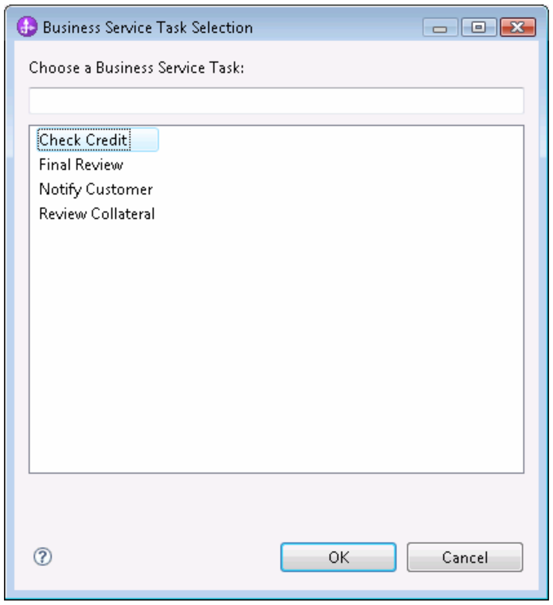
Overview | Business Application | Imported References | Exported References | Components

2. Check the box “Maps to a Business Application Module” to enable the rest of this page.

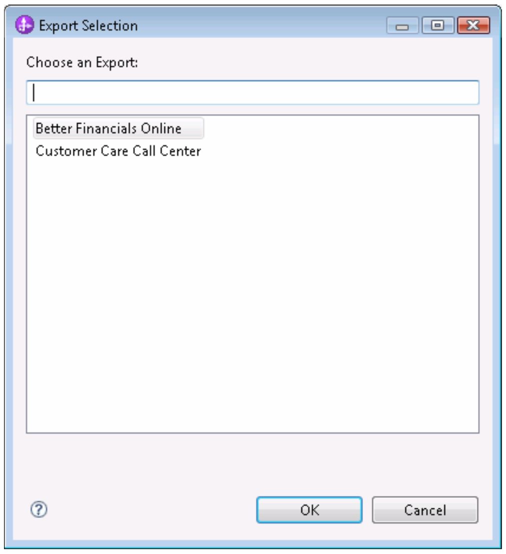
3. Click the button for Business Application, select Loan Origination, and click OK.



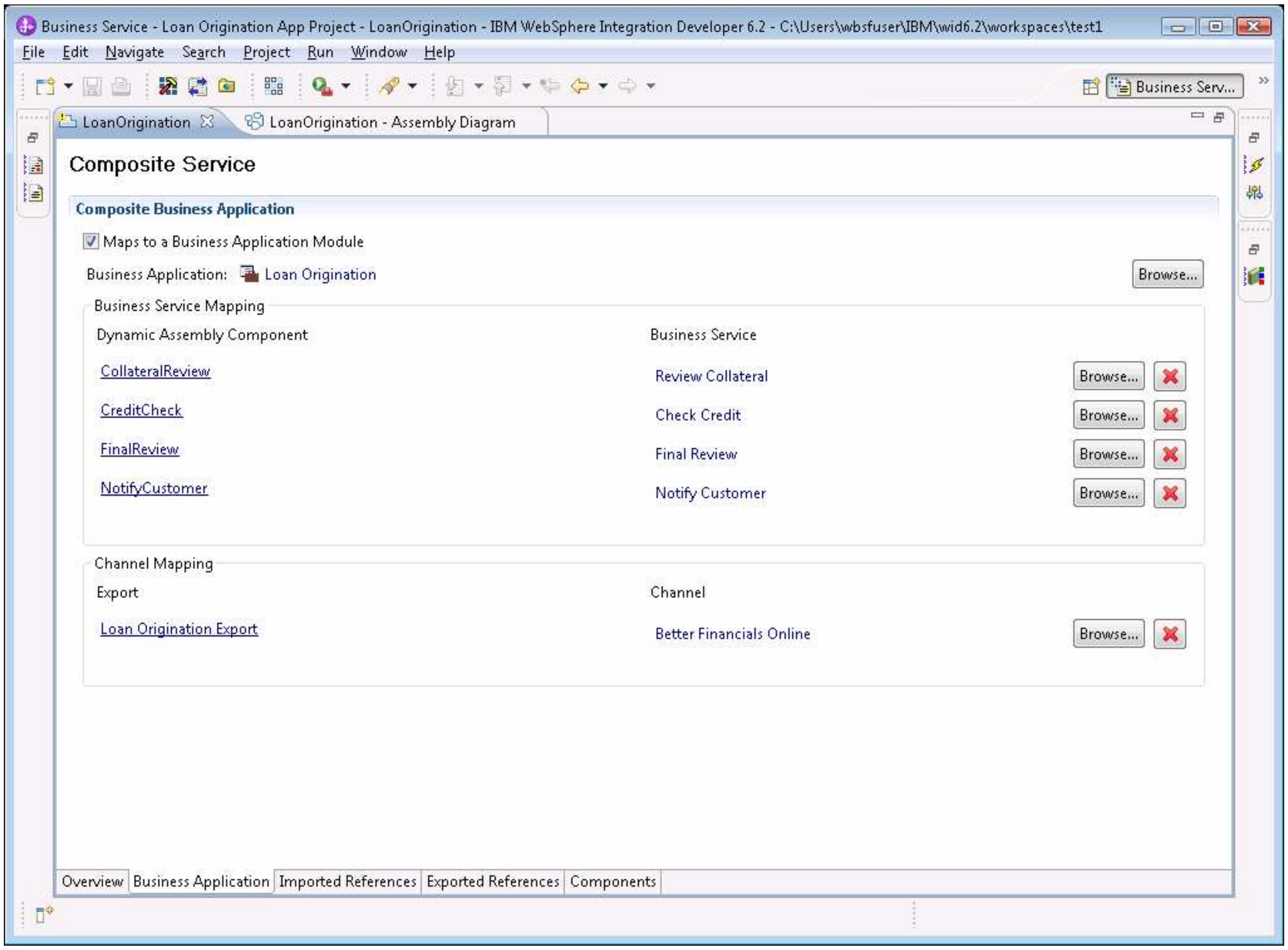
4. Then, for each Dynamic Assembly Component, choose the business service task that it represents.



5. For each export in the module, select the Business channel that is invoking it.



6. After these mappings, the tab should look like this:


Business Service - Loan Origination App Project - LoanOrigination - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\wid6.2\workspaces\test1

Composite Service

Composite Business Application

☑ Maps to a Business Application Module

Business Application: Loan Origination

Business Service Mapping

Dynamic Assembly Component	Business Service
CollateralReview	Review Collateral
CreditCheck	Check Credit
FinalReview	Final Review
NotifyCustomer	Notify Customer

Channel Mapping

Export	Channel
Loan Origination Export	Better Financials Online

At runtime, these mappings allow the system to use the active export, module, and DA component to find:

- The business **channel** associated to the export
- The **application** associated with the module
- The **business service** and optional **role** associated with each DA Component.

These business dimensions are then made available in the context for invoking business and technical policies.

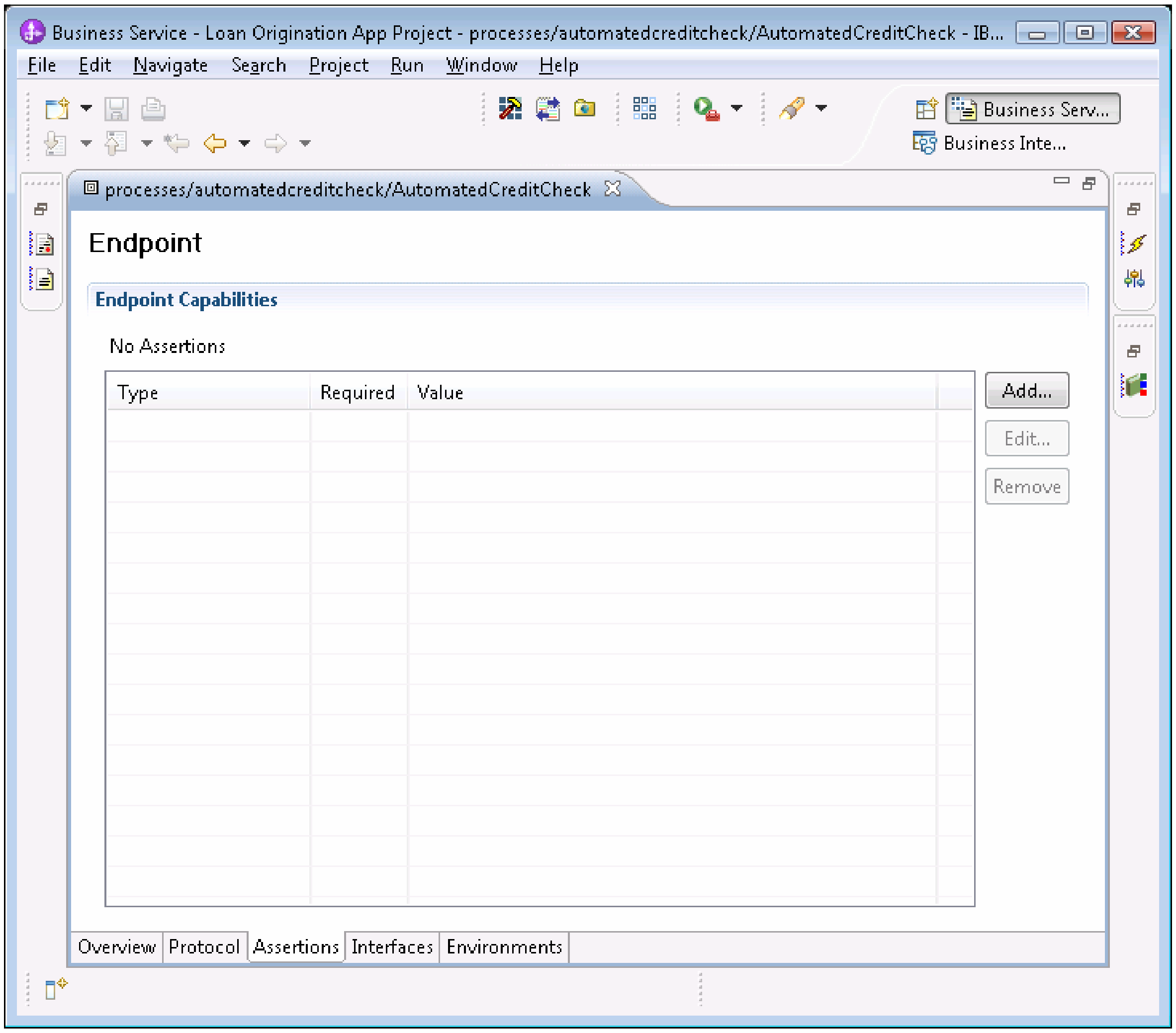
Important: Policies authored in the business tooling will only be considered by dynamic assembly components that have been configured, in this way, as part of a business application module.

Associate process variations with endpoints

A dynamic assembly component uses policies to determine the most appropriate endpoint at runtime. When a business policy establishes that a given process variation of a business service should be selected, the system looks for an endpoint that declares that it supports this process variation.

In general, there might be several endpoints capable of supporting a given process variation and technical policies can enforce additional constraints on the selection. But for now, you have one endpoint for each variation.

1. Open the Automated Credit Check endpoint and switch to the "Assertions" tab.


Business Service - Loan Origination App Project - processes/automatedcreditcheck/AutomatedCreditCheck - IB...
File Edit Navigate Search Project Run Window Help
Business Serv...
Business Inte...
processes/automatedcreditcheck/AutomatedCreditCheck
Add...
Edit...
Remove
Overview Protocol Assertions Interfaces Environments

Endpoint

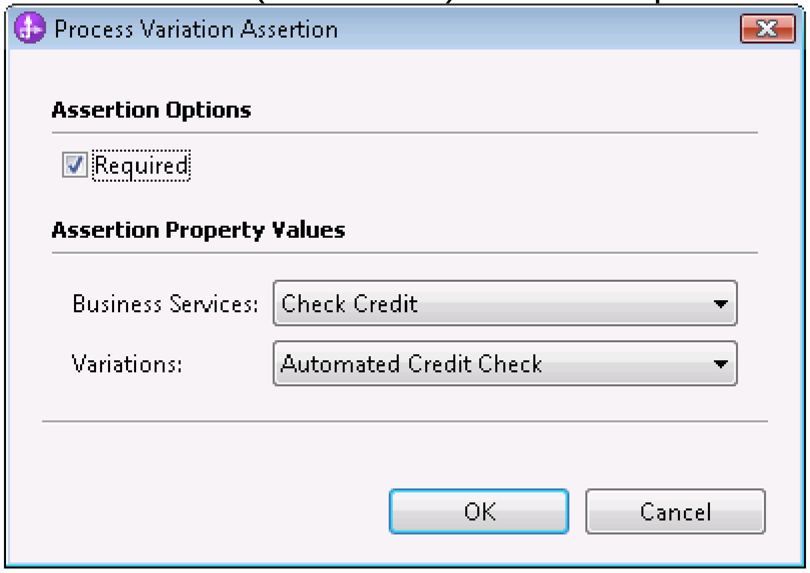
Endpoint Capabilities

No Assertions

Type	Required	Value

2. Click the Add button. Select Process Variation Assertion from the dialog and fill in the appropriate business service (Credit Check) and

variation (Automated). Select Required. Click OK.



[✓] Required

Business Services: Check Credit

Variations: Automated Credit Check

Setting **required** on an endpoint assertion means that the endpoint will only be considered if the selection policy contains a matching assertion. It is a good practice to set required when adding Process Variation assertions.

Set up all process variation assertions as follows:

Endpoint	Business Service	Variation
AutomatedCreditCheck	Check Credit	Automated Credit Check
AutomatedUnderwriting	Final Review	Automated Underwriting
AutomaticNotification	Notify Customer	Automatic Notification
HumanCollateralReview	Review Collateral	Human Collateral Review
HumanNotification	Notify Customer	Human Notification
HumanUnderwriting	Final Review	Human Underwriting
NoCollateralReview	Review Collateral	No Collateral Review

Define context specifications

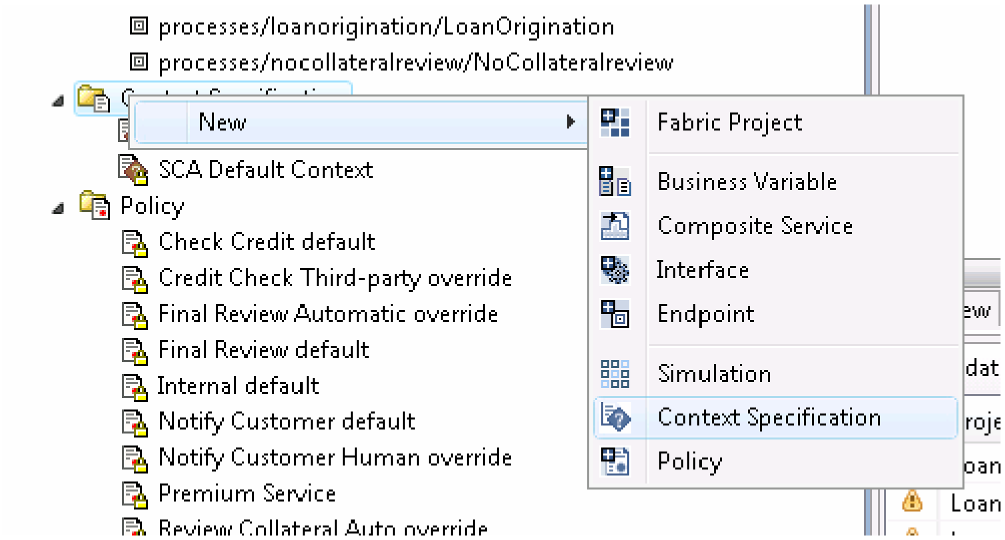
The next step is to identify the context elements necessary for a dynamic assembly component to apply the policies that have been implemented by the business user.

A context specification is used to ensure that all dimensions declared as required are made available at runtime. At design time a context specification also determines what context dimensions are available for test simulations of dynamic assembly.

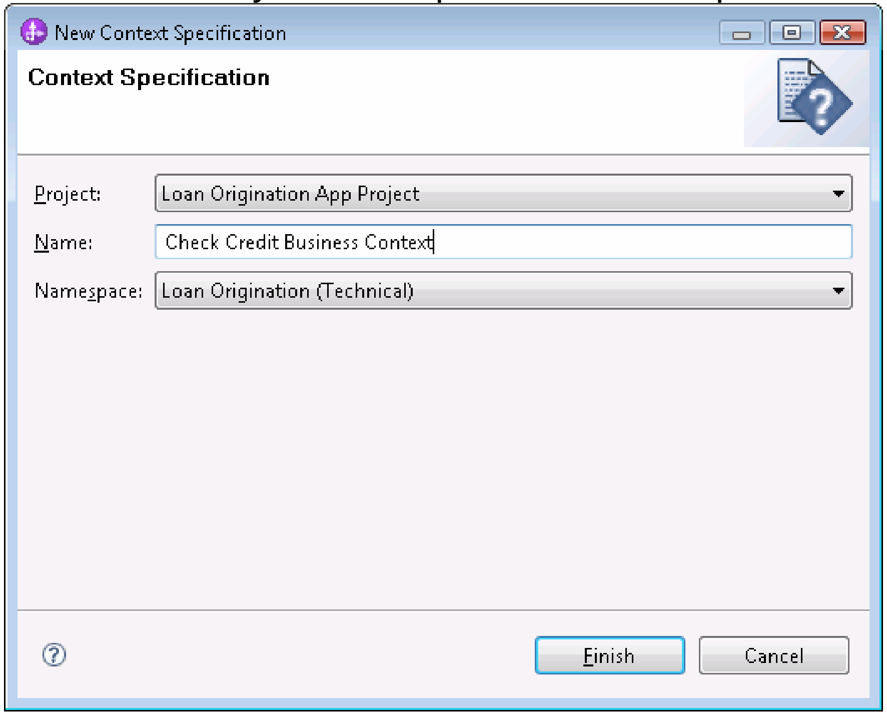
A new context specification automatically includes certain well-known context dimensions that are provided automatically, such as the Service Interface and Business Service dimension. Generally, it is necessary to determine what additional business concepts are used in policy conditions and include these in a context specification as well.

Next configure a context specification for Check Credit, the first business service in your flow. From looking at the business policies, you can tell that the policies need to know whether the customer applying for the loan is a new or existing customer. Also, you can tell that the amount of the loan is needed for processing.

1. From the Business Application Explorer, select New -> Context Specification.



2. Provide a name “Check Credit Business Context”, and click Finish. The newly created specification will open as an editor.


New Context Specification

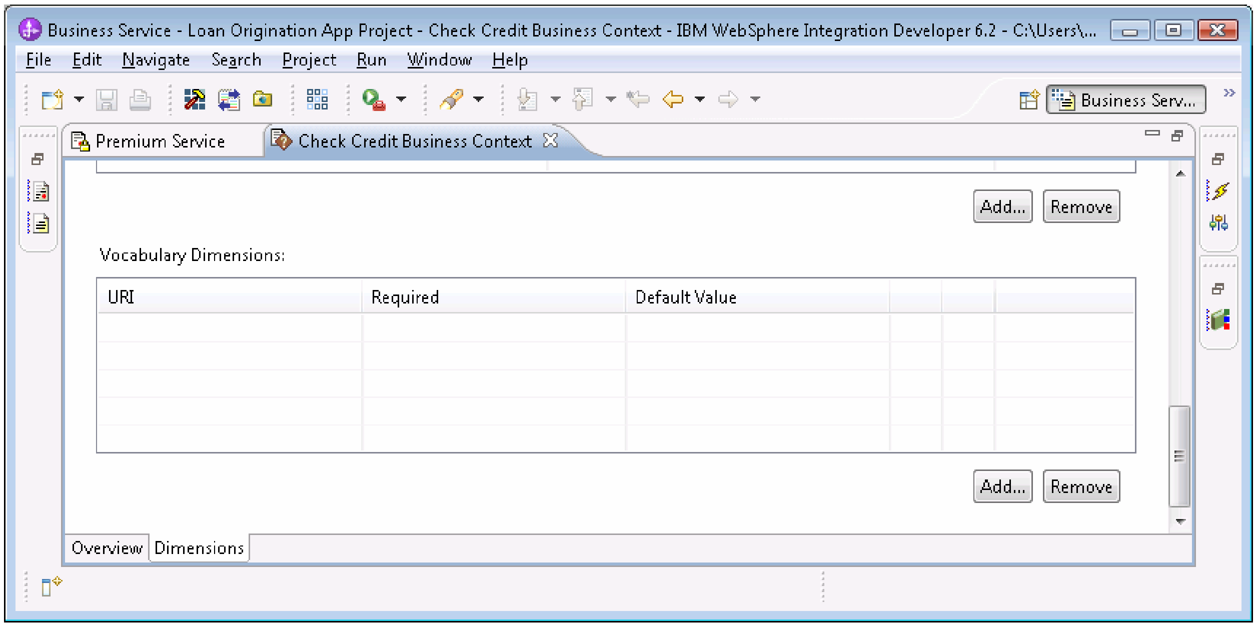
Context Specification

Project: Loan Origination App Project

Name: Check Credit Business Context

Namespace: Loan Origination (Technical)

3. Switch to the Dimensions tab of the editor.

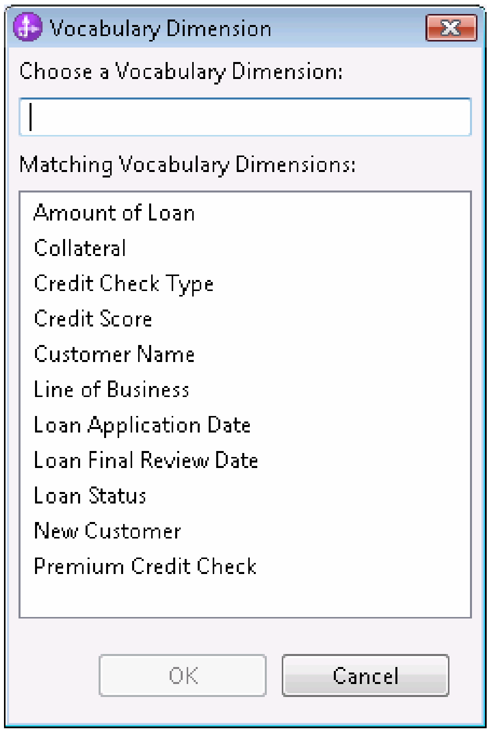

Business Service - Loan Origination App Project - Check Credit Business Context - IBM WebSphere Integration Developer 6.2 - C:\Users\...

File Edit Navigate Search Project Run Window Help
Business Serv...
Premium Service
Check Credit Business Context
Add... Remove
Overview Dimensions

Vocabulary Dimensions:

URI	Required	Default Value

4. Under Vocabulary Dimensions, click the Add button. Select Amount of Loan, and click OK.

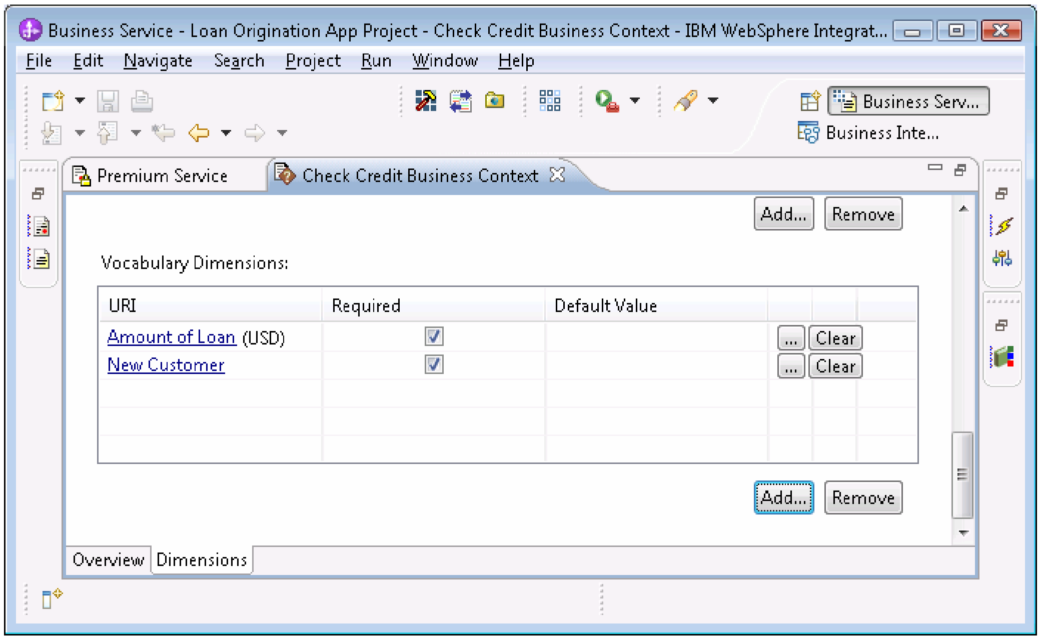


Choose a Vocabulary Dimension:

Matching Vocabulary Dimensions:

Amount of Loan
Collateral
Credit Check Type
Credit Score
Customer Name
Line of Business
Loan Application Date
Loan Final Review Date
Loan Status
New Customer
Premium Credit Check

5. Follow the same steps to add New Customer as second vocabulary dimension. The resulting specification looks like this:


Business Service - Loan Origination App Project - Check Credit Business Context - IBM WebSphere Integrat...
File Edit Navigate Search Project Run Window Help
Business Serv...
Business Inte...
Premium Service
Check Credit Business Context
Add... Remove
Overview Dimensions

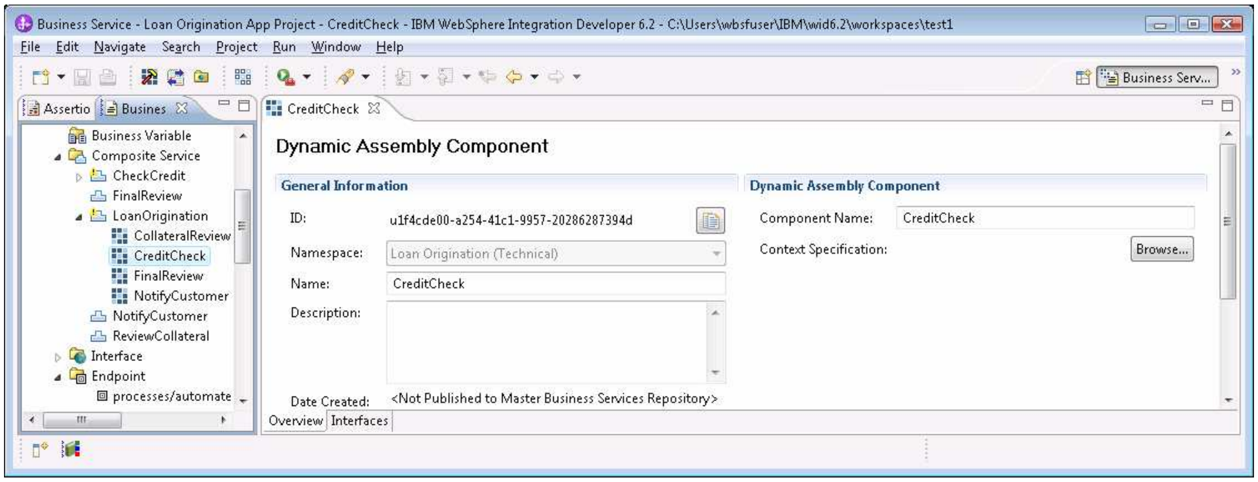
Vocabulary Dimensions:

URI	Required	Default Value		
Amount of Loan (USD)	✓		...	Clear
New Customer	✓		...	Clear

The **required** option is selected by default for newly added dimensions. It is a good idea to mark a context dimension as required if it should always have a value. If a dimension is only sometimes available in the context, deselect required. The system will not signal any error if an optional dimension is not provided.

A **default value** can be specified for a vocabulary dimension. Setting a default value has the same meaning as using a context extractor (next chapter) to specify a hard-coded value. However, this has the advantage that these values can be modified without changing or redeploying code. Use a default value if the value for a given decision is fixed. In your examples, you derive all values from business objects and other policies, so there is no need to set default values.

6. Open the LoanOrigination composite service, and double-click to open the editor for the CreditCheck Dynamic Assembly Component.


Business Service - Loan Origination App Project - CreditCheck - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\wid6.2\workspaces\test1

Dynamic Assembly Component

General Information

ID: u1f4cde00-a254-41c1-9957-20286287394d

Namespace: Loan Origination (Technical)

Name: CreditCheck

Description:

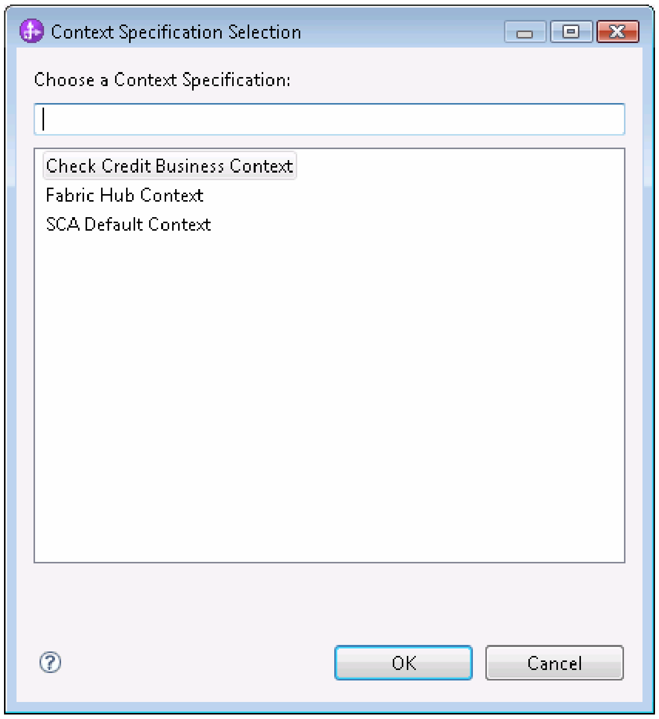
Date Created: <Not Published to Master Business Services Repository>

Dynamic Assembly Component

Component Name: CreditCheck

Context Specification: Browse...

7. Click Browse … to provide a value for the Context Specification field. Select Check Credit Business Context. Click OK. Save.


Context Specification Selection

Choose a Context Specification:

Check Credit Business Context
Fabric Hub Context
SCA Default Context

Note that a validation warning opens for any dynamic assembly components that do not have context specifications associated with them. Check for these validation warnings to make sure that this configuration step is not forgotten.

8. Use this table to define context specifications for the remaining DA components.

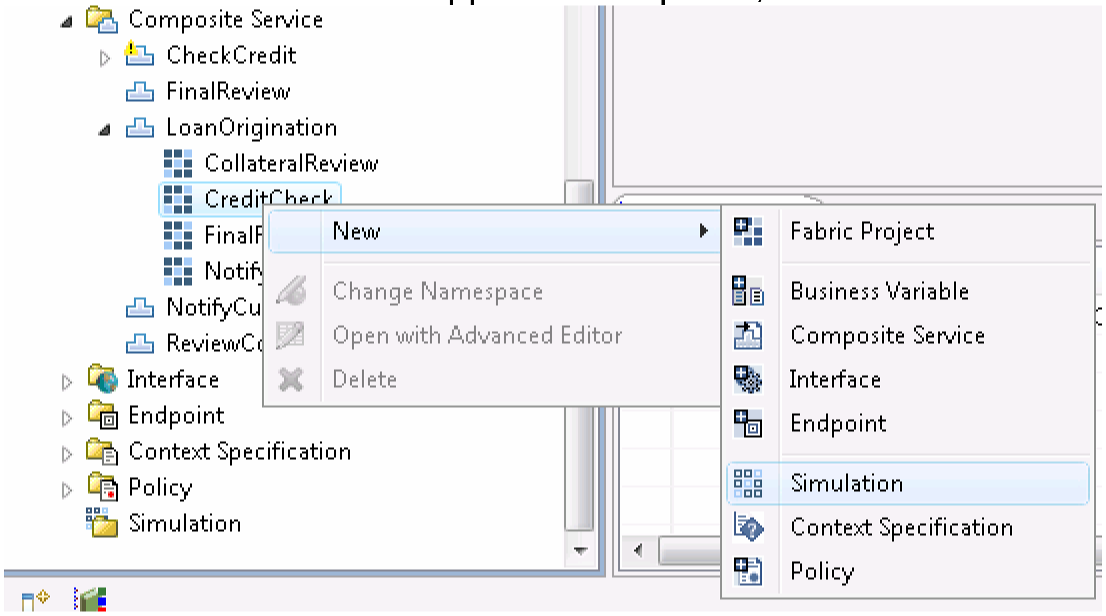
Dynamic Assembler	Additional Context Requirements
CreditCheck	New Customer, Amount of Loan
CollateralReview	Line of Business
FinalReview	Amount of Loan, Loan Status
Notify Customer	Loan Status

Simulate (test) business service decisions

A simulation in Composition Studio is a test that represents how the dynamic assembler behaves in a given business situation. It is a good idea to define a set of scenarios that represent the various inputs and expected outcomes for a given decision. For example, you have policies that determine whether you should review collateral based on line of business. To test this, you should have at least one case where the line of business causes a collateral review and one where it does not.

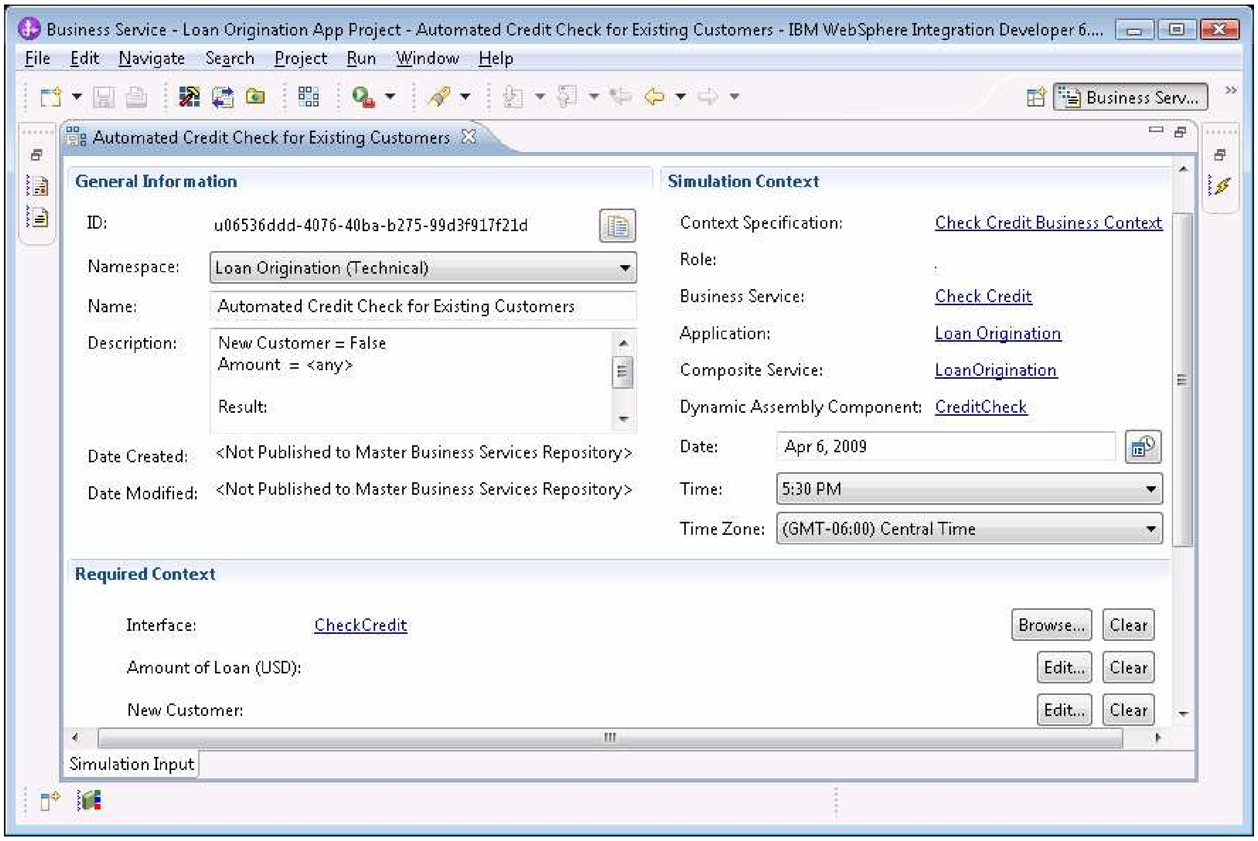
Next create your first test for the CreditCheck dynamic assembly component.

1. From the Business Application Explorer, select New -> Simulation.



2. Fill in these details and click Finish:

Name	DA Component	Description
Automated Credit Check for Existing Customers	CreditCheck	New Customer = False Amount = <any> ---------------------------- Select Automated Variation & Establish Credit Check Type = INTERNAL


Business Service - Loan Origination App Project - Automated Credit Check for Existing Customers - IBM WebSphere Integration Developer 6....

Automated Credit Check for Existing Customers

General Information

ID: u06536ddd-4076-40ba-b275-99d3f917f21d

Namespace: Loan Origination (Technical)

Name: Automated Credit Check for Existing Customers

Description: New Customer = False
Amount = <any>

Result:

Date Created: <Not Published to Master Business Services Repository>

Date Modified: <Not Published to Master Business Services Repository>

Simulation Context

Context Specification: Check Credit Business Context

Role:

Business Service: Check Credit

Application: Loan Origination

Composite Service: LoanOrigination

Dynamic Assembly Component: CreditCheck

Date: Apr 6, 2009

Time: 5:30 PM

Time Zone: (GMT-06:00) Central Time

Required Context

Interface: CheckCredit

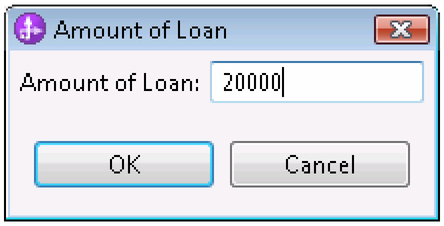
Amount of Loan (USD):

New Customer:

Since you might have several simulations for each Dynamic Assembler, it is a good idea to name your simulations to identify the test scenario. The description field is a good place to summarize the inputs and expected outcome of the simulation.

Note that the Business Service and Application dimensions are automatically provided under “Simulation Context”. The simulator is using the Business Application mappings that you established earlier to provide these fields. If you do not see these fields provided automatically, it is a good indication that the mapping was not performed and that the dynamic assembler will not consider policies authored in business space.

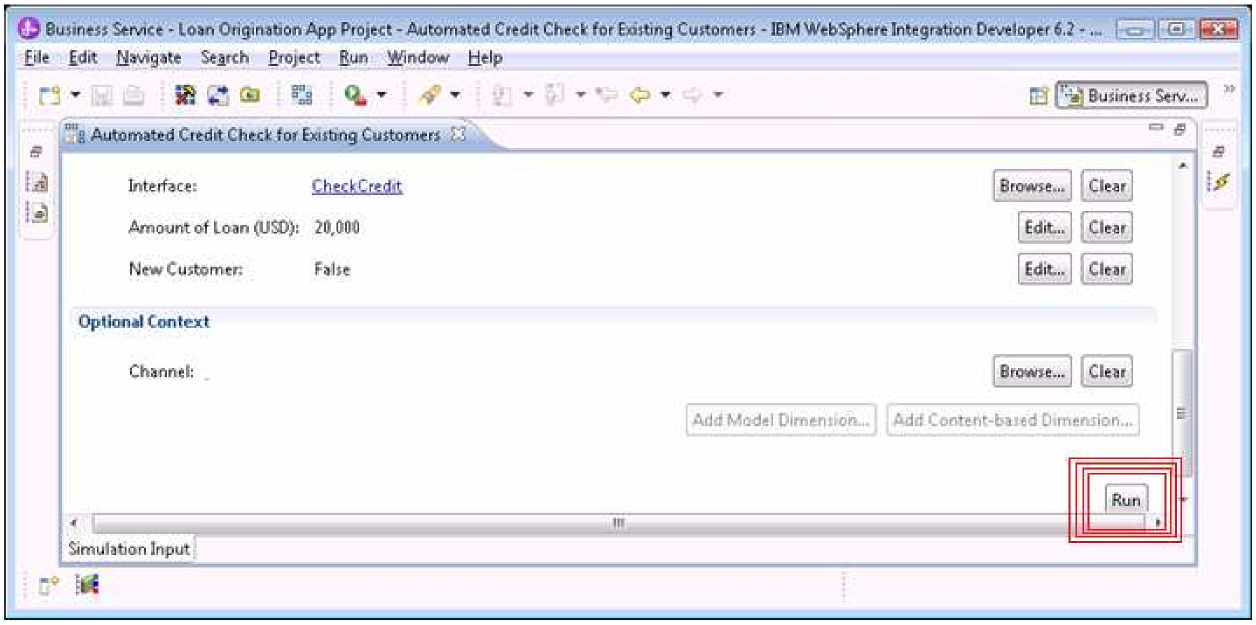
3. Fill out the context elements in the simulation editor. Click Edit… for Amount of Loan. Enter 20000. Click OK.


Amount of Loan

Amount of Loan: 20000

4. This simulation is for an existing customer, so set New Customer = False.

5. Click Run.


Business Service - Loan Origination App Project - Automated Credit Check for Existing Customers - IBM WebSphere Integration Developer 6.2 - ...

Automated Credit Check for Existing Customers

Interface: CheckCredit

Amount of Loan (USD): 20,000

New Customer: False

Optional Context

Channel:

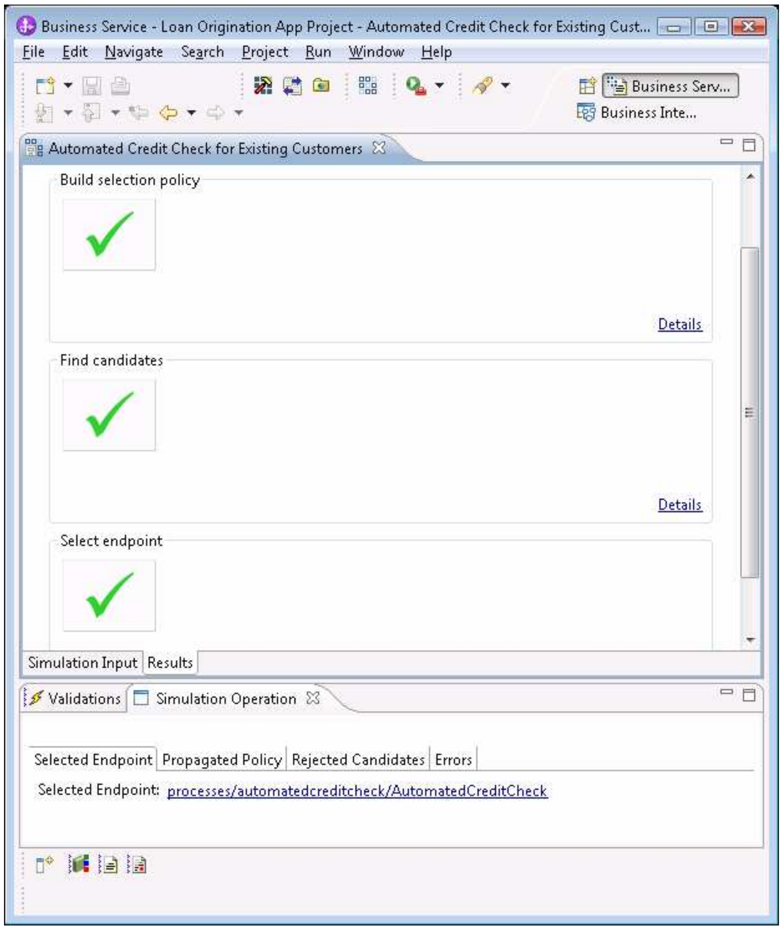
The results of running a simulation display the results for three phases of dynamic assembly. A green check mark shows up for each phase that successfully executed, or a red X will display on the first step that fails to execute. Keep in mind that an “all green” simulation might still be reflecting incorrect results. It is good idea to inspect the result of each phase.

During the **Build selection policy** phase, the system finds all policies that apply given the simulation input settings. It then merges these policies into a Composite Policy. During the merge, priority rules can cause a setting of one policy to override another. Inspecting the composite policy shows which settings made it into the resulting composite policy and indicates which policy won out in contributing each setting. It is a good idea to check these results to make sure that policies are being applied in the

way you expect.
During the **Find Candidates** phase, the system finds all endpoints that support the given interface and matches these endpoints against the composite policy. If an endpoint’s assertions do not match required assertions in the composite policy, the output will indicate that the endpoint was filtered using the selection policy. An endpoint will also be rejected at this phase if a required assertion on the endpoint is not satisfied by the selection policy. Remaining endpoints are according to increasing cost and a suitability score based on the closeness of the match for non-required policy assertions. This rank is referred to as a **tier**.

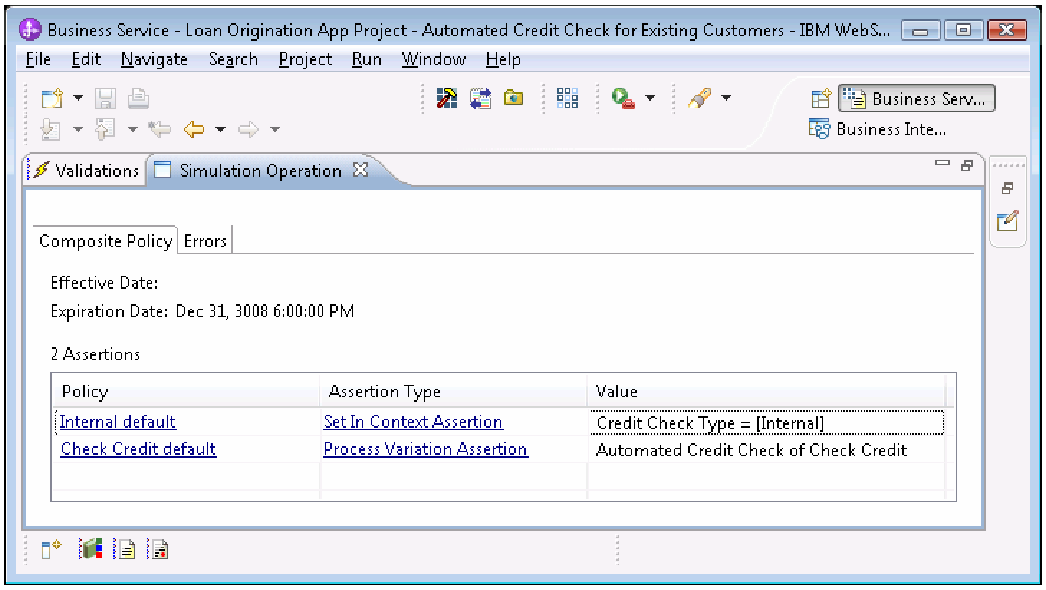
During the **Select endpoint** phase, the system selects an endpoint with the lowest tier value from the remaining candidates. If an endpoint is not available at the given time due to endpoint status, hours of operations, or other temporal concerns, that endpoint is rejected and the next best endpoint is considered.

6. Note that the simulation had no failures and that an endpoint supporting the automated credit check was successfully chosen.



Selected Endpoint: processes/automatedcreditcheck/AutomatedCreditCheck

7. Click the “Details” link for the Build selection policy phase. Verify that the policy sets the Credit Check Type to Internal.


Business Service - Loan Origination App Project - Automated Credit Check for Existing Customers - IBM WebS...
File Edit Navigate Search Project Run Window Help
Business Serv...
Business Inte...
Validations
Simulation Operation
Composite Policy
Errors

Effective Date:
Expiration Date: Dec 31, 3008 6:00:00 PM

2 Assertions

Policy	Assertion Type	Value
Internal default	Set In Context Assertion	Credit Check Type = [Internal]
Check Credit default	Process Variation Assertion	Automated Credit Check of Check Credit

8. Create simulations for the other business service decisions as indicated by the table below:

DA Component	Scenario	Description
CheckCredit	Premium Credit Check for New Customers	New Customer = True Amount = 30,000 --------------------------------- Select Automated Credit Check Credit Check Type = Third-Party Premium Credit Service = True
CheckCredit	Standard Credit Check for New Customers	New Customer = True Amount = 20,000 ------------------------------ Select Automated Credit Check Credit Check Type = Third-Party
CollateralReview	No Collateral Review for Auto Loans	Line of Business = Auto ------------------------------------- Select No Collateral Variation
CollateralReview	Human Collateral Review for RV Loans	Line of Business = RV -------------------------------- Select Human Collateral Review
FinalReview	Automated Underwriting for Small Loans	Loan Amount = 10,000 Loan Status = In Progress ------------------------------ Select Automated Underwriting
FinalReview	Human	Loan Amount = 30,000

	Underwriting for Large Loans	Loan Status = In Progress --------------------------------------- Select Human Underwriting
FinalReview	Automatic Underwriting Pass Through for Rejected Loans	Loan Amount = 50,000 Loan Status = Rejected ___________________________________ Select Automatic Underwriting
NotifyCustomer	Automated Notification for Approved Status	Loan Status = Approved --------------------------- Select Automated Notification
NotifyCustomer	Human Notification When More Info Needed	Loan Status = More Info ------------------------------------ Select Human Notification

Configuring technical policies

Within the Automated Credit Check process variation you have place a dynamic assembly component responsible for selecting which credit service to use. To configure this endpoint selection, you will create some policies. Policies created in Composition Studio are not visible to business users using the Fabric authoring space, but the business has specified two concerns that are controlled with business policies:

- Credit Check Type = whether to use an internal credit check system or to get a fresh report from a vendor credit reporting agency.
- Premium Credit Service = whether to require the use of a credit service that offers better data, though probably at a higher cost.

A policy can be thought of as a statement of the form:

```
FOR <Policy Target>
    FROM <effective date>
    TO <expiration date>
WHEN <Conditions …>
THEN <assertions …>
```

Policies do not directly specify a required outcome, such as the selection of a particular endpoint. Instead, a policy provides one or more assertions that establish what must be considered when the policy is enforced. The dynamic assembler, as a policy enforcement point, filters out endpoints that are not consistent with a policy established for the invocation.

Set-in-context assertion

A **Set-in-context assertion** specifies a value for a business concept that should be set in the context. When the dynamic assembler sees this assertion in a composite policy, it will establish the specified content dimension with the specified value in the business context before invoking the selected endpoint.

In your example, you have a business –authored policy that sets a value for Credit Check Type (either internal or third-party) in the context before invoking the Automated Credit Check Process Variation.
Note that using Set-In-Context does not allow for forward-chaining in the evaluation of policies for a single dynamic assembly decision. In other words, the assertion impacts the context available for the endpoint that is invoked. It does not cause the system to reevaluate the policies for the current assembly step.

Note also, that a Set-In-Context Assertion is ignored for the purposes of selecting an endpoint. In fact, this assertion can only be added to policies. The **required** setting, since it only applies to matching assertions can be ignored.

Using fill from context for a set-in-context assertion is nonsense since that will direct the system to establish a value in the context that already exists.

Matching endpoints with vocabulary-based assertions

A v**ocabulary constraint assertion**, or constraint, is used to define an expression that serves as a constraint for endpoint selection. For example, a statement like “*the value of Credit Check Type must be equal to INTERNAL”* can be expressed with this type of assertion*.*

A v**ocabulary assertion**, or fact, is used to define facts that are considered when evaluating constraints specified with Vocabulary Constraint Assertions. For example, the statement “Credit Check Type is set to INTERNAL” can be represented as a fact.

During the Find Candidates stage of processing, the system first narrows the set of endpoints by eliminating endpoints that do not meet the requirements of the selection policy. Constraints in the policy are compared to facts about the same concept declared through Vocabulary Assertions on the endpoint. If the facts do not satisfy the constraints, the endpoint is rejected.

After filtering the by selection policy, the system considers constraints on the endpoint against facts declared in the selection policy and eliminates endpoints whose constraints are not satisfied.

Use the **required** option for constraints. If the assertion is not marked as required on the policy side, the system might arrive at a slightly worse score for an endpoint instead of filtering it out-right. On the endpoint side, a missing required flag will cause the constraint to have no effect on selection. Since facts are not actively involved in filtering, the required flag has no effect.

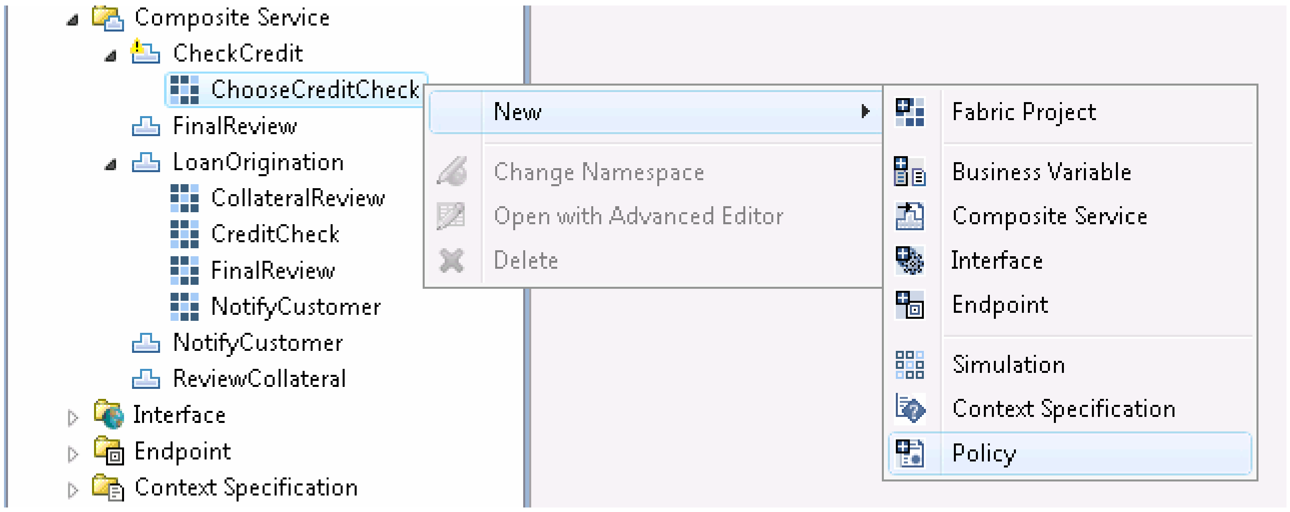
The **fill from context** option is only available for assertions in policies. When this option is used, it means that the value field for the assertion should use the value for the selected concept as established in the context. In your example, business policies use the set-in-context capability to establish values in the context. You will use the fill-from-context option to pick up these business settings endpoint matching purposes.

Next, you will create your first technical policy. Extending the pseudo-grammar introduced earlier, you want your policy to express:

FOR <ChooseCreditCheck>
WHEN *always*
THEN Assert the FACT (Credit Check Type is <fill-from-context>)

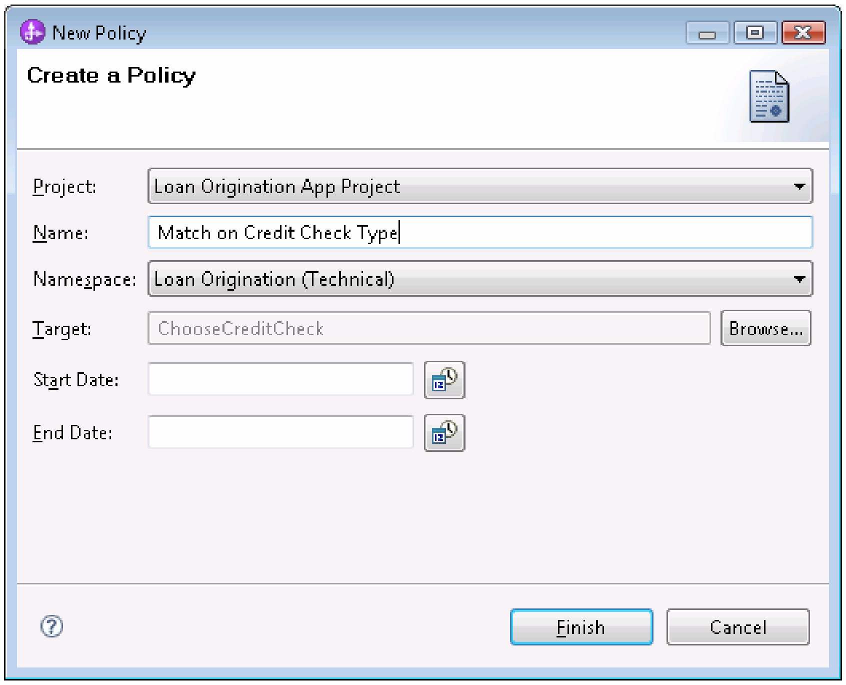
This policy will cause the business policies that set values for Credit Check Type to play a role in how the endpoint selection decision is made.

1. Select the ChooseCreditCheck DA Component and select New -> Policy with the context menu.



When creating a policy, the system will use the selected element as the policy target, possible. You can also select the target explicitly in the create wizard.

2. Enter “Match on Credit Check Type” for the name. Click Finish.



Project: Loan Origination App Project

Name: Match on Credit Check Type

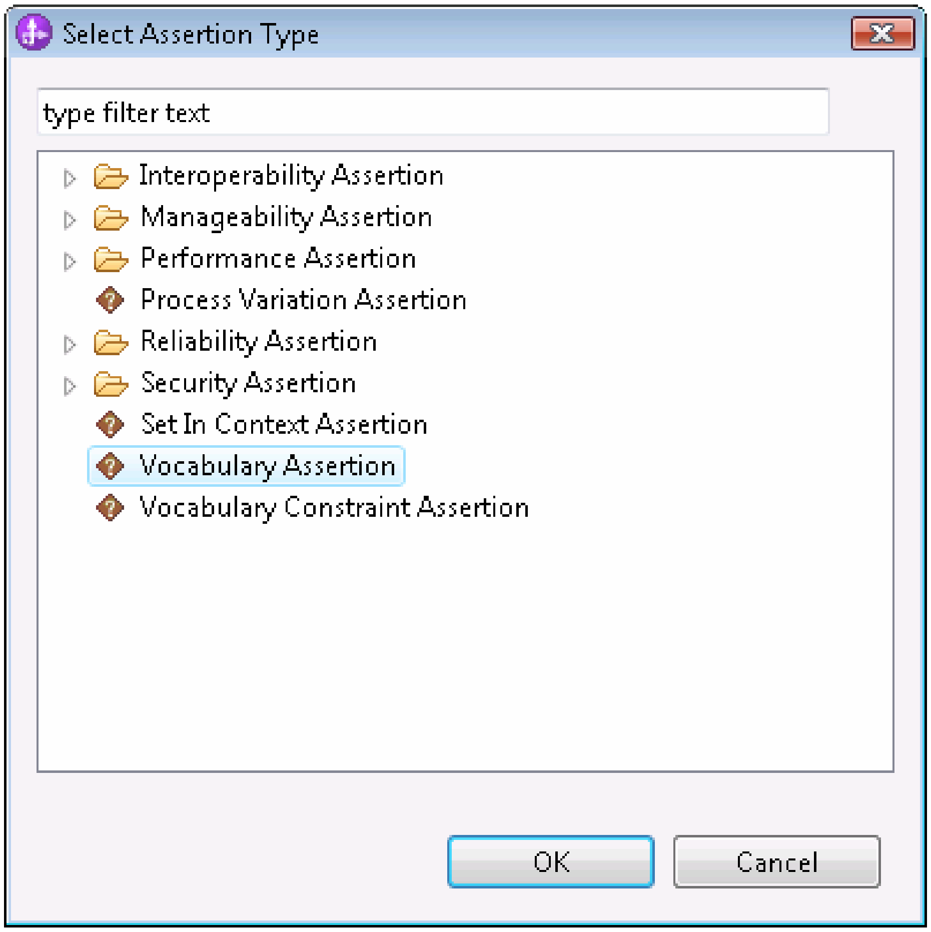
Namespace: Loan Origination (Technical)

Target: ChooseCreditCheck

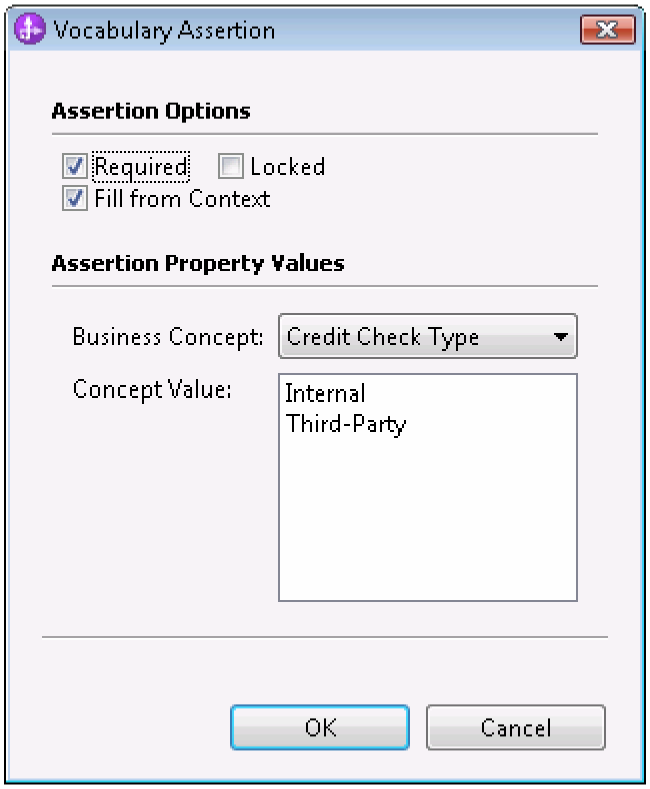
Start Date:

End Date:

3. Go to the “Policy Expression” tab of the policy editor and click the Add button to add a new assertion. You are configuring a fact, so choose Vocabulary Assertion.



4. Select the Credit Check Type concept and enable “Fill from context”. Click OK.


Vocabulary Assertion

Assertion Options

[✓] Required [ ] Locked
[✓] Fill from Context

Assertion Property Values

Business Concept: Credit Check Type

Concept Value: Internal, Third-Party

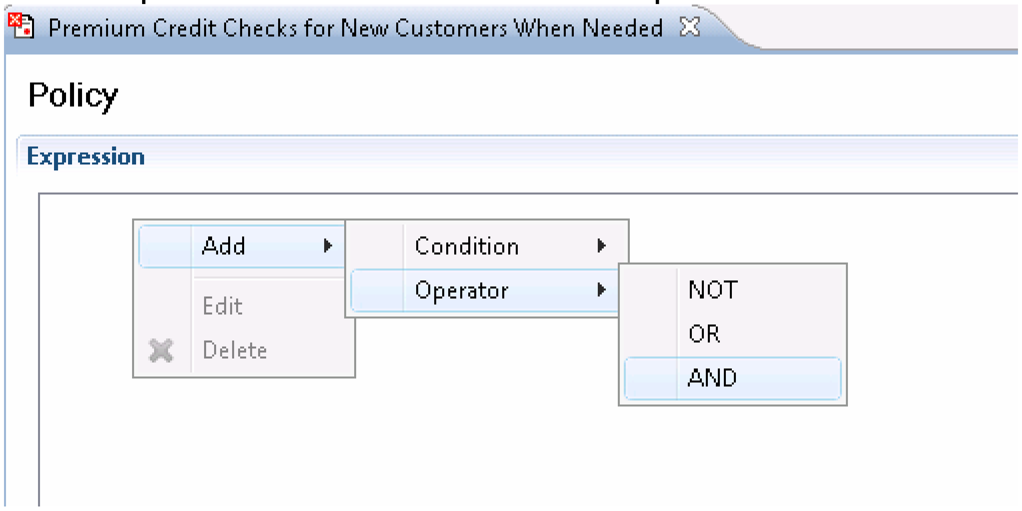
Sometimes, a validation error will occur if the Concept Value field is not filled. If you see this error, it can be resolved by picking any value for the concept. If fill from context is enabled, the system will always replace any value you select with the active value in the context. However, if no value is supplied in the context, the system will use the value you specified. If a value from context must always be used, set the vocabulary dimension as required in the context specification and the system will never allow the value entered here to be used. On the other hand, if the value will not always be available in the context, make the dimension optional and the value supplied here is used as a default.

Next, you will create a second policy that will allow the dynamic assembler to enforce the business decision about when to use a Premium Credit service. In pseudo-grammar:

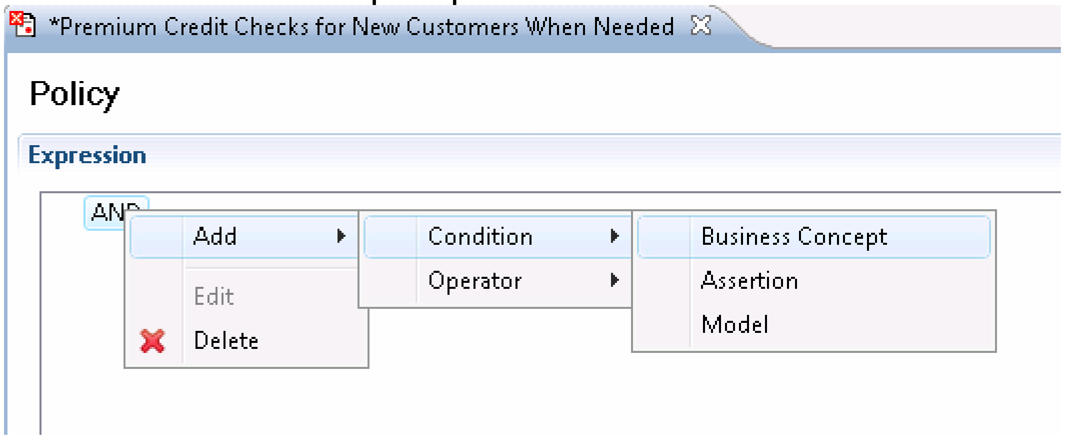
```
FOR <ChooseCreditCheck>
WHEN Premium Credit Check = True AND New Customer = True
THEN Assert Constraint (Premium Credit Check must equal True)
```

5. Create a second policy targeted at ChooseCreditCheck with the name:
   Premium Credit Checks for New Customers When Needed

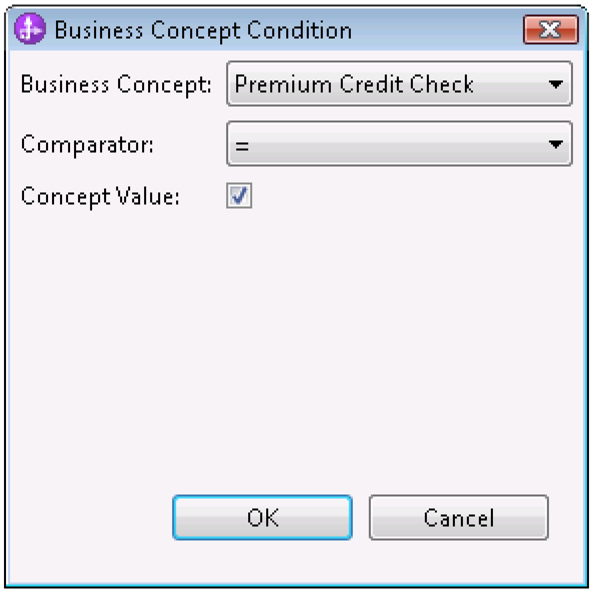
6. Switch to the Expression tab and use the context menu in the expression area to add an AND operator.



7. Select the AND operator and use the context menu to add a new Business Concept expression.



8. Select the Premium Credit Check concept, the equals comparator, and the value true.

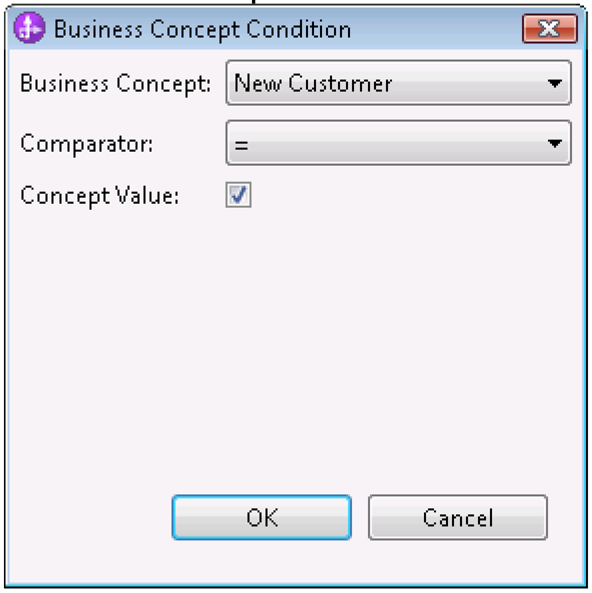

Business Concept Condition

Business Concept: Premium Credit Check

Comparator: =

Concept Value: ✓

9. Add a second business concept expression to the AND operator. Fill it in to express “New Customer = True”.

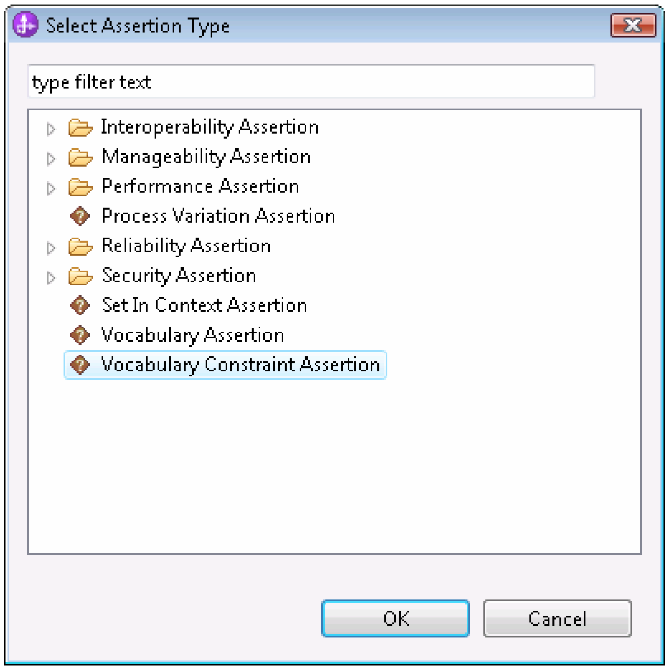

Business Concept Condition

Business Concept: New Customer

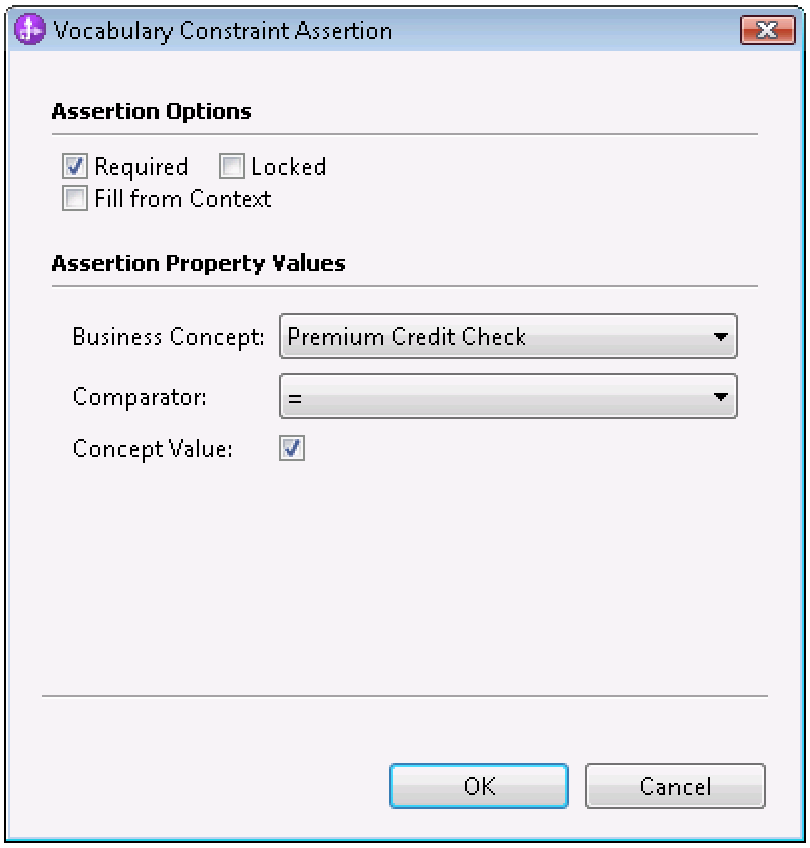
Comparator: =

Concept Value: ✓

10. Click the Add button to add a new assertion to the policy. Select Vocabulary Constraint Assertion. Click OK.



11. Use the dialog to express the constraint “Premium Credit Check = True”



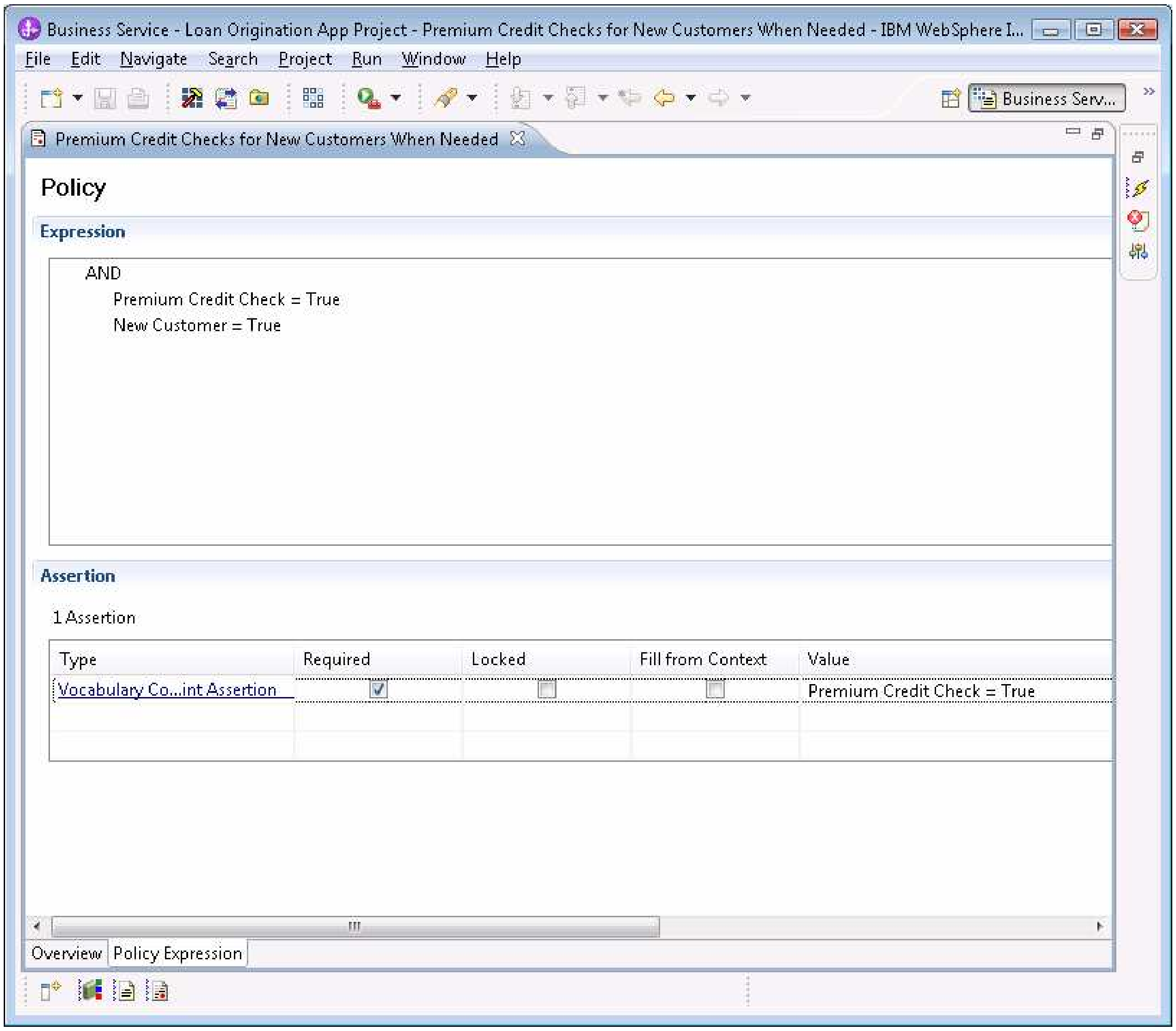
Assertion Options

[✓] Required [ ] Locked
[ ] Fill from Context

Assertion Property Values

Business Concept: Premium Credit Check
Comparator: =
Concept Value: [✓]

The resulting policy should look like this:


Business Service - Loan Origination App Project - Premium Credit Checks for New Customers When Needed - IBM WebSphere I...

Premium Credit Checks for New Customers When Needed

Policy

Expression

```
AND
    Premium Credit Check = True
    New Customer = True
```

Assertion

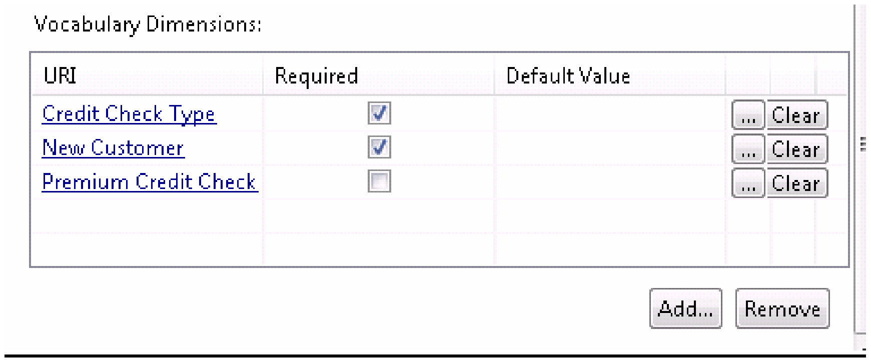
1 Assertion

Type	Required	Locked	Fill from Context	Value
Vocabulary Co...int Assertion	✓			Premium Credit Check = True

Overview | Policy Expression

12. Create a new context specification as shown earlier. Add these vocabulary dimensions:

- Credit Check Type, required
- Premium Credit Service, optional
- New Customer, required


Vocabulary Dimensions:

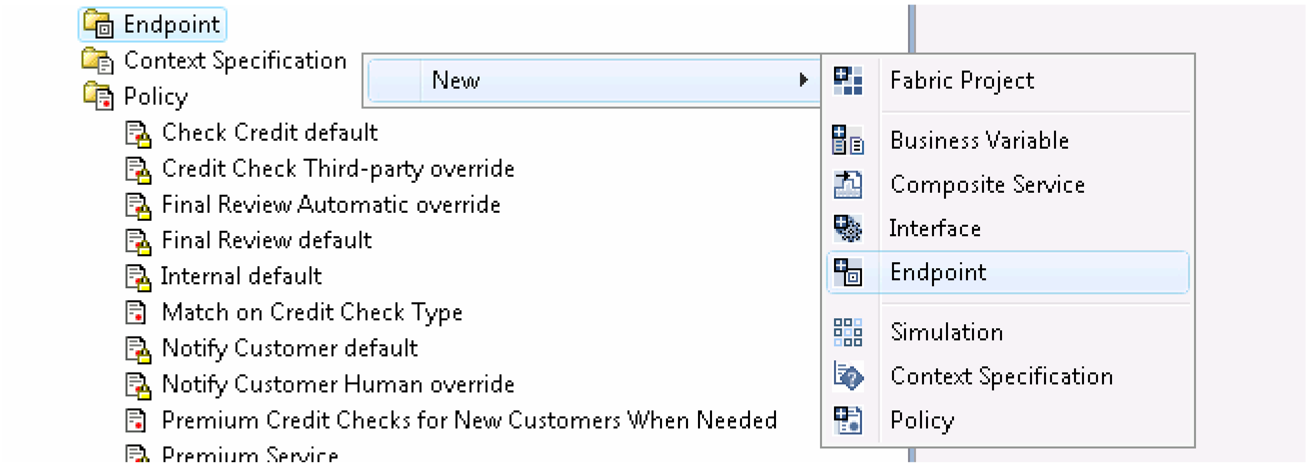
URI	Required	Default Value		
Credit Check Type	✓		...	Clear
New Customer	✓		...	Clear
Premium Credit Check	☐		...	Clear

13. Open the editor for the ChooseCreditCheck DA Component and associate the new context specification.

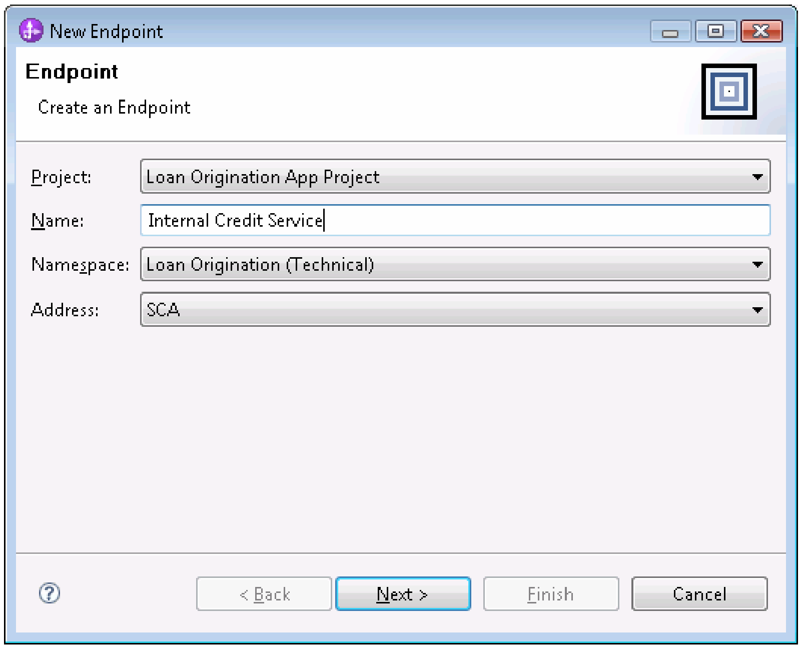
Create endpoints for credit check service

Your ChooseCreditCheck dynamic assembly component needs to have some endpoints to consider. Earlier, you created endpoints for your process variations by importing some SCA modules. Here you will create some endpoints for existing credit check services.

1. From the Business Application Explorer, select New -> Endpoint.



2. This endpoint will represent your internal credit service. Provide a label and click Next.


New Endpoint

Endpoint

Create an Endpoint

Project: Loan Origination App Project

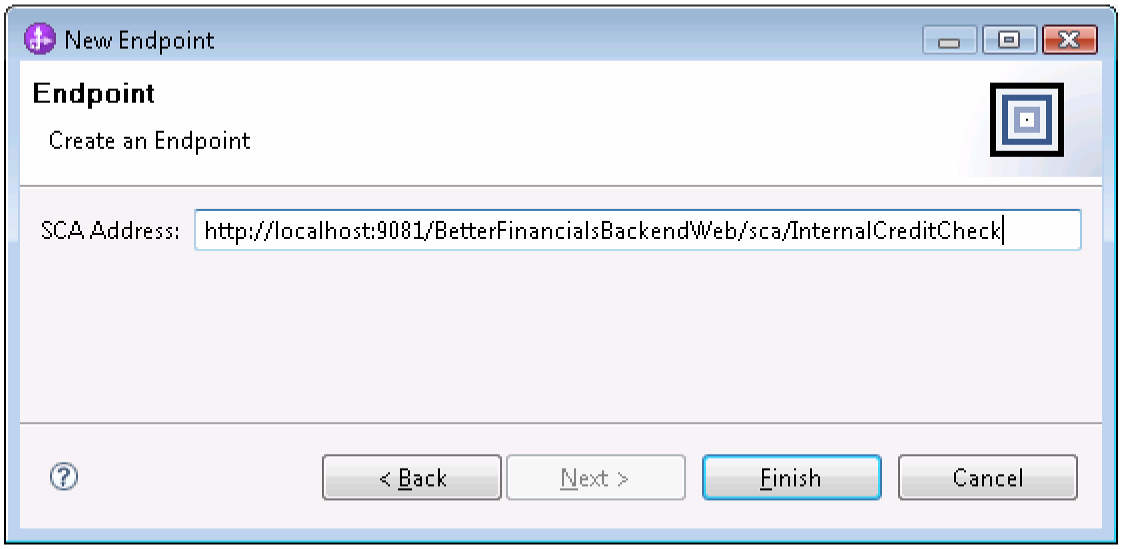
Name: Internal Credit Service

Namespace: Loan Origination (Technical)

Address: SCA

There are several available address types for endpoints, but you only need to use the SCA address type.

3. Using the server and port where the credit services are deployed enter a URL like: http://<host>:<port>/BetterFinancialsBackendWeb/sca/InternalCreditCheck

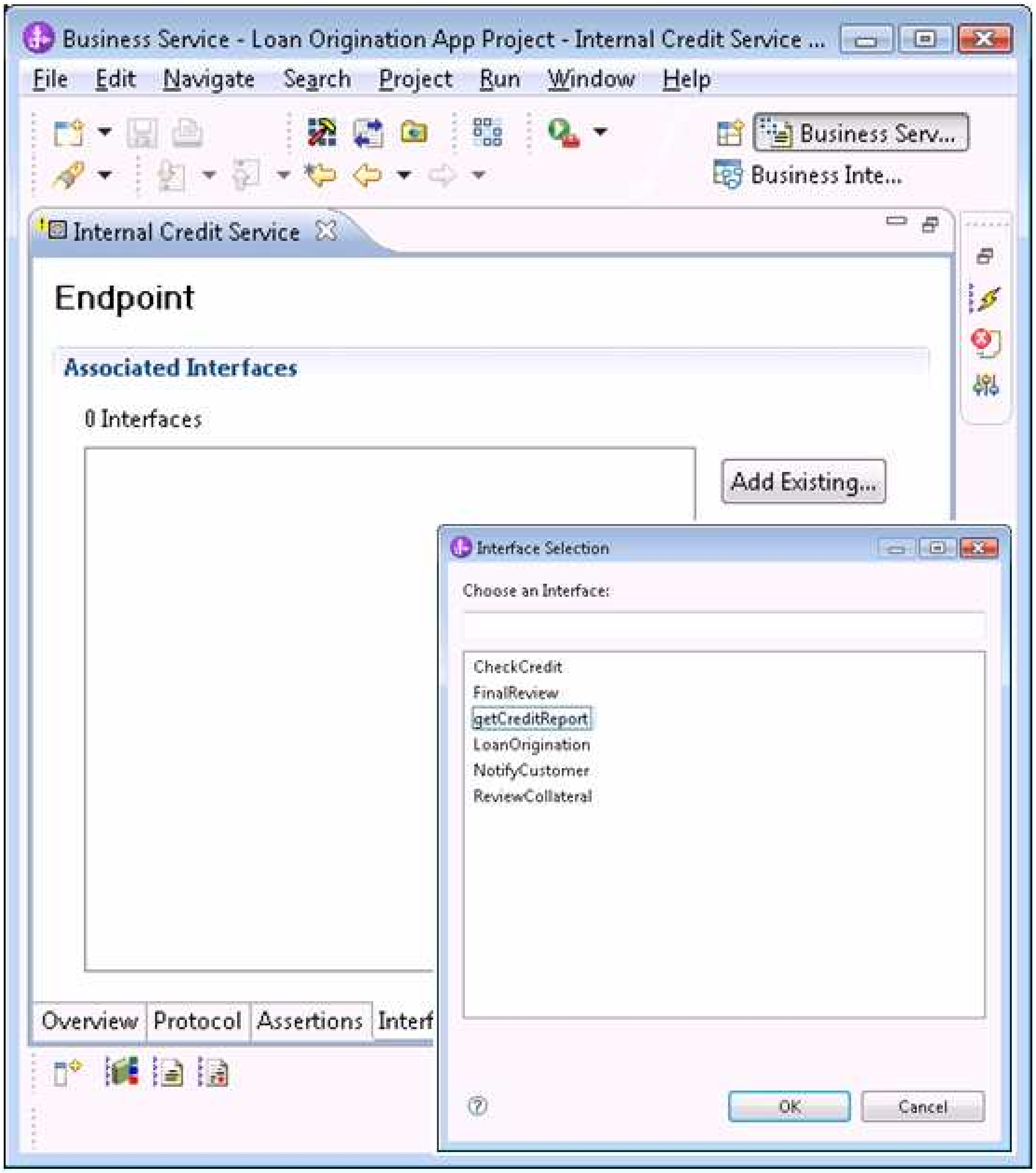

New Endpoint

Endpoint

Create an Endpoint

SCA Address: http://localhost:9081/BetterFinancialsBackendWeb/sca/InternalCreditCheck

4. Go to the Interfaces tab of the Endpoint editor and add the getCreditReport Interface.


Business Service - Loan Origination App Project - Internal Credit Service ...
File Edit Navigate Search Project Run Window Help
Business Serv...
Business Inte...
Internal Credit Service

Endpoint

Associated Interfaces

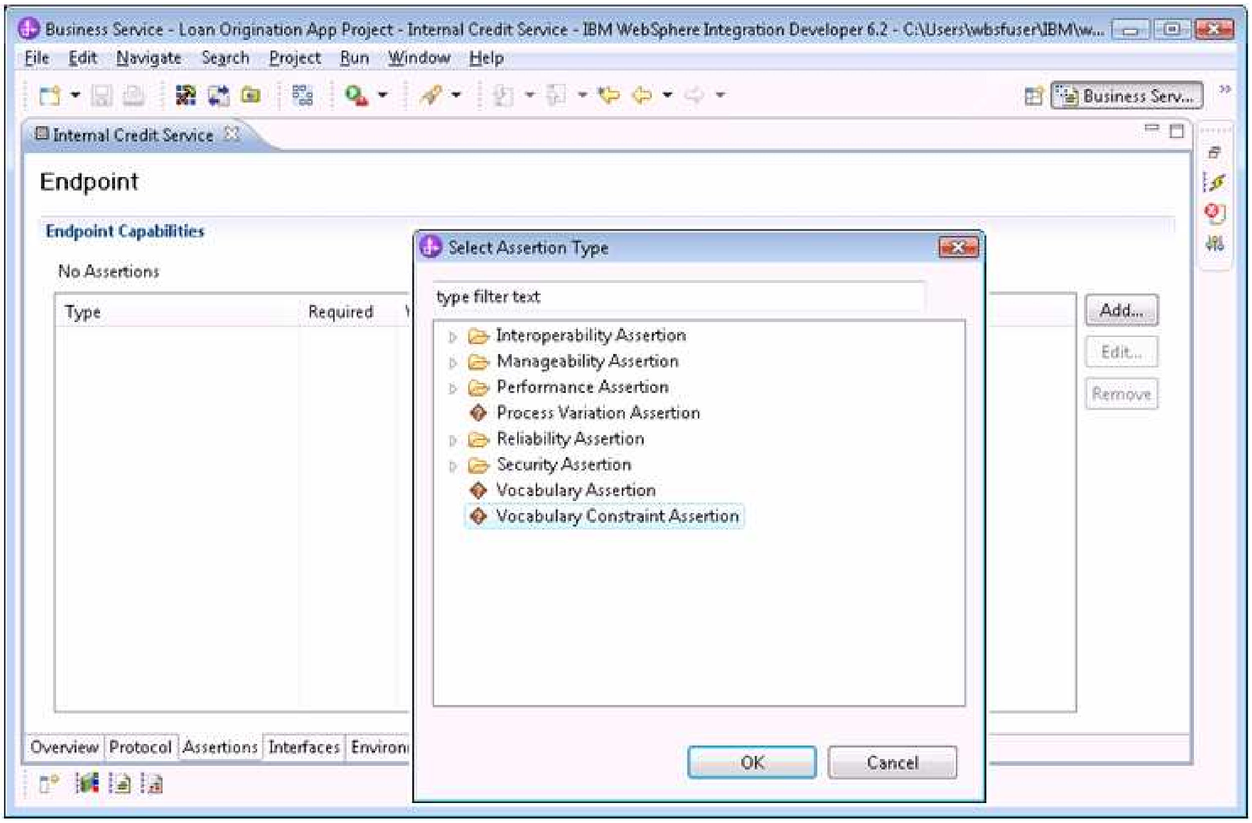
0 Interfaces

Add Existing...
Overview Protocol Assertions Interf
Interface Selection

Choose an Interface:

CheckCredit
FinalReview
getCreditReport
LoanOrigination
NotifyCustomer
ReviewCollateral

5. Go to the Assertions tab, click the Add button to add a new assertion, select Vocabulary Constraint Assertion, and click OK.


Business Service - Loan Origination App Project - Internal Credit Service - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\w...

Endpoint

Endpoint Capabilities

No Assertions

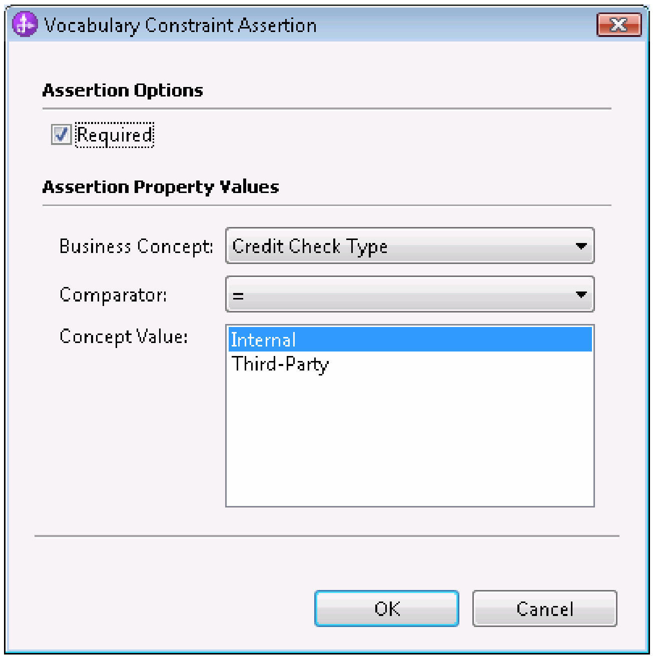
Type	Required

Select Assertion Type

type filter text

- Interoperability Assertion
- Manageability Assertion
- Performance Assertion
- Process Variation Assertion
- Reliability Assertion
- Security Assertion
- Vocabulary Assertion
- Vocabulary Constraint Assertion

6. Specify the constraint: Credit Check Type = Internal. Select the Required option, click OK.


Vocabulary Constraint Assertion

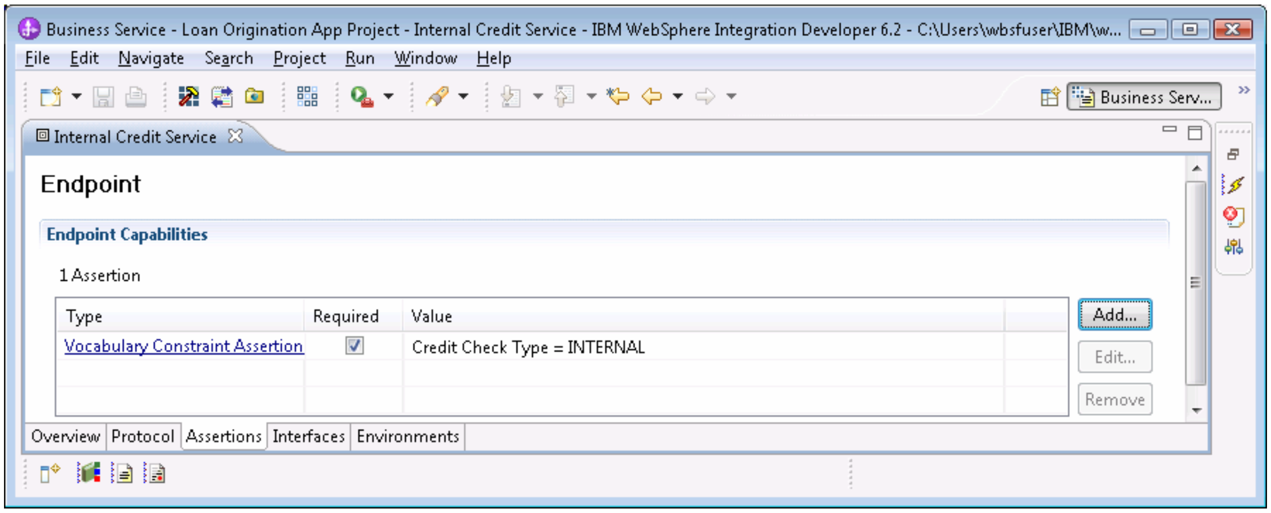
Assertion Options

[✓] Required

Assertion Property Values

Business Concept:	Credit Check Type
Comparator:	=
Concept Value:	Internal (selected)
	Third-Party

The resulting assertions should appear as follows:


Business Service - Loan Origination App Project - Internal Credit Service - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\w...

Internal Credit Service

Endpoint

Endpoint Capabilities

1 Assertion

Type	Required	Value
Vocabulary Constraint Assertion	✓	Credit Check Type = INTERNAL

7. Follow the same steps to create an endpoint for the Experian credit service. Use settings:

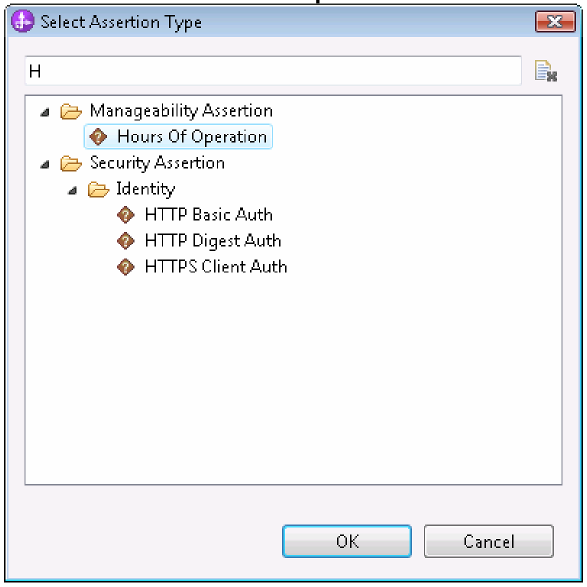
Name	Experian Credit Service
Interfaces	getCreditReport
Address	http://<server>:<port>/BetterFinancialsBackendWeb/sca/ExperianCreditCheck
Assertions	Vocabulary Constraint Assertion : Credit Check Type = Third-Party, Required Vocabulary Assertion: Premium Credit Check = True

8. Follow the same steps to create an endpoint for the TransUnion credit service. Use settings:

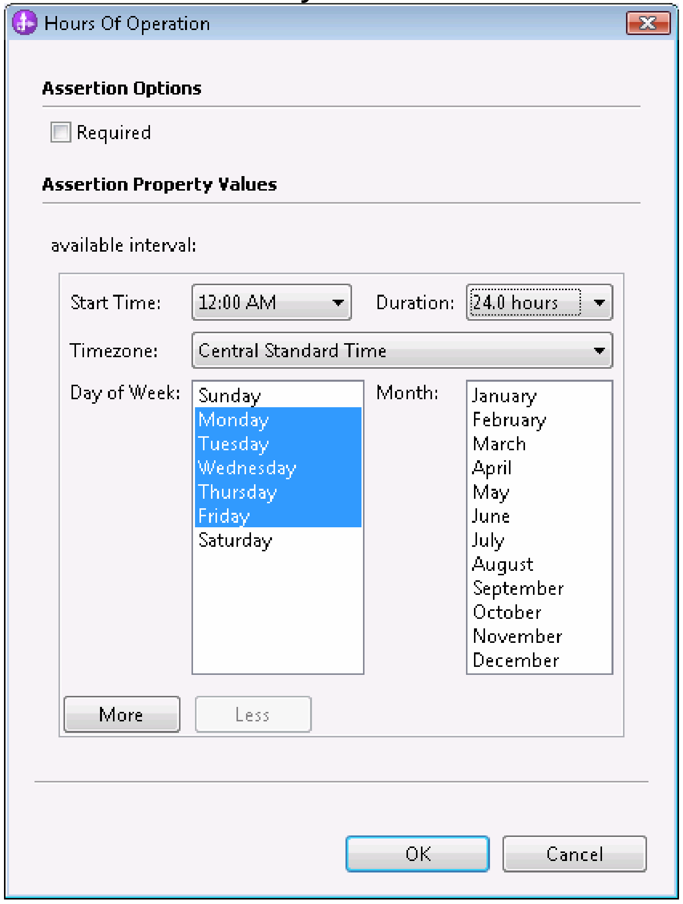
Name	TransUnion Credit Service
Interfaces	getCreditReport
Address	http://localhost:<port>/BetterFinancialsBackendWeb/sca/TransUnionCreditCheck
Assertions	Vocabulary Constraint Assertion : Credit Check Type = Third-Party, Required Vocabulary Assertion: Premium Credit Check = False

The TransUnion service is only available on weekdays. You will use an Hours of Operation assertion to describe when this endpoint is considered as available.

9. Add an Hours of Operation Assertion.

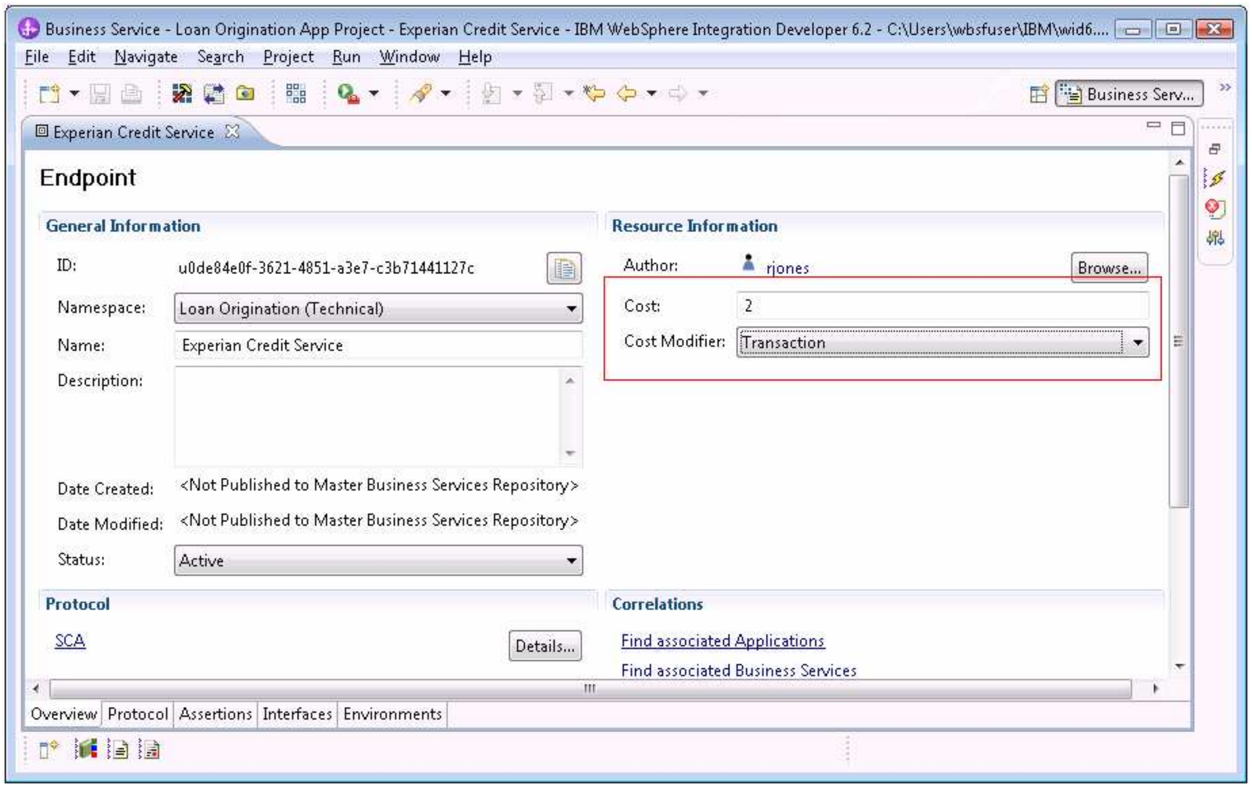


10. Set the start time to 12AM, Duration to 24 hours, and select Mon – Friday.



The Experian Credit service is a premium service, but it also costs more to use this service than it does to invoke the TransUnion service. Use the cost field to establish a cost-based priority on the endpoints.

11. Open the Experian Credit Service endpoint. Set the cost field to two and the cost modifier to "Transaction".


Business Service - Loan Origination App Project - Experian Credit Service - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\wid6....
File Edit Navigate Search Project Run Window Help
Business Serv...
Experian Credit Service

Endpoint

General Information

ID:	u0de84e0f-3621-4851-a3e7-c3b71441127c
Namespace:	Loan Origination (Technical)
Name:	Experian Credit Service
Description:	
Date Created:	<Not Published to Master Business Services Repository>
Date Modified:	<Not Published to Master Business Services Repository>
Status:	Active

Resource Information

Author:	rjones
Cost:	2
Cost Modifier:	Transaction

Browse...
Protocol
SCA
Details...
Correlations
Find associated Applications
Find associated Business Services
Overview Protocol Assertions Interfaces Environments

12. Edit the TransUnion endpoint to set a cost of one per transaction.

Testing the choose credit check decision

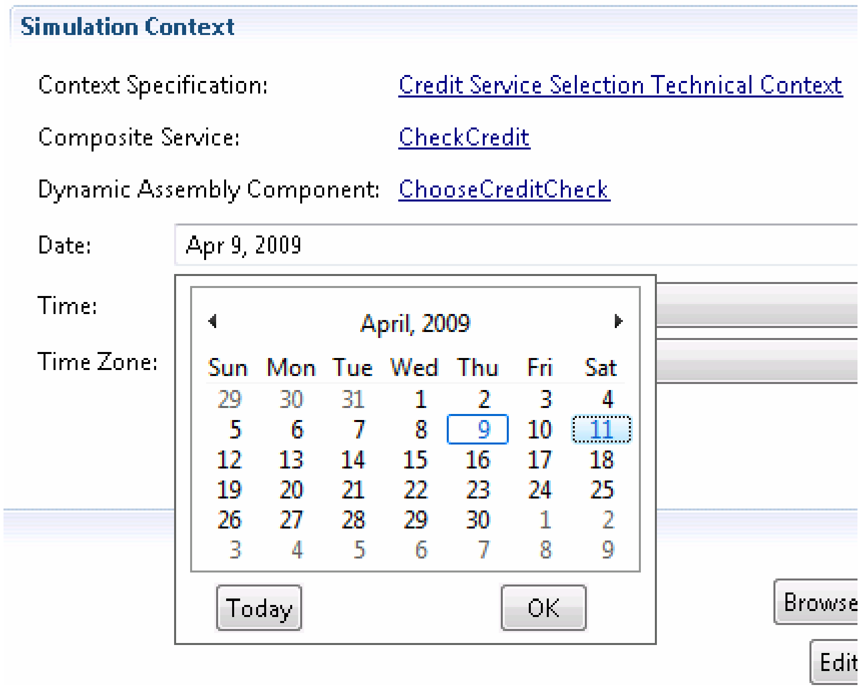
You will need to create a few more simulations to test your policies and other configuration for the ChooseCreditCheck decision.

Create these simulations for the ChooseCreditCheck dynamic assembly component.

Scenario	Description
Select Internal Credit	Credit Check Type = Internal New Customer = False --------------------------------- Select Internal Endpoint
Select Premium, Third-Party Credit	Credit Check Type = Third-Party Premium Service = True New Customer = True ---------------------------- Select Experian
Select Normal Credit Service(Weekdays)	Day of Week = Mon-Friday Credit Check Type = Third-Party Premium Credit Service = (not supplied) New Customer =True --- Select TransUnion
Select Normal Credit Service (Weekend)	Day of Week = Saturday or Sunday Credit Check Type = Third-Party Premium Credit Service = (not supplied) New Customer =True

By default, a new simulation will fill in the time and date fields with the current date and time. In situation, where it is important to test hours of operation or other temporal concerns such as the effective date range for policies, use the date and time controls to specify exactly what time/date is being simulated.

For example, you can test a weekend request by picking any Saturday on the calendar:



Simulation Context

Context Specification: Credit Service Selection Technical Context

Composite Service: CheckCredit

Dynamic Assembly Component: ChooseCreditCheck

Date: Apr 9, 2009

Time:

Time Zone:

April, 2009

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

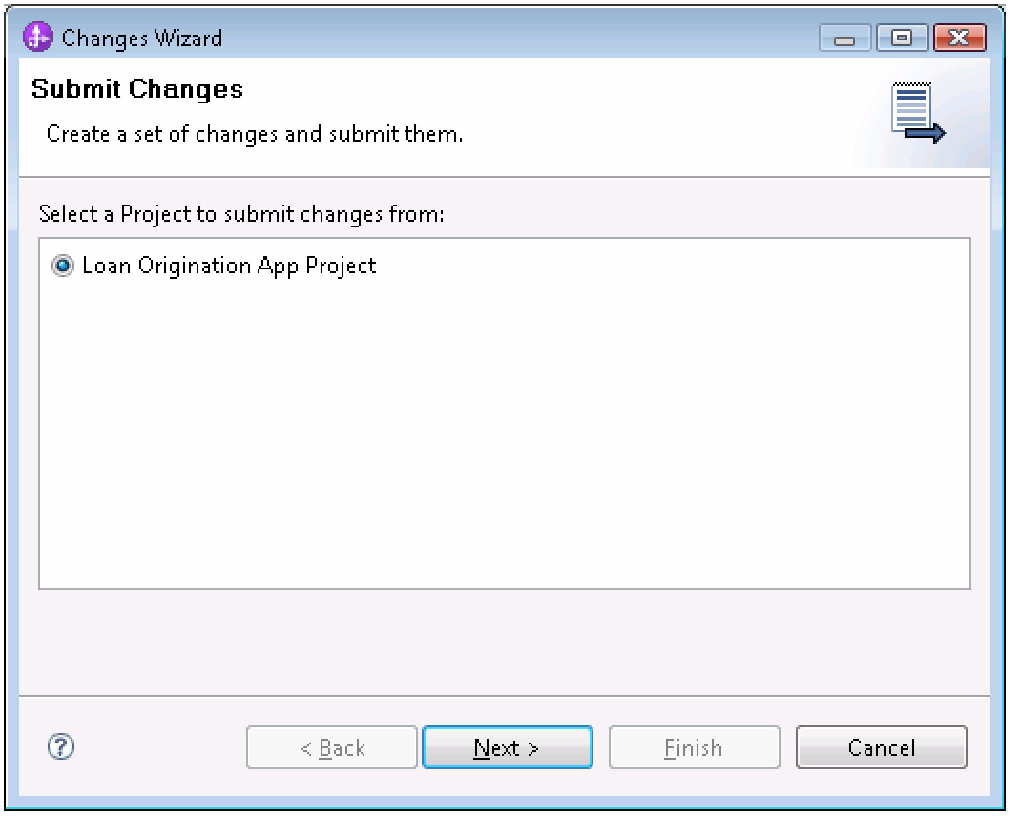
Submitting changes for approval

You are done configuring the necessary metadata and testing of endpoint selection. Now you need to submit your changes to the Fabric Server so that this configuration is available on the server at runtime.

A Fabric Tool Pack install configures a Fabric Server Unit Test Environment to automatically approve and publish changes submitted from

Composition Studio. For the purpose of this guide, it will show the full process.

1. Right click the Fabric Project in the Repository Changes View, or in the Business Application Explorer and select “Submit Changes”.



Submit Changes

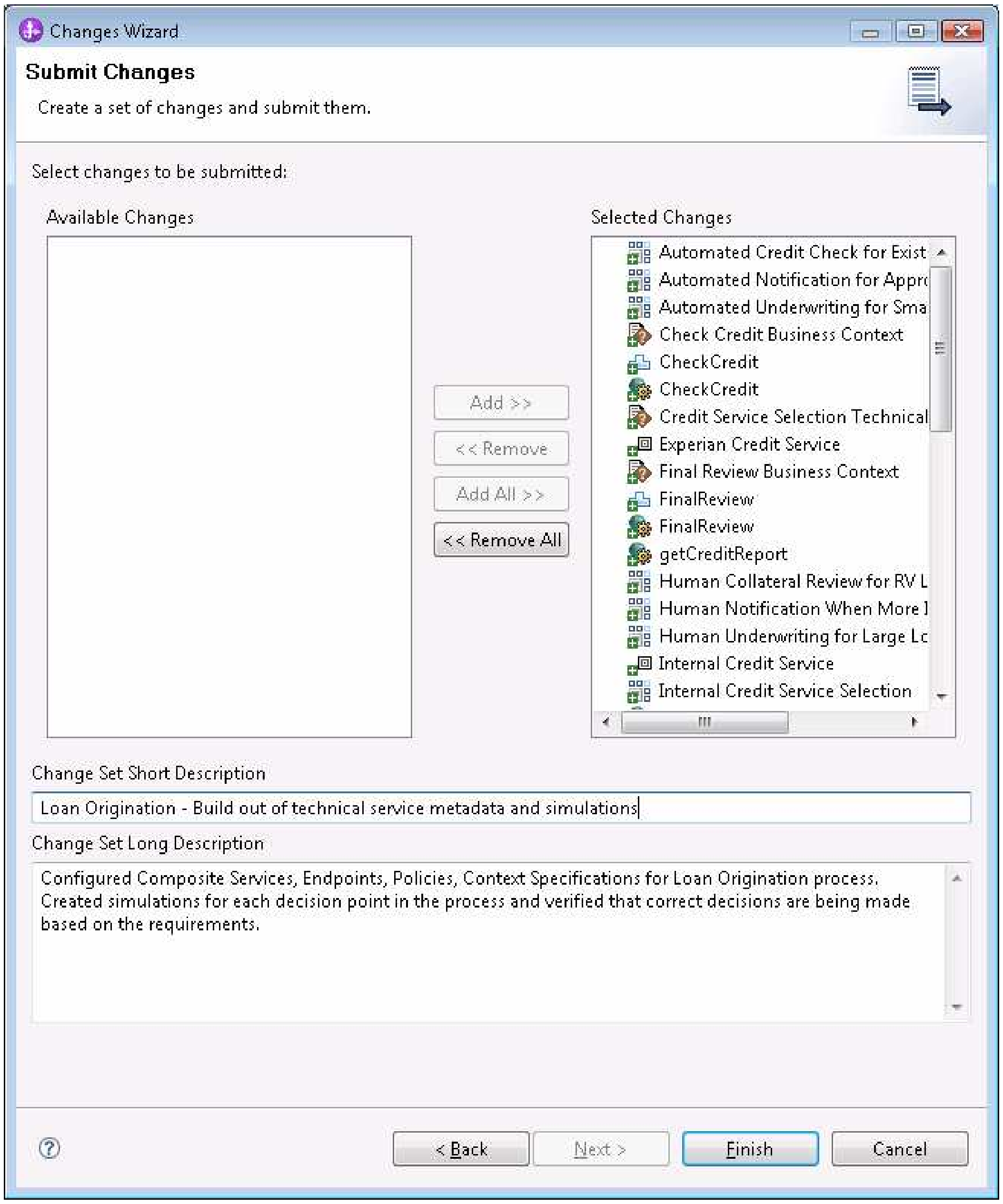
Create a set of changes and submit them.

Select a Project to submit changes from:

(●) Loan Origination App Project

   Select the project, and click Next.

2. Click the “Add All” button to select all changes. Provide a short description of the changes for the reviewer. Click Finish.



Automated Credit Check for Exist
Automated Notification for Appro
Automated Underwriting for Sma
Check Credit Business Context
CheckCredit
CheckCredit
Credit Service Selection Technical
Experian Credit Service
Final Review Business Context
FinalReview
FinalReview
getCreditReport
Human Collateral Review for RV L
Human Notification When More I
Human Underwriting for Large Lo
Internal Credit Service
Internal Credit Service Selection

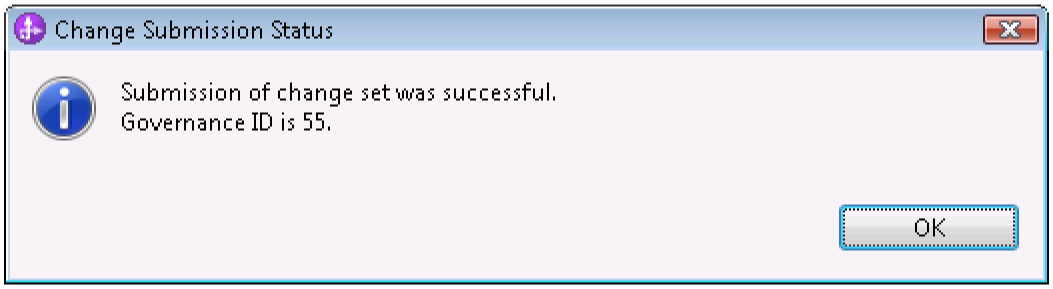
Change Set Short Description

Loan Origination - Build out of technical service metadata and simulations

Change Set Long Description

Configured Composite Services, Endpoints, Policies, Context Specifications for Loan Origination process. Created simulations for each decision point in the process and verified that correct decisions are being made based on the requirements.

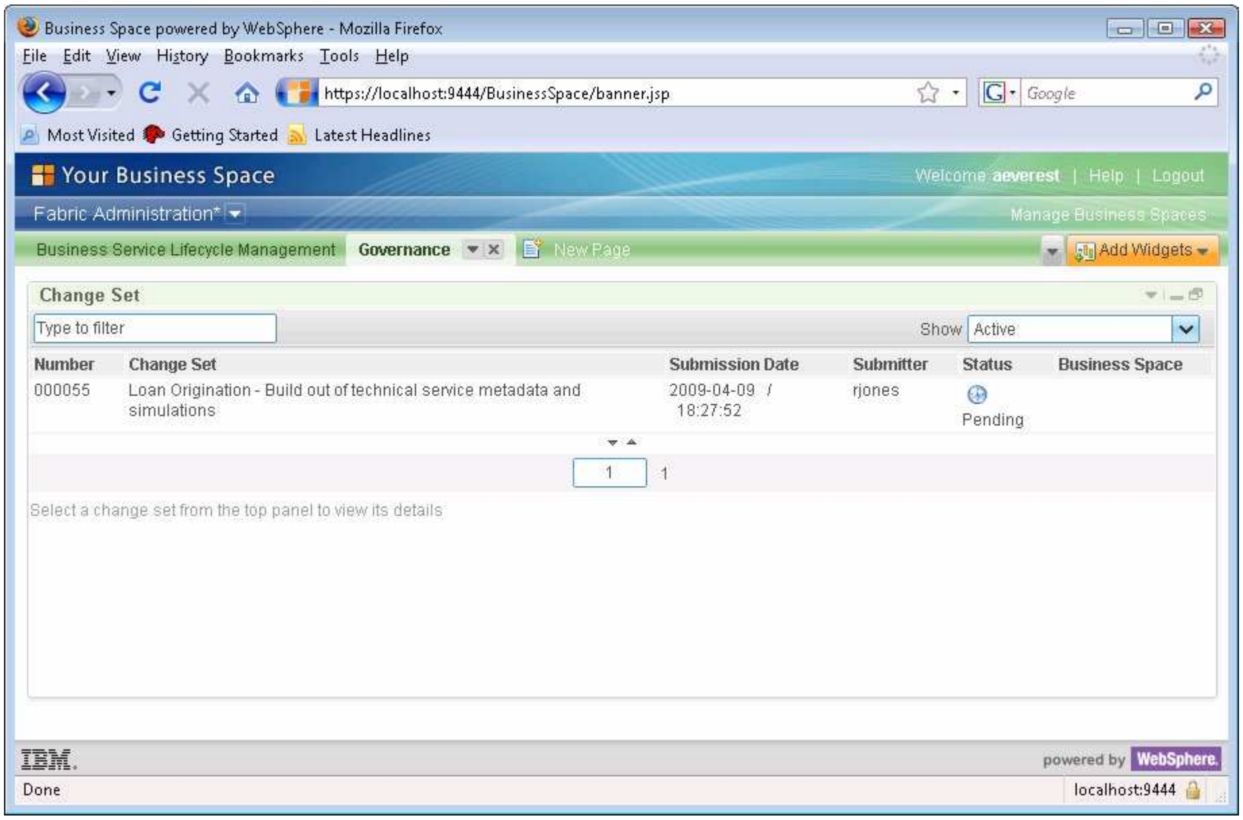
3. A dialog notifies the successful submission of the change set and provides a numeric ID for the change set.


Change Submission Status

Submission of change set was successful.
Governance ID is 55.

After submitting the changes, all objects in the change set become read-only until an update tells the system that the change set has been published or that the changes were rejected.

4. Your Administrator, Andy, logs into Business Space and opens the governance tab of the Fabric Administration space. The change set submitted from Studio should appear as an active change set with a status of Pending.

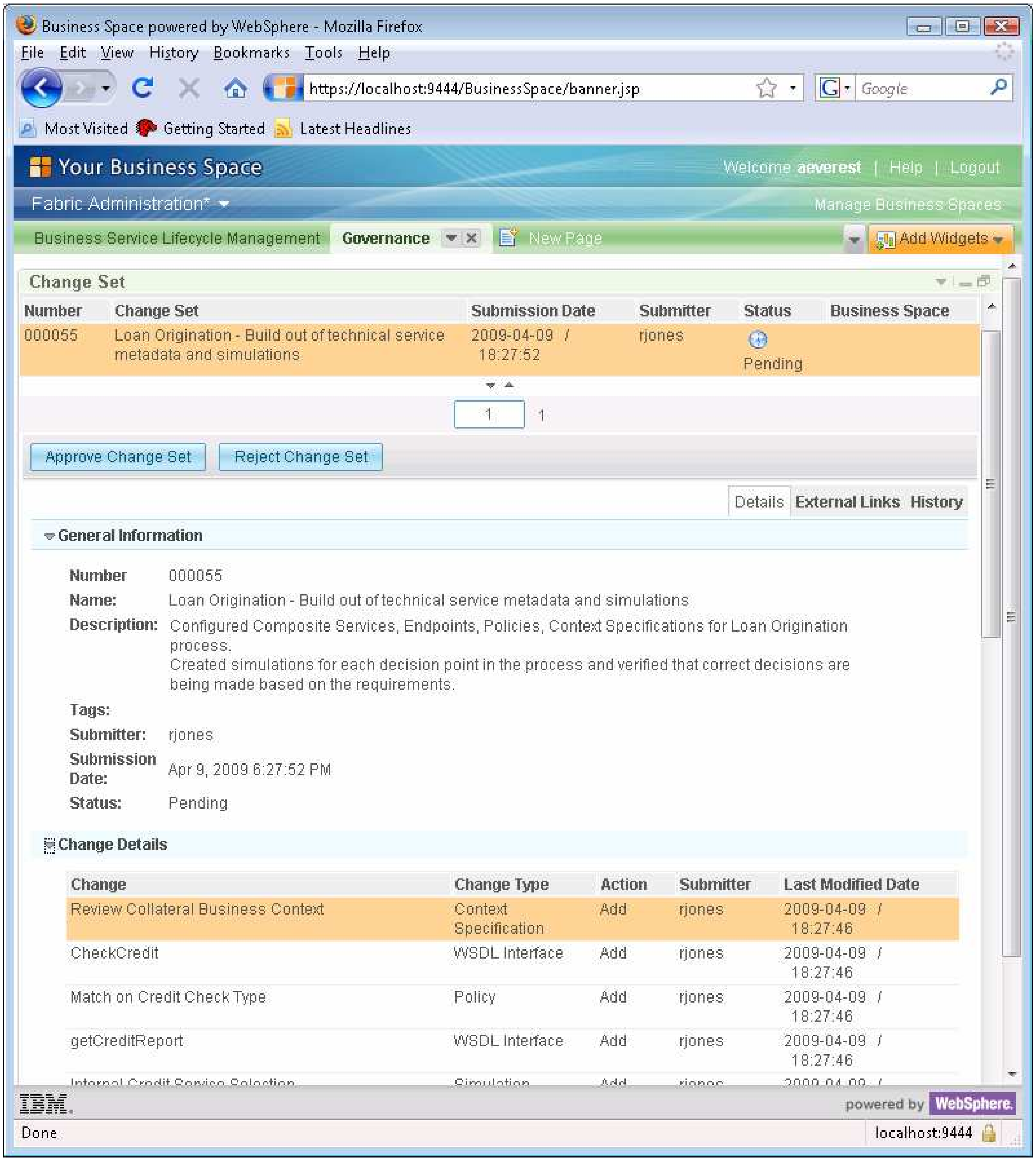

Business Space powered by WebSphere - Mozilla Firefox

https://localhost:9444/BusinessSpace/banner.jsp

Number	Change Set	Submission Date	Submitter	Status	Business Space
000055	Loan Origination - Build out of technical service metadata and simulations	2009-04-09 / 18:27:52	rjones	Pending	

Select a change set from the top panel to view its details

5. Open the change set and review the details.


Business Space powered by WebSphere - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://localhost:9444/BusinessSpace/banner.jsp
Most Visited Getting Started Latest Headlines
Your Business Space
Welcome aeverest | Help | Logout
Fabric Administration*
Manage Business Spaces
Business Service Lifecycle Management
Governance
New Page
Add Widgets
Change Set
Approve Change Set
Reject Change Set
Details
External Links
History
Done
localhost:9444

Number	Change Set	Submission Date	Submitter	Status	Business Space
000055	Loan Origination - Build out of technical service metadata and simulations	2009-04-09 / 18:27:52	rjones	Pending	

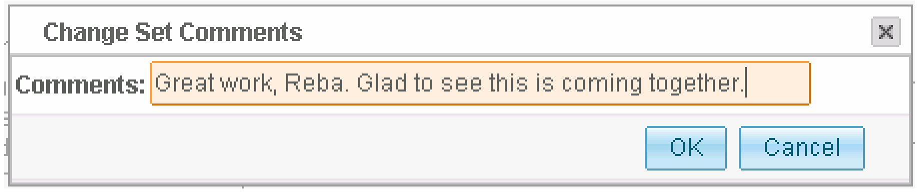
General Information

Number	000055
Name:	Loan Origination - Build out of technical service metadata and simulations
Description:	Configured Composite Services, Endpoints, Policies, Context Specifications for Loan Origination process. Created simulations for each decision point in the process and verified that correct decisions are being made based on the requirements.
Tags:	
Submitter:	rjones
Submission Date:	Apr 9, 2009 6:27:52 PM
Status:	Pending

Change Details

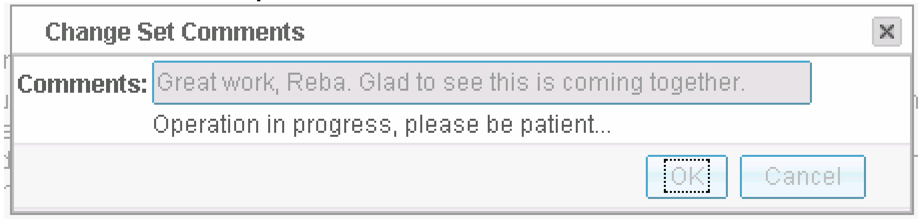
Change	Change Type	Action	Submitter	Last Modified Date
Review Collateral Business Context	Context Specification	Add	rjones	2009-04-09 / 18:27:46
CheckCredit	WSDL Interface	Add	rjones	2009-04-09 / 18:27:46
Match on Credit Check Type	Policy	Add	rjones	2009-04-09 / 18:27:46
getCreditReport	WSDL Interface	Add	rjones	2009-04-09 / 18:27:46
Internal Credit Service Selection	Simulation	Add	rjones	2009-04-09 /

6. Click the Approve Change set button and provide a comment.



Great work, Reba. Glad to see this is coming together.

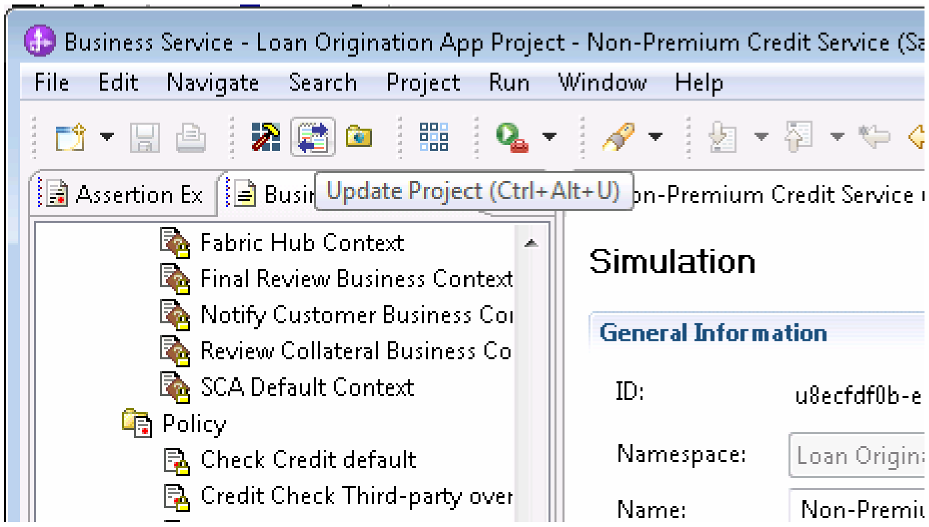
7. Click the publish button.



Great work, Reba. Glad to see this is coming together.

Operation in progress, please be patient...

8. Back in WebSphere Integration Developer, click the Update Project button.

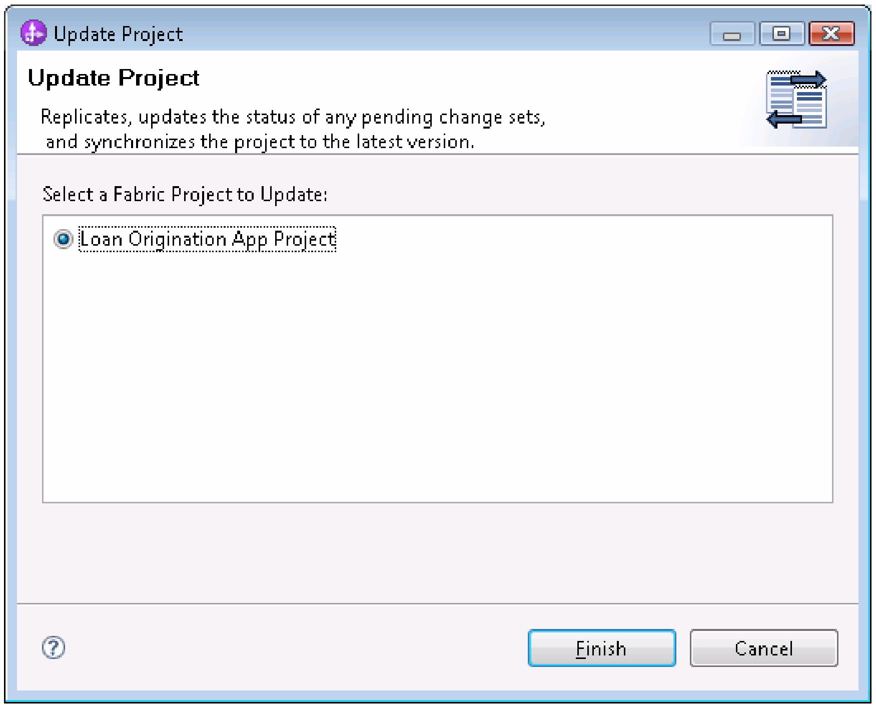

Business Service - Loan Origination App Project - Non-Premium Credit Service (Sa
File Edit Navigate Search Project Run Window Help
Assertion Ex
Busi
Update Project (Ctrl+Alt+U)
n-Premium Credit Service
Fabric Hub Context
Final Review Business Context
Notify Customer Business Co
Review Collateral Business Co
SCA Default Context
Policy
Check Credit default
Credit Check Third-party over
Simulation
General Information
ID:
Namespace:
Name:

u8ecfdf0b-e

Loan Origin

Non-Premi

9. Select the project to be updated, click Finish.



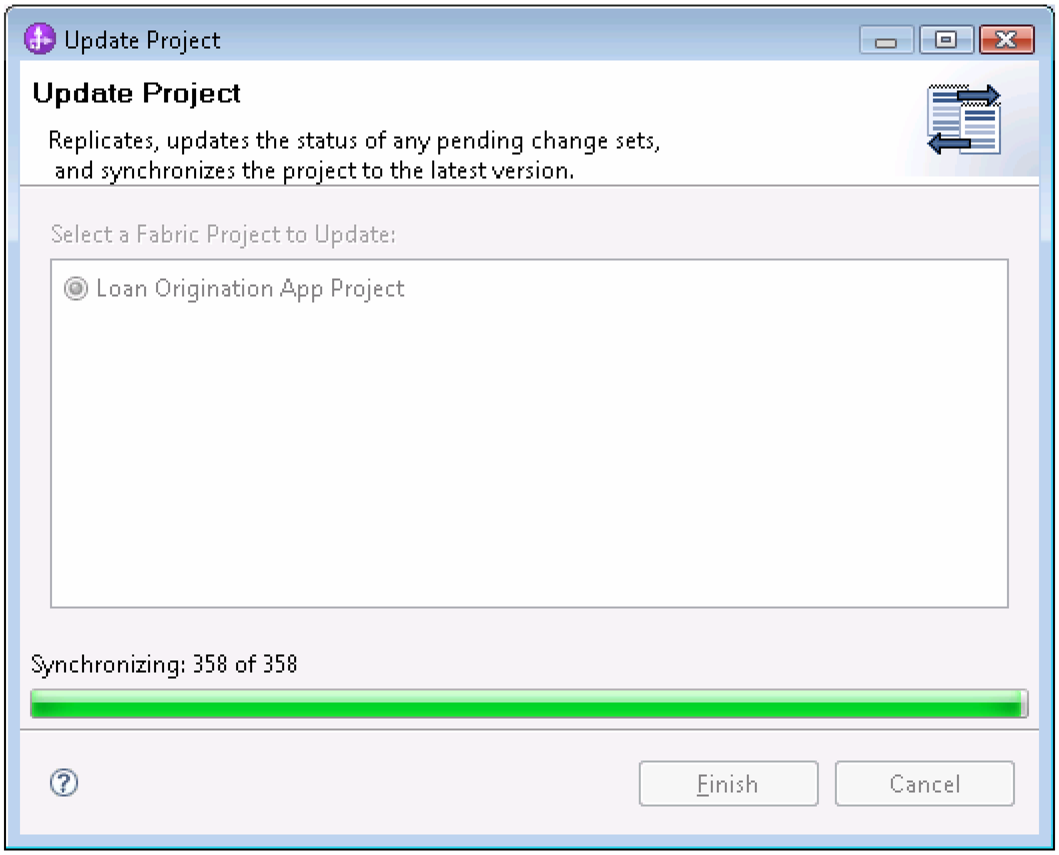
Update Project

Replicates, updates the status of any pending change sets, and synchronizes the project to the latest version.

Select a Fabric Project to Update:

[✓ Loan Origination App Project]

The system will first replicate any changes from the server. These changes can include your own change submission, or changes made by other users.


Update Project

Update Project

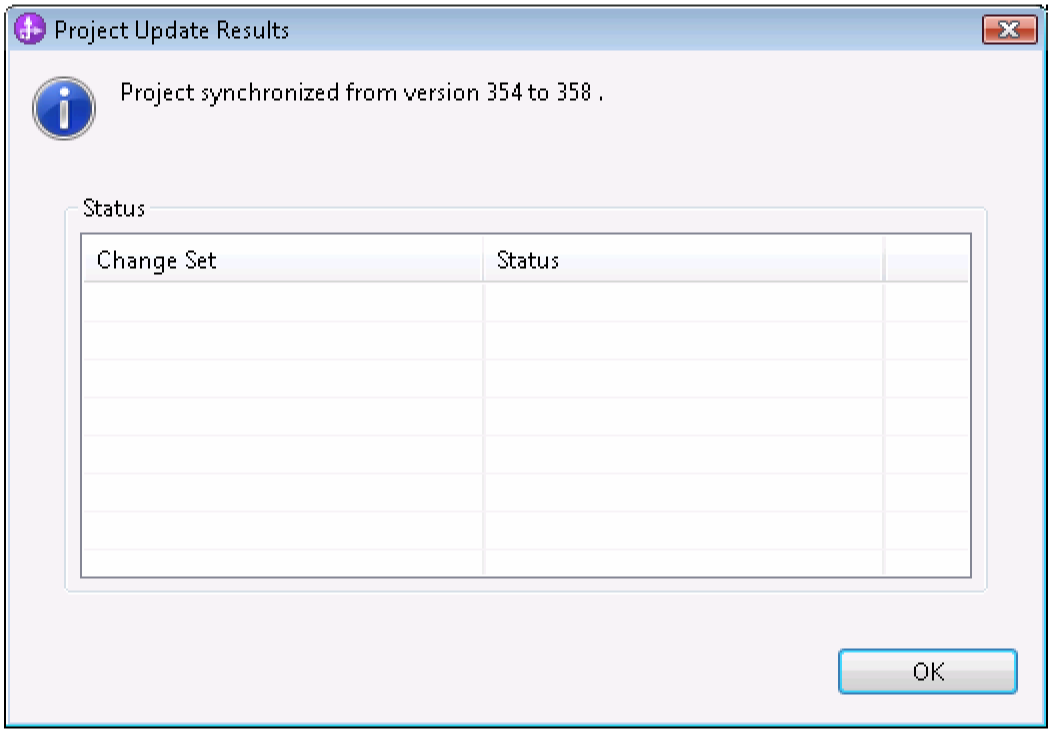
Replicates, updates the status of any pending change sets, and synchronizes the project to the latest version.

Select a Fabric Project to Update:

● Loan Origination App Project

Synchronizing: 358 of 358

The system tells you what version of the repository the project is now synchronized with and provides the current status of any outstanding change sets.

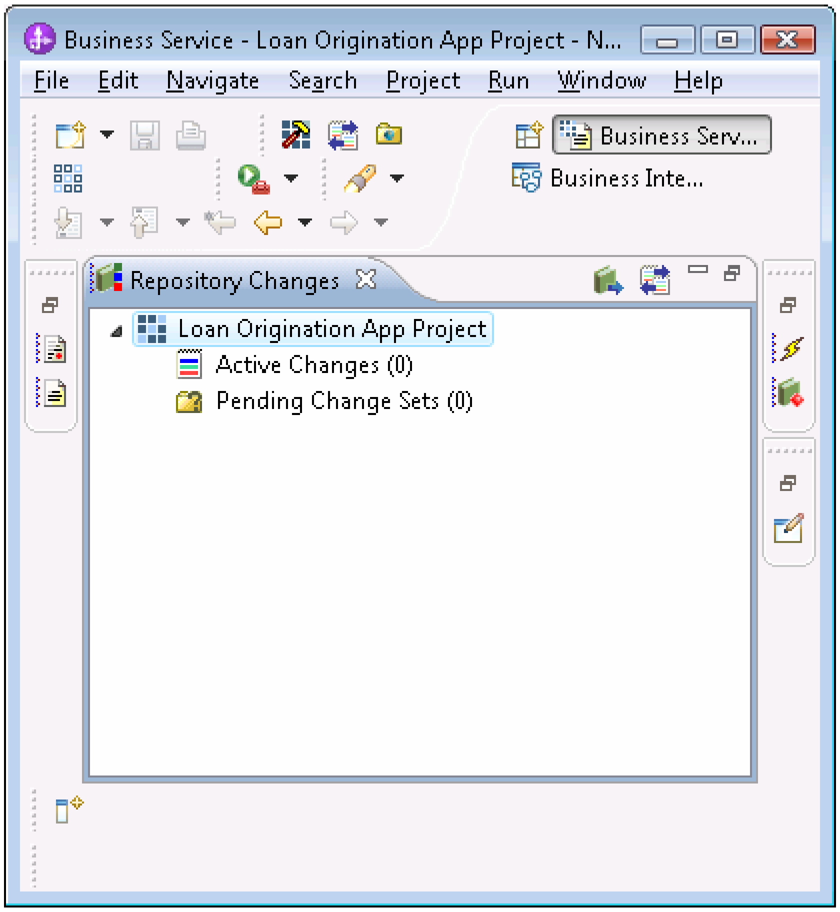

Project Update Results

Project synchronized from version 354 to 358 .

Status

Change Set	Status

After performing the update, the pending change set disappears since the project has now been updated to incorporate changes that took place on the server. Objects that were read-only while pending are unlocked so that further edits can take place.


Business Service - Loan Origination App Project - N...
File Edit Navigate Search Project Run Window Help
Business Serv...
Business Inte...
Repository Changes

Loan Origination App Project
Active Changes (0)
Pending Change Sets (0)

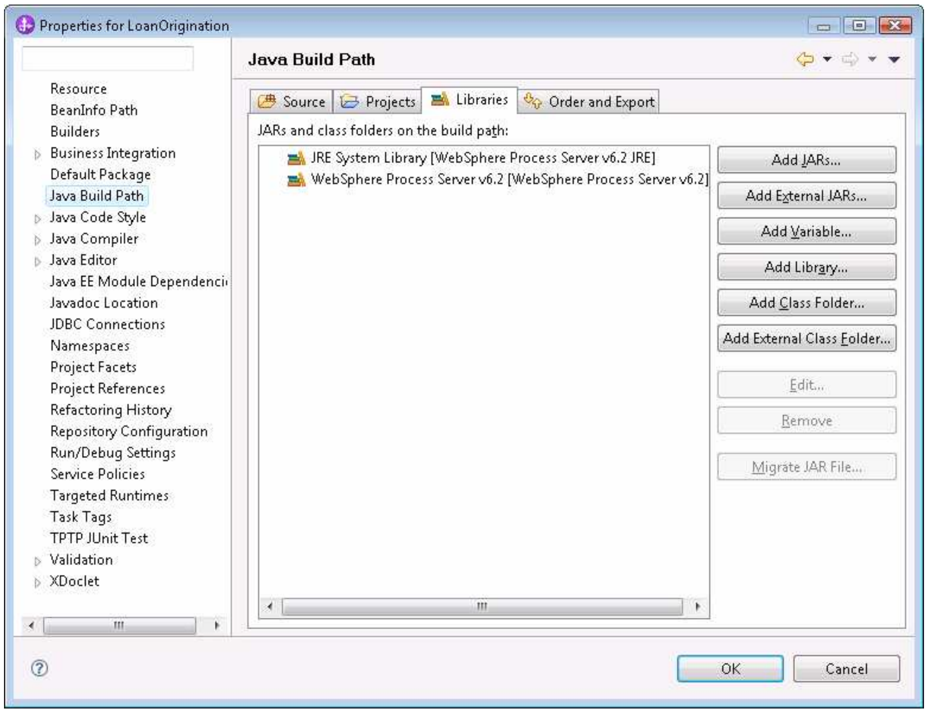
Next, you will need to switch to the Business Integration perspective to finish the assembly of your modules and perform some testing.

Implementing context extractors

A context extractor is a Java component wired to a dynamic assembler that implements the ContextExtractor interface. The ContextExtractor interface and other Fabric APIs can be used in Java programming only after Fabric libraries are added the project class path.

Extending the class path to include Fabric libraries

1. Right-click the LoanOrigination project and select Properties. Go to the Java Build Path property sheet.


Properties for LoanOrigination

Java Build Path

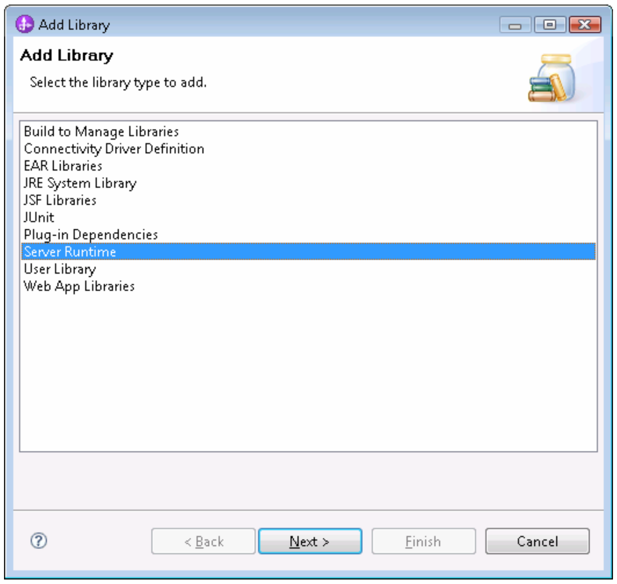
Resource
BeanInfo Path
Builders
Business Integration
Default Package
Java Build Path
Java Code Style
Java Compiler
Java Editor
Java EE Module Dependenci
Javadoc Location
JDBC Connections
Namespaces
Project Facets
Project References
Refactoring History
Repository Configuration
Run/Debug Settings
Service Policies
Targeted Runtimes
Task Tags
TPTP JUnit Test
Validation
XDoclet

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

JRE System Library [WebSphere Process Server v6.2 JRE]
WebSphere Process Server v6.2 [WebSphere Process Server v6.2]

2. Click Add Library… Choose Server Runtime. Click Next.

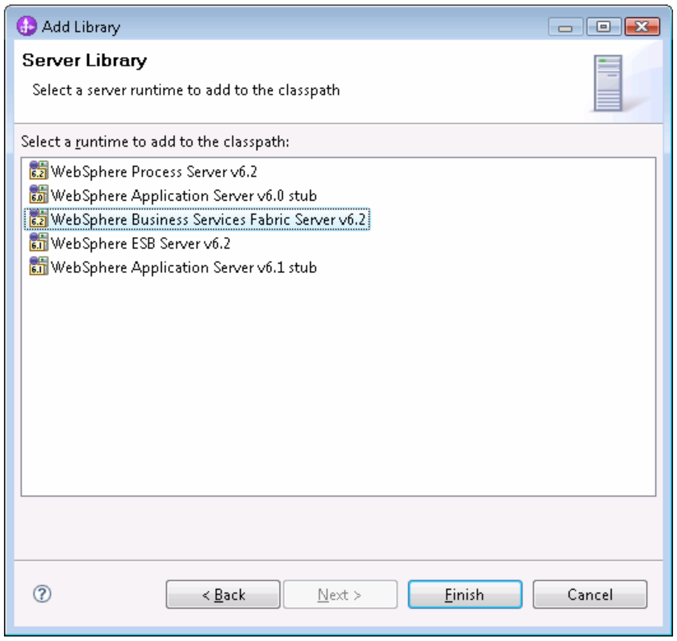

Add Library

Add Library

Select the library type to add.

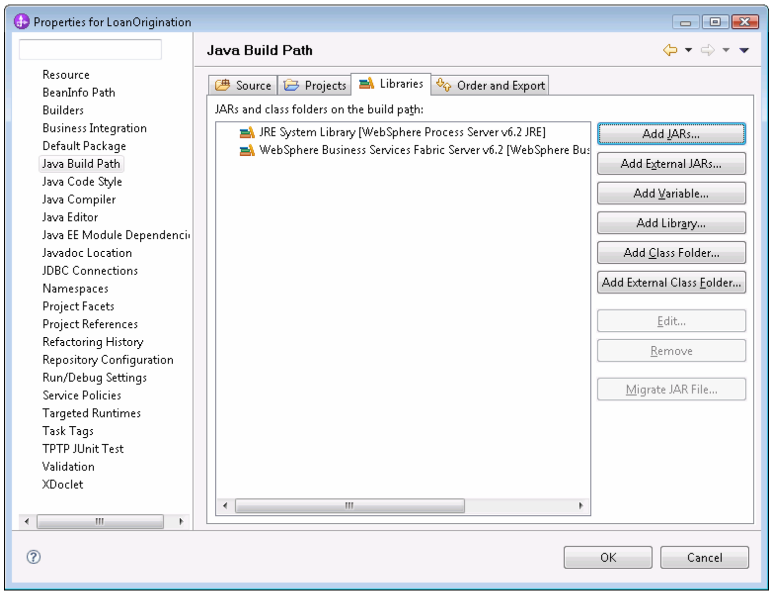
Build to Manage Libraries
Connectivity Driver Definition
EAR Libraries
JRE System Library
JSF Libraries
JUnit
Plug-in Dependencies
Server Runtime
User Library
Web App Libraries

3. Select WebSphere Business Services Fabric Server V6.2. Click Finish.


Add Library
Server Library
Select a server runtime to add to the classpath
Select a runtime to add to the classpath:
WebSphere Process Server v6.2
WebSphere Application Server v6.0 stub
WebSphere Business Services Fabric Server v6.2
WebSphere ESB Server v6.2
WebSphere Application Server v6.1 stub
< Back
Next >
Finish
Cancel

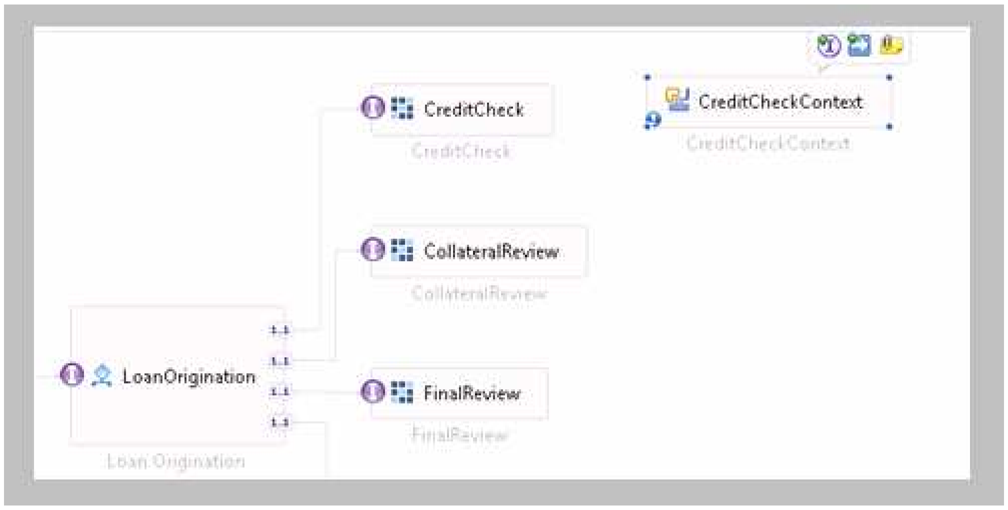
At this point, you will have both the Fabric and WebSphere Process Server libraries. Since the Fabric library is a super set of the default WebSphere Process Server library, you can remove that one.

4. Remove the WebSphere Process Server V6.2 library. Click Ok.

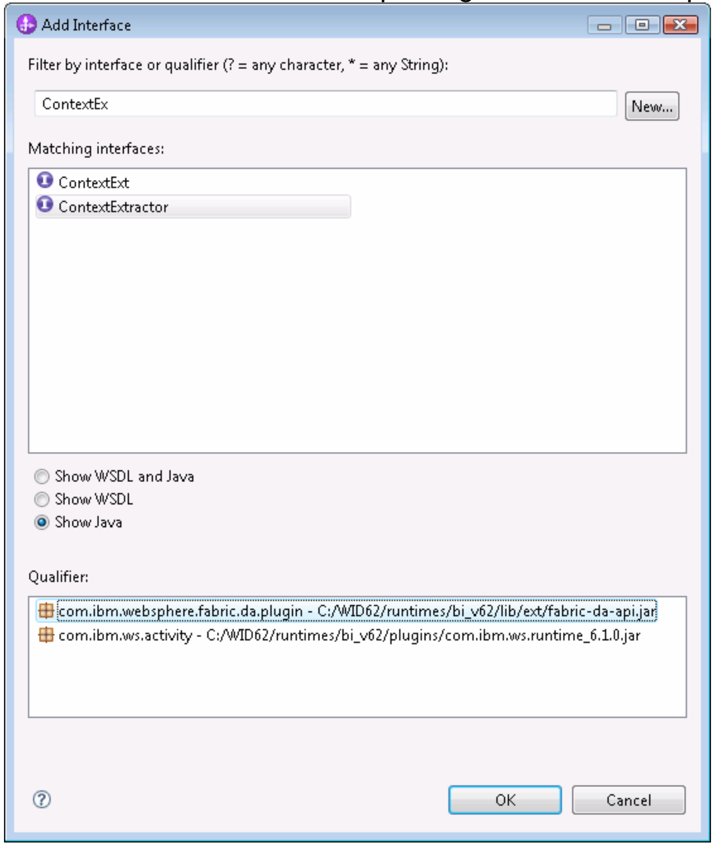


Implementing a context extractor for credit check

1. Select the Java Component type from the palette, drop a new component on the assembly, and edit the name to “CreditCheckContext”.



2. Click the add Interface button to define an interface for the new component. Use the “show Java” setting and the filter box to narrow the search for “ContextExtractor”. Select the interface with the package com.ibm.websphere.fabric.da.plugin. Click OK.



Add Interface

Filter by interface or qualifier (? = any character, * = any String):

ContextEx

Matching interfaces:

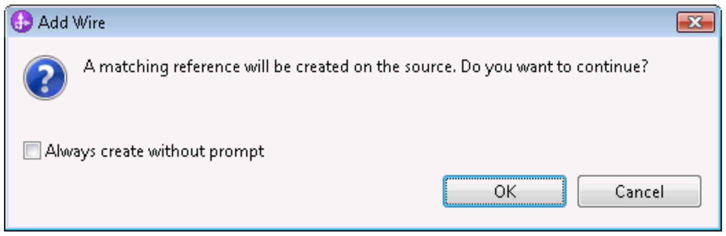
- ContextExt
- ContextExtractor

- Show WSDL and Java
- Show WSDL
- [✓ Show Java]

Qualifier:

- com.ibm.websphere.fabric.da.plugin - C:/WID62/runtimes/bi_v62/lib/ext/fabric-da-api.jar
- com.ibm.ws.activity - C:/WID62/runtimes/bi_v62/plugins/com.ibm.ws.runtime_6.1.0.jar

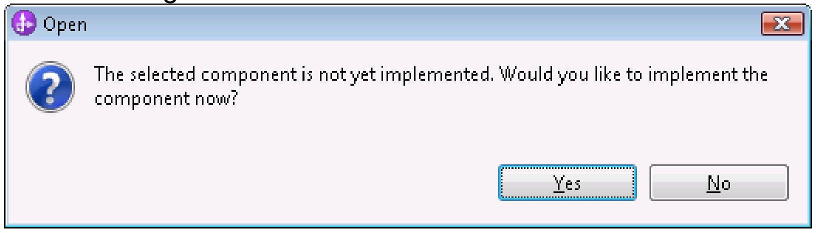
3. Draw a wire from the CreditCheck DA Component to the new Java component. Click OK if you see this dialog:


Add Wire

A matching reference will be created on the source. Do you want to continue?

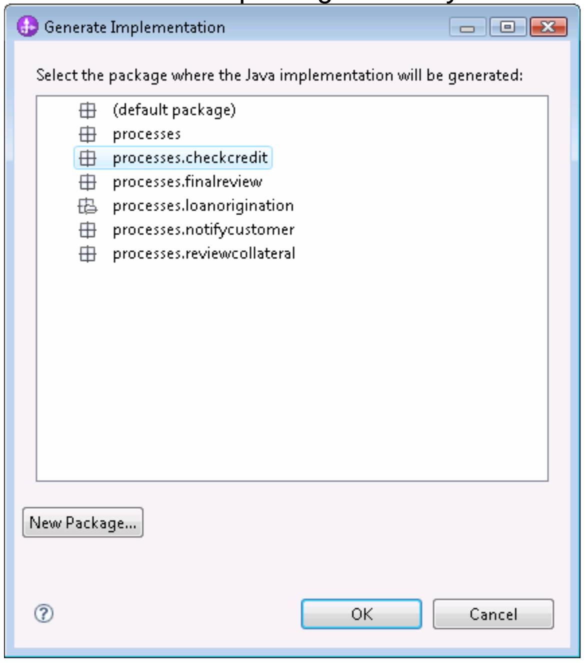
Always create without prompt

4. Double-click the Java component to generate an implementation. Answer Yes to this dialog:


Open

The selected component is not yet implemented. Would you like to implement the component now?

5. Select the package where you the Java code to belong.

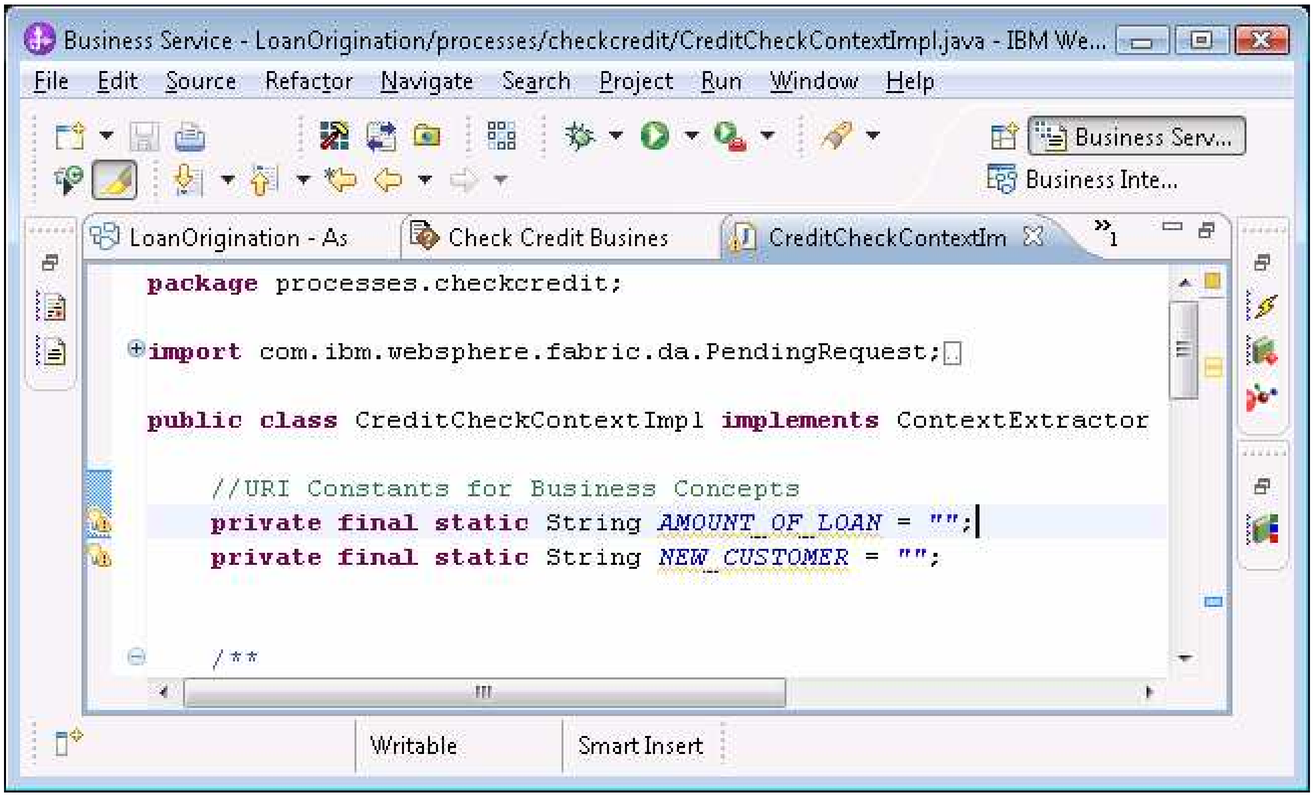

Generate Implementation

Select the package where the Java implementation will be generated:

(default package)
processes
processes.checkcredit
processes.finalreview
processes.loanorigination
processes.notifycustomer
processes.reviewcollateral

A skeleton implementation is created. You will need to fill in the implementation for the extractContext method. When setting properties in the context, it is necessary to use the full URI of a vocabulary concept as the property key. Based on the context specification you created, you know that you will need to provide values for New Customer and Amount of Loan.

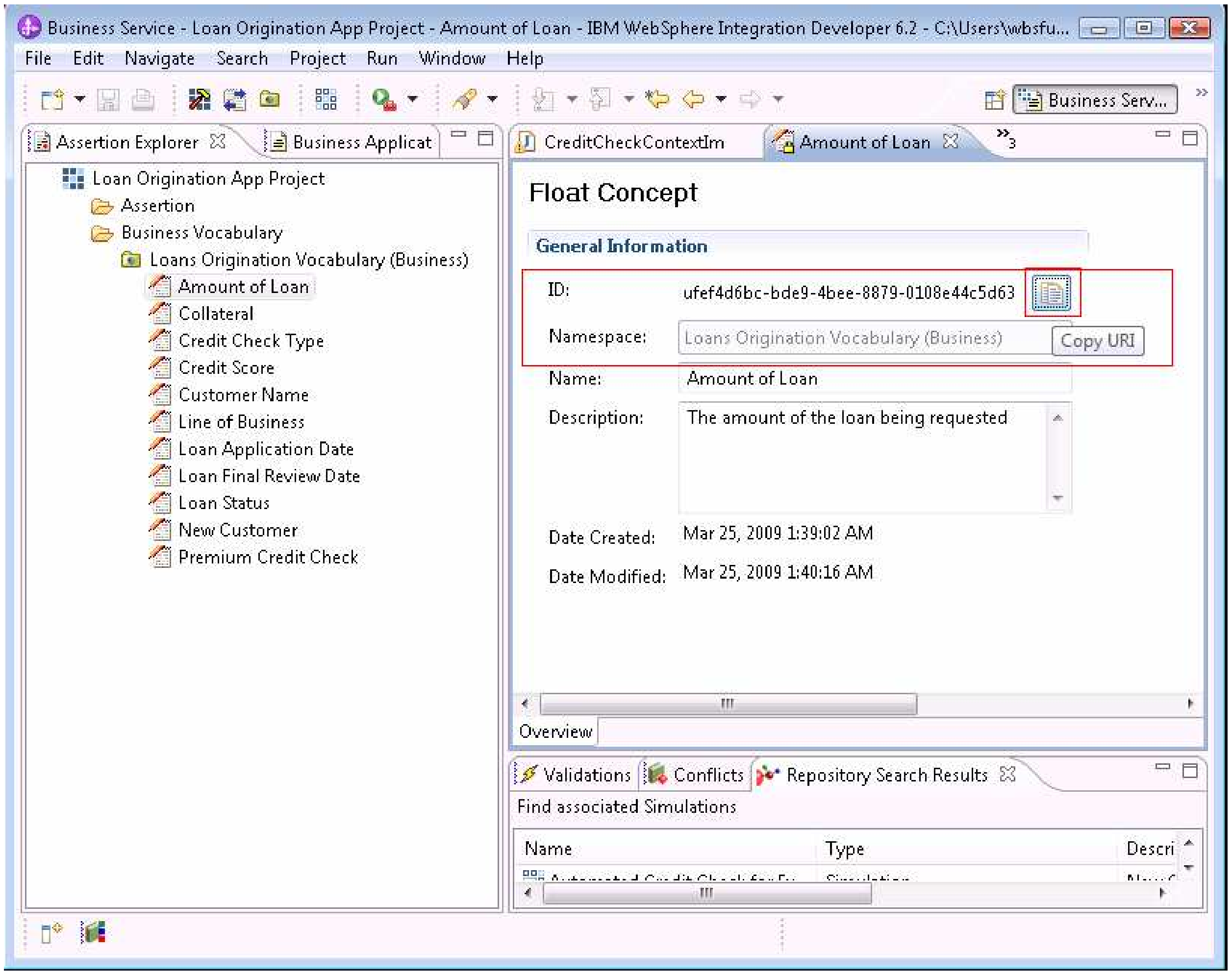
6. Define placeholder Java constants for the two URIs.


Business Service - LoanOrigination/processes/checkcredit/CreditCheckContextImpl.java - IBM We...
File Edit Source Refactor Navigate Search Project Run Window Help
Business Serv...
Business Inte...
LoanOrigination - As
Check Credit Busines
CreditCheckContextIm
Writable
Smart Insert

```
package processes.checkcredit;

import com.ibm.websphere.fabric.da.PendingRequest;

public class CreditCheckContextImpl implements ContextExtractor

    //URI Constants for Business Concepts
    private final static String AMOUNT_OF_LOAN = "";
    private final static String NEW_CUSTOMER = "";


    /**
```

The URI values can be found using the Assertion Explorer available in the Business Service perspective. If the Business Vocabulary folder in the explorer tree is empty, it probably means that the namespace import steps described at the end of Chapter five were not carried out. Use the Fabric administrative console to make the Loan Origination project import the Loan Vocabulary namespace from the Business Glossary project. Then update the Fabric project in studio to pull down these changes. Finally, for namespaces added after a fabric project was originally created, it is necessary to make the namespace visible by editing the properties of the fabric project in WebSphere Integration Developer. After these steps, the concepts in your imported vocabularies should display in the Assertion Explorer.

7. Switch to the Business Service perspective. In the Assertion Explorer view find and double-click the Amount of Loan concept to open its editor. Click the Copy URI button next to the ID field to put the full URI of the concept on your clipboard.



Business Service - Loan Origination App Project - Amount of Loan - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfu...

File Edit Navigate Search Project Run Window Help

Assertion Explorer | Business Applicat

- Loan Origination App Project
  - Assertion
  - Business Vocabulary
    - Loans Origination Vocabulary (Business)
      - Amount of Loan
      - Collateral
      - Credit Check Type
      - Credit Score
      - Customer Name
      - Line of Business
      - Loan Application Date
      - Loan Final Review Date
      - Loan Status
      - New Customer
      - Premium Credit Check

CreditCheckContextIm | Amount of Loan

Float Concept

General Information

ID: ufef4d6bc-bde9-4bee-8879-0108e44c5d63

Namespace: Loans Origination Vocabulary (Business) Copy URI

Name: Amount of Loan

Description: The amount of the loan being requested

Date Created: Mar 25, 2009 1:39:02 AM

Date Modified: Mar 25, 2009 1:40:16 AM

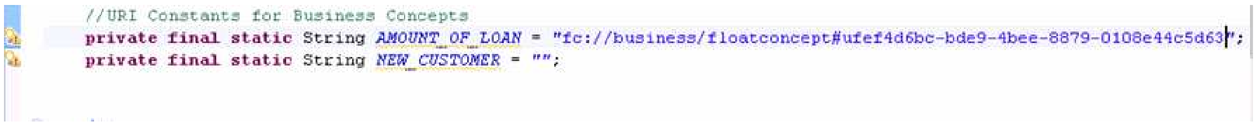
Overview

Validations | Conflicts | Repository Search Results

Find associated Simulations

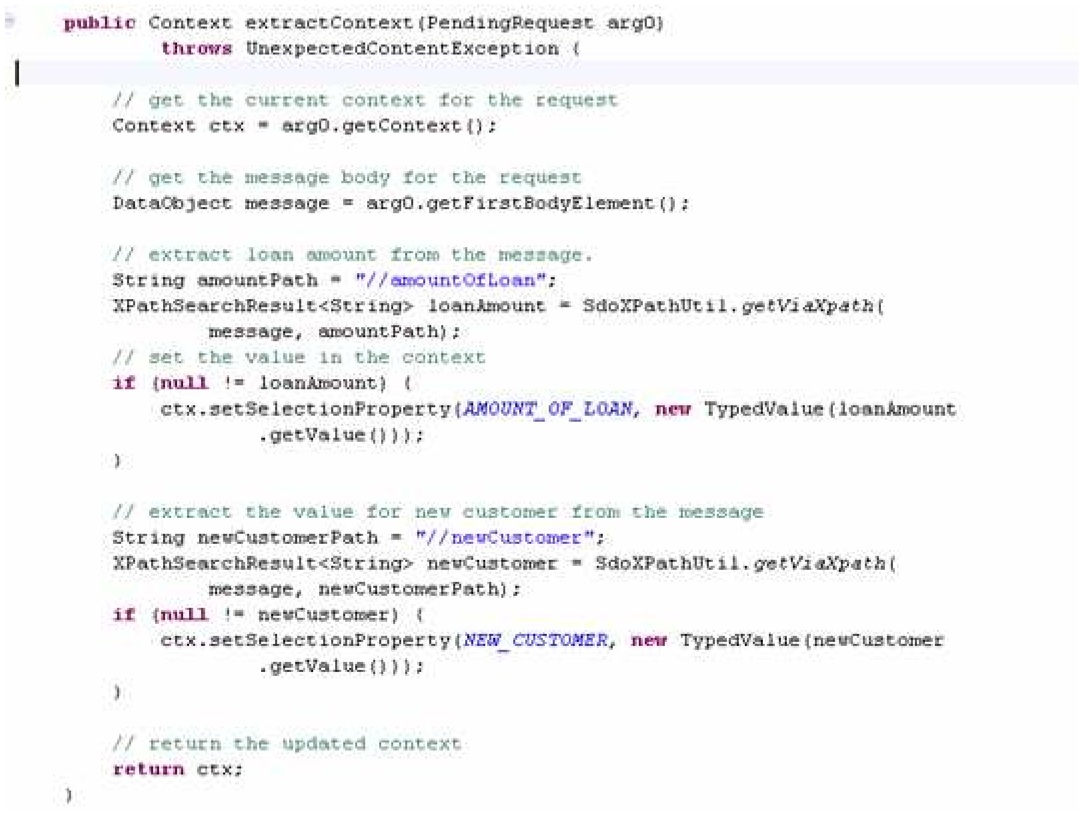
Name	Type	Descri

8. Paste the URI as the value for the constant you defined in the context extractor code. Repeat for the New Customer concept.



```
//URI Constants for Business Concepts
private final static String AMOUNT_OF_LOAN = "fc://business/floatconcept#ufef4d6bc-bde9-4bee-8879-0108e44c5d63";
private final static String NEW_CUSTOMER = "";
```

9. Fill in the implementation of extractContext().



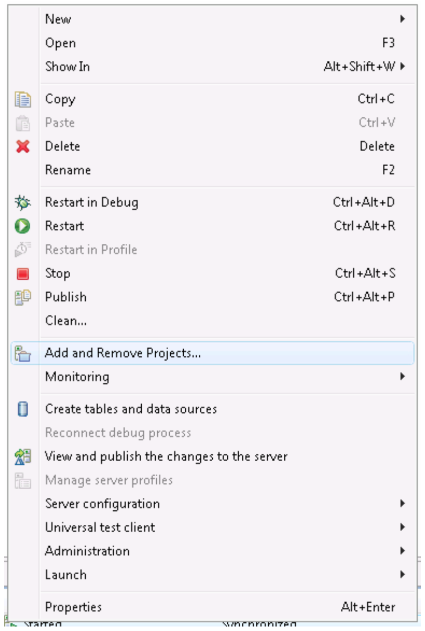
```
    public Context extractContext(PendingRequest arg0)
            throws UnexpectedContentException {

        // get the current context for the request
        Context ctx = arg0.getContext();

        // get the message body for the request
        DataObject message = arg0.getFirstBodyElement();

        // extract loan amount from the message.
        String amountPath = "//amountOfLoan";
        XPathSearchResult<String> loanAmount = SdoXPathUtil.getViaXpath(
                message, amountPath);
        // set the value in the context
        if (null != loanAmount) {
            ctx.setSelectionProperty(AMOUNT_OF_LOAN, new TypedValue(loanAmount
                    .getValue()));
        }

        // extract the value for new customer from the message
        String newCustomerPath = "//newCustomer";
        XPathSearchResult<String> newCustomer = SdoXPathUtil.getViaXpath(
                message, newCustomerPath);
        if (null != newCustomer) {
            ctx.setSelectionProperty(NEW_CUSTOMER, new TypedValue(newCustomer
                    .getValue()));
        }

        // return the updated context
        return ctx;
    }
```

The PendingRequest argument to extractContext () provides access to the current context and the request message. For each value that needs to be injected into the context, a call is made to ctx.setSelectionProperty() with the URI of the concept and a value from the message.

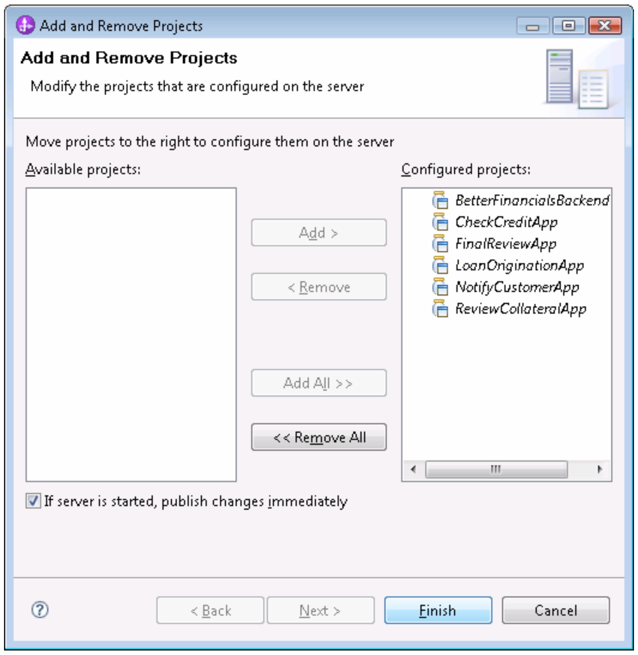
The helper class SDOXPathUtil makes allows the code to process XPath expressions against the message to extract values to be set into the context.

Deploying modules to the server

1. Go to the Servers View in the Business Integration perspective. Right-click the Fabric Server and select Add or Remove projects.

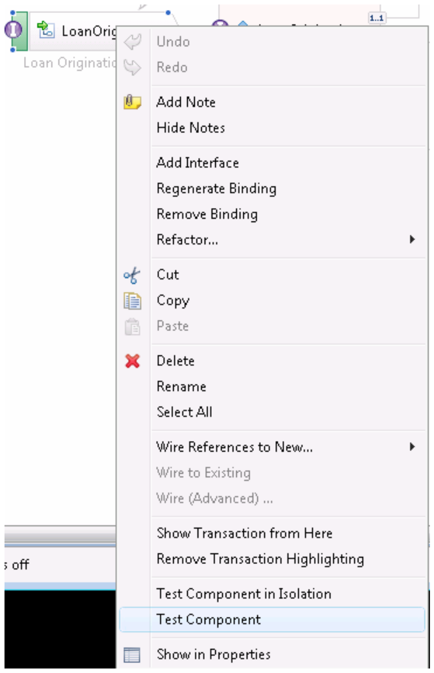


2. Click the Add All button to deploy all of the projects. Click Finish to deploy these applications on the server.



Testing round trips

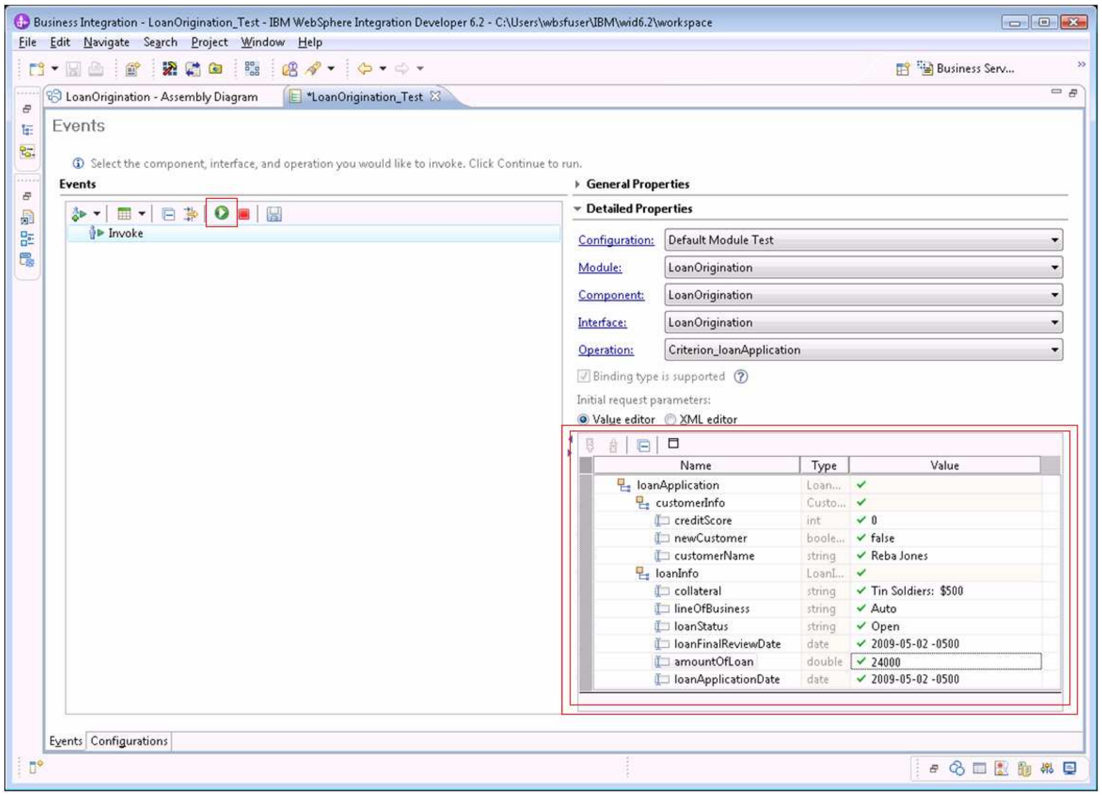
1. Right-click the LoanOrigination export in the LoanOrigination assembly diagram. Select Test Component.



2. Fill in the loan application request with these details and click the continue button.

Field	Value	Notes
customerName	Reba Jones	Customer seeking the loan.
newCustomer	True	New customers will end up using a vendor credit agency.
lineOfBusiness	Auto	Auto loans use the automated NoCollateralReview variation.
loanStatus	OPEN	Loans applications start in Open status.
amoutOfLoan	24000	Within the range allowed for

automated underwriting.


Business Integration - LoanOrigination_Test - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\wid6.2\workspace

File Edit Navigate Search Project Window Help

LoanOrigination - Assembly Diagram | *LoanOrigination_Test

Events

Select the component, interface, and operation you would like to invoke. Click Continue to run.

Events

Invoke

General Properties

Detailed Properties

Configuration: Default Module Test

Module: LoanOrigination

Component: LoanOrigination

Interface: LoanOrigination

Operation: Criterion_loanApplication

Binding type is supported

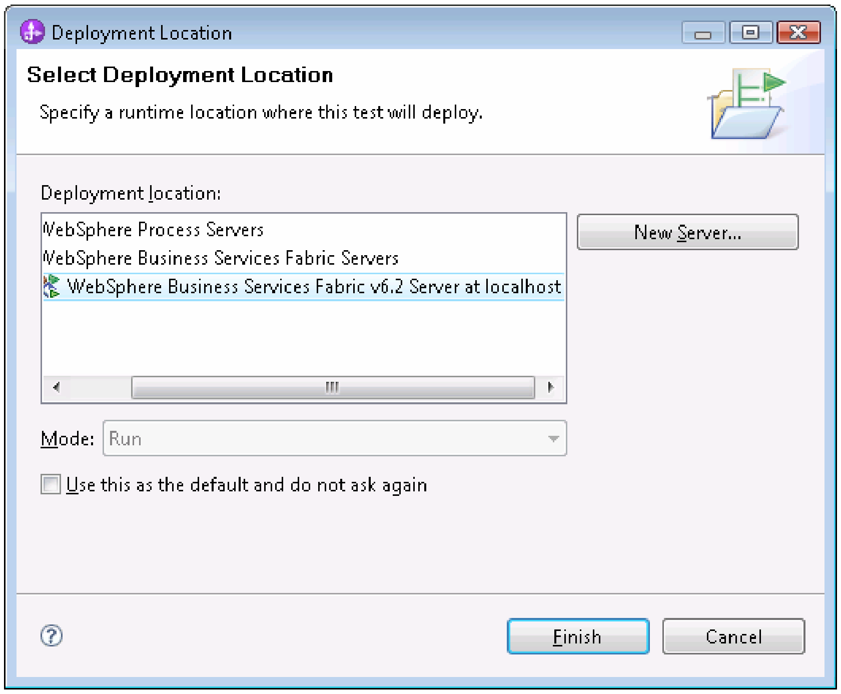
Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
loanApplication	Loan...	✓
customerInfo	Custo...	✓
creditScore	int	✓ 0
newCustomer	boole...	✓ false
customerName	string	✓ Reba Jones
loanInfo	LoanI...	✓
collateral	string	✓ Tin Soldiers: $500
lineOfBusiness	string	✓ Auto
loanStatus	string	✓ Open
loanFinalReviewDate	date	✓ 2009-05-02 -0500
amountOfLoan	double	✓ 24000
loanApplicationDate	date	✓ 2009-05-02 -0500

Events | Configurations

3. Choose the Fabric Server as the deployment location. Click Finish.


Deployment Location

Select Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

WebSphere Process Servers
WebSphere Business Services Fabric Servers
WebSphere Business Services Fabric v6.2 Server at localhost

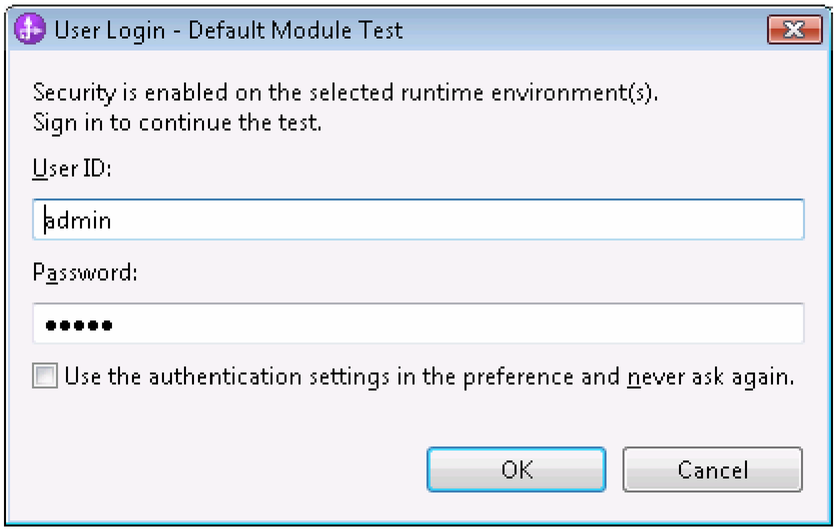
New Server...

Mode: Run

☐ Use this as the default and do not ask again

Finish | Cancel

4. If prompted with a user login, provide username/password of admin/admin and click OK.


User Login - Default Module Test

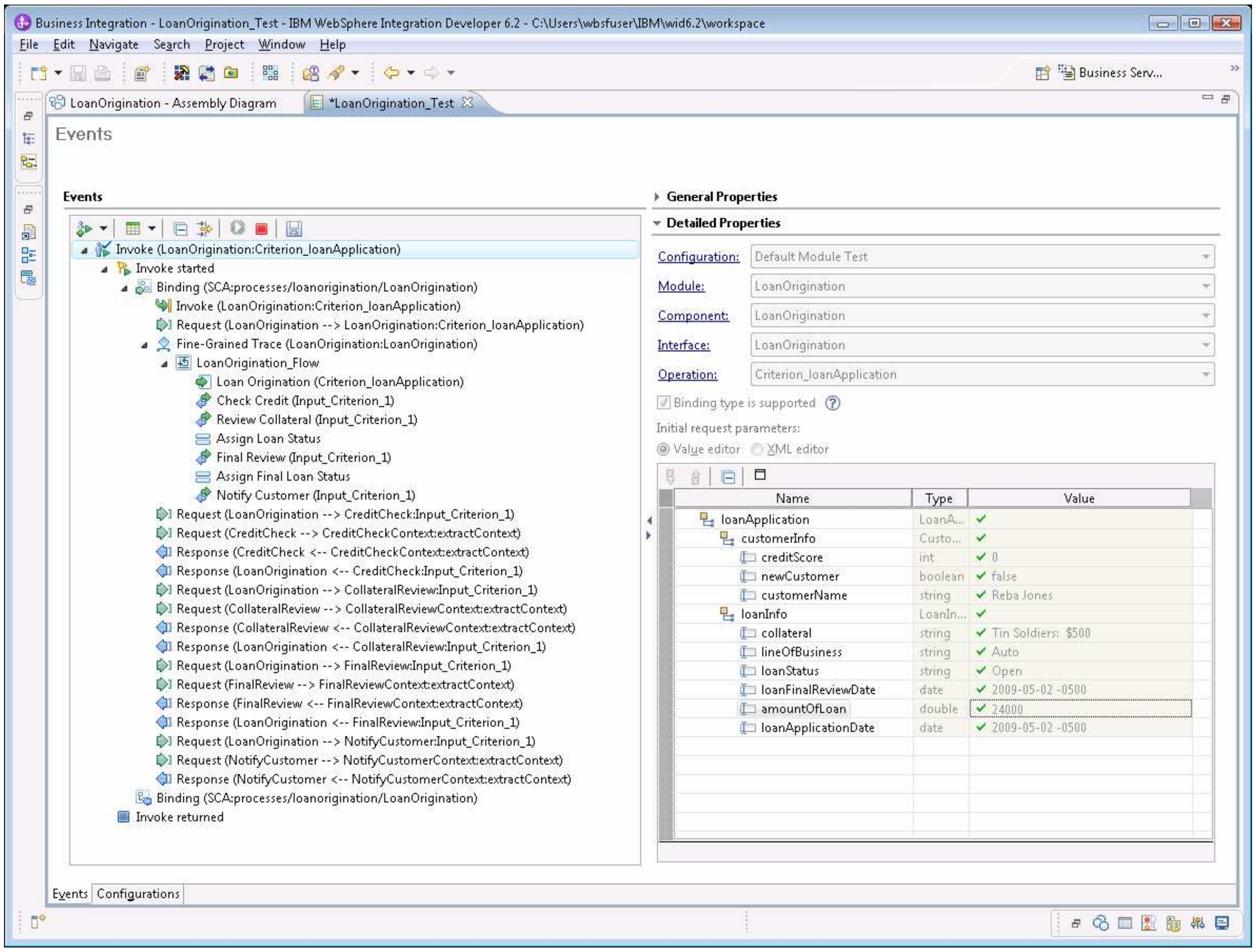
Security is enabled on the selected runtime environment(s).
Sign in to continue the test.

User ID: admin

Password: ●●●●●

☐ Use the authentication settings in the preference and never ask again.

A trace of the interaction unfolds in the events section. By clicking on request or responses it is possible to inspect the request and response messages sent or returned from various processing steps. The server console will display the message from the automated customer notification which indicates that Reba was approved for her loan.


Business Integration - LoanOrigination_Test - IBM WebSphere Integration Developer 6.2 - C:\Users\wbsfuser\IBM\wid6.2\workspace

File Edit Navigate Search Project Window Help
Business Serv...
LoanOrigination - Assembly Diagram
*LoanOrigination_Test

Events

Events

```
Invoke (LoanOrigination:Criterion_loanApplication)
  Invoke started
    Binding (SCA:processes/loanorigination/LoanOrigination)
      Invoke (LoanOrigination:Criterion_loanApplication)
      Request (LoanOrigination --> LoanOrigination:Criterion_loanApplication)
      Fine-Grained Trace (LoanOrigination:LoanOrigination)
        LoanOrigination_Flow
          Loan Origination (Criterion_loanApplication)
          Check Credit (Input_Criterion_1)
          Review Collateral (Input_Criterion_1)
          Assign Loan Status
          Final Review (Input_Criterion_1)
          Assign Final Loan Status
          Notify Customer (Input_Criterion_1)
      Request (LoanOrigination --> CreditCheck:Input_Criterion_1)
      Request (CreditCheck --> CreditCheckContext:extractContext)
      Response (CreditCheck <-- CreditCheckContext:extractContext)
      Response (LoanOrigination <-- CreditCheck:Input_Criterion_1)
      Request (LoanOrigination --> CollateralReview:Input_Criterion_1)
      Request (CollateralReview --> CollateralReviewContext:extractContext)
      Response (CollateralReview <-- CollateralReviewContext:extractContext)
      Response (LoanOrigination <-- CollateralReview:Input_Criterion_1)
      Request (LoanOrigination --> FinalReview:Input_Criterion_1)
      Request (FinalReview --> FinalReviewContext:extractContext)
      Response (FinalReview <-- FinalReviewContext:extractContext)
      Response (LoanOrigination <-- FinalReview:Input_Criterion_1)
      Request (LoanOrigination --> NotifyCustomer:Input_Criterion_1)
      Request (NotifyCustomer --> NotifyCustomerContext:extractContext)
      Response (NotifyCustomer <-- NotifyCustomerContext:extractContext)
    Binding (SCA:processes/loanorigination/LoanOrigination)
  Invoke returned
```

General Properties

Detailed Properties

Configuration: Default Module Test
Module: LoanOrigination
Component: LoanOrigination
Interface: LoanOrigination
Operation: Criterion_loanApplication

[✓] Binding type is supported

Initial request parameters:

[● Value editor] ○ XML editor

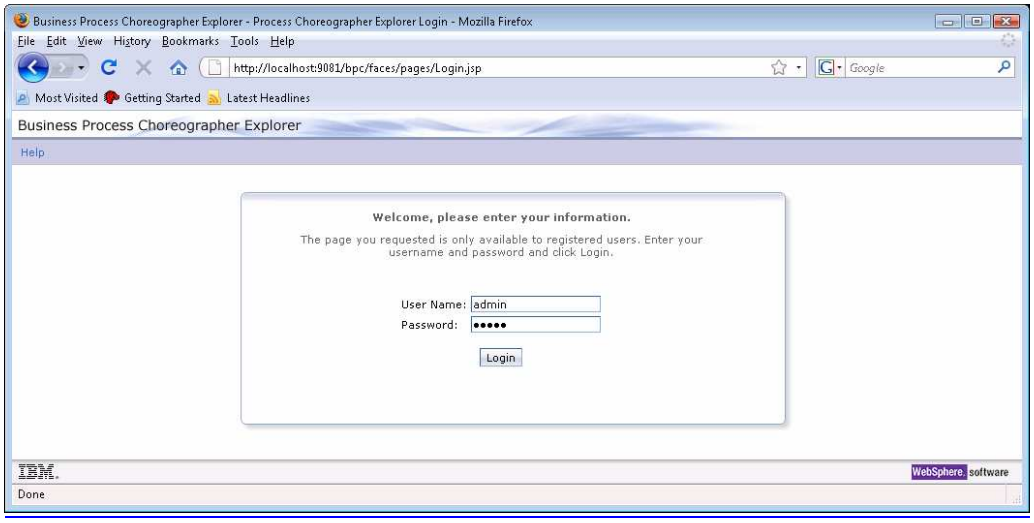
Name	Type	Value
loanApplication	LoanA...	✓
customerInfo	Custo...	✓
creditScore	int	✓ 0
newCustomer	boolean	✓ false
customerName	string	✓ Reba Jones
loanInfo	LoanIn...	✓
collateral	string	✓ Tin Soldiers: $500
lineOfBusiness	string	✓ Auto
loanStatus	string	✓ Open
loanFinalReviewDate	date	✓ 2009-05-02 -0500
amountOfLoan	double	✓ 24000
loanApplicationDate	date	✓ 2009-05-02 -0500

Next, try a round trip that uses the variations with human tasks.

5. Create and run new invocation with these settings in the loan application:

Field Name	Value	Comments
newCustomer	False	This will trigger the internal credit check to be used.
customerName	Reba Jones	
collateral	Vacation Home: $200,000	Textual description of collateral that can be reviewed.
lineOfBusiness	RV	RV loans require a human collateral review.
amountOfLoan	45000	Current policies say loan amounts in excess of 25K require a human review.

6. With a Web browser log into the business process choreographer at http://<server>:<port>/bpc.

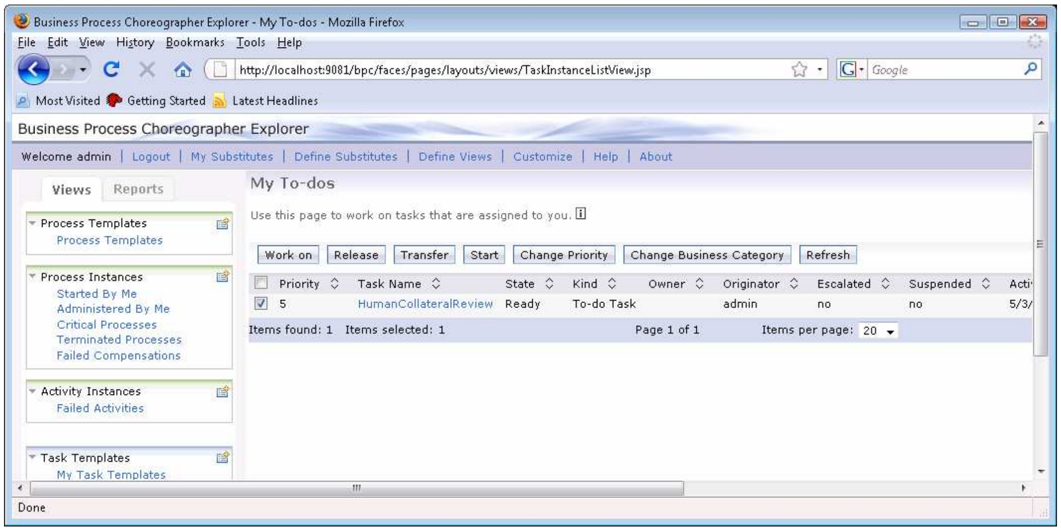

Business Process Choreographer Explorer - Process Choreographer Explorer Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9081/bpc/faces/pages/Login.jsp
Most Visited Getting Started Latest Headlines
Business Process Choreographer Explorer
Help
Login
Done

Welcome, please enter your information.

The page you requested is only available to registered users. Enter your username and password and click Login.

User Name: admin
Password: •••••

7. A human task for a collateral review is waiting. Select the task and click the Work on button.

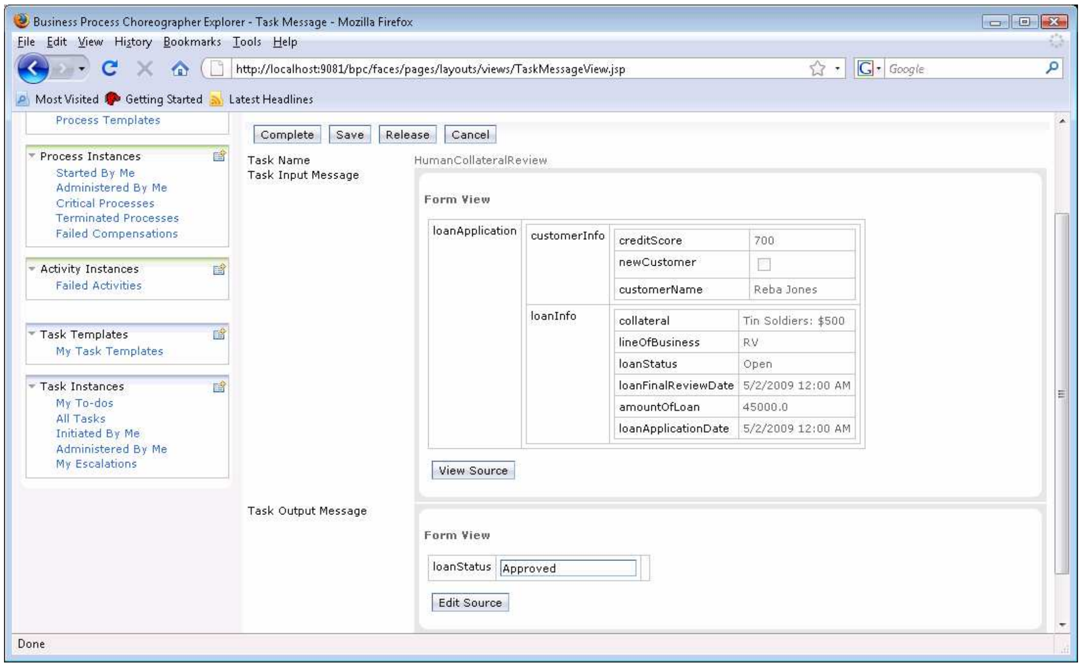

Business Process Choreographer Explorer - My To-dos - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9081/bpc/faces/pages/layouts/views/TaskInstanceListView.jsp
Most Visited Getting Started Latest Headlines
Business Process Choreographer Explorer
Welcome admin | Logout | My Substitutes | Define Substitutes | Define Views | Customize | Help | About
Views Reports
Process Templates
Process Templates
Process Instances
Started By Me
Administered By Me
Critical Processes
Terminated Processes
Failed Compensations
Activity Instances
Failed Activities
Task Templates
My Task Templates
My To-dos
Work on
Release
Transfer
Start
Change Priority
Change Business Category
Refresh
Done

Use this page to work on tasks that are assigned to you.

	Priority	Task Name	State	Kind	Owner	Originator	Escalated	Suspended	Acti
☑	5	HumanCollateralReview	Ready	To-do Task		admin	no	no	5/3/

Items found: 1 Items selected: 1 Page 1 of 1 Items per page: 20

8. Enter the value "APPROVED" for the loanStatus in the Task Output Message. Click complete.



Task Name HumanCollateralReview

Task Input Message

Form View

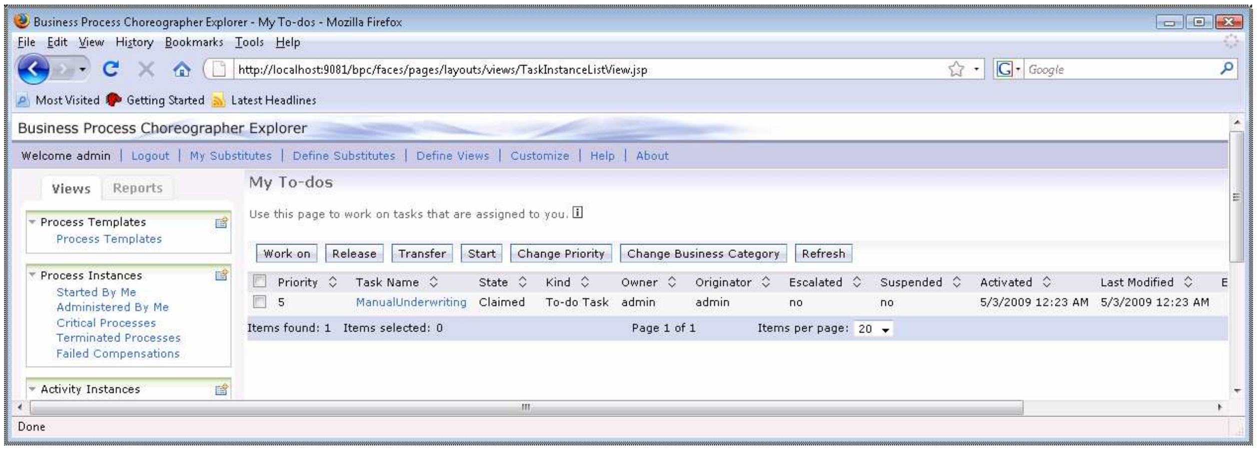
loanApplication	customerInfo	creditScore	700
		newCustomer	☐
		customerName	Reba Jones
	loanInfo	collateral	Tin Soldiers: $500
		lineOfBusiness	RV
		loanStatus	Open
		loanFinalReviewDate	5/2/2009 12:00 AM
		amountOfLoan	45000.0
		loanApplicationDate	5/2/2009 12:00 AM

Task Output Message

Form View

loanStatus	Approved

9. For the final review a new human task is created. If necessary, click Refresh for it to appear. Select this task and click the Work on button.


Business Process Choreographer Explorer - My To-dos - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9081/bpc/faces/pages/layouts/views/TaskInstanceListView.jsp
Most Visited Getting Started Latest Headlines
Business Process Choreographer Explorer
Welcome admin | Logout | My Substitutes | Define Substitutes | Define Views | Customize | Help | About
Views Reports
Process Templates
Process Templates
Process Instances
Started By Me
Administered By Me
Critical Processes
Terminated Processes
Failed Compensations
Activity Instances
Work on
Release
Transfer
Start
Change Priority
Change Business Category
Refresh
Done

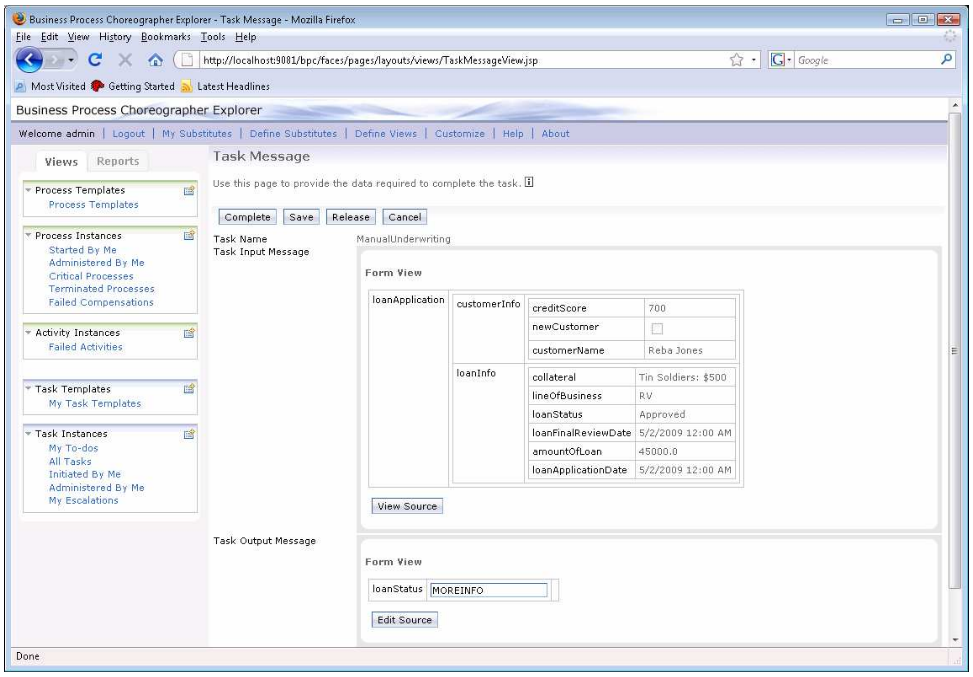
My To-dos

Use this page to work on tasks that are assigned to you.

	Priority	Task Name	State	Kind	Owner	Originator	Escalated	Suspended	Activated	Last Modified
☐	5	ManualUnderwriting	Claimed	To-do Task	admin	admin	no	no	5/3/2009 12:23 AM	5/3/2009 12:23 AM

Items found: 1 Items selected: 0 Page 1 of 1 Items per page: 20

10. Enter MOREINFO in the loanStatus and click complete.


Business Process Choreographer Explorer - Task Message - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9081/bpc/faces/pages/layouts/views/TaskMessageView.jsp
Most Visited Getting Started Latest Headlines
Business Process Choreographer Explorer
Welcome admin | Logout | My Substitutes | Define Substitutes | Define Views | Customize | Help | About
Views Reports
Process Templates
Process Templates
Process Instances
Started By Me
Administered By Me
Critical Processes
Terminated Processes
Failed Compensations
Activity Instances
Failed Activities
Task Templates
My Task Templates
Task Instances
My To-dos
All Tasks
Initiated By Me
Administered By Me
My Escalations
Complete Save Release Cancel
View Source
Edit Source
Done

Task Message

Use this page to provide the data required to complete the task.

Task Name ManualUnderwriting

Task Input Message

Form View

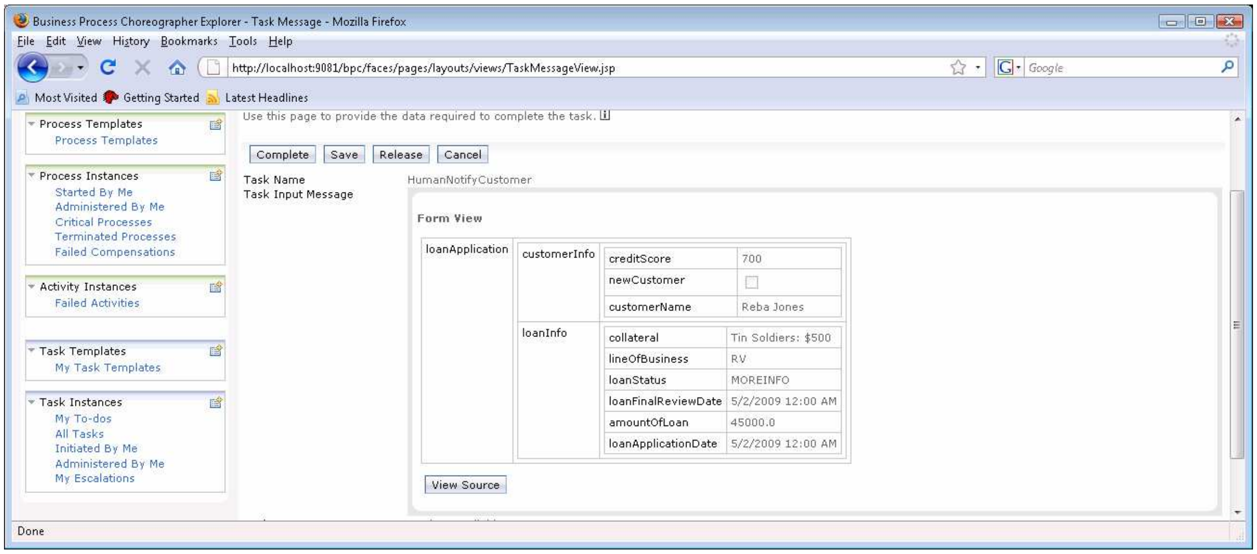
loanApplication	customerInfo	creditScore	700
		newCustomer	☐
		customerName	Reba Jones
	loanInfo	collateral	Tin Soldiers: $500
		lineOfBusiness	RV
		loanStatus	Approved
		loanFinalReviewDate	5/2/2009 12:00 AM
		amountOfLoan	45000.0
		loanApplicationDate	5/2/2009 12:00 AM

Task Output Message

Form View

loanStatus	MOREINFO

11. Finally, because more information is needed to process the loan, the human variant of Notify Customer is selected. Work on the task. Note that notify customer is modeled as a one-way operation. Clicking complete tells the system that the customer has been



Use this page to provide the data required to complete the task.

Task Name HumanNotifyCustomer

Task Input Message

Form View

loanApplication	customerInfo	creditScore	700
		newCustomer	☐
		customerName	Reba Jones
	loanInfo	collateral	Tin Soldiers: $500
		lineOfBusiness	RV
		loanStatus	MOREINFO
		loanFinalReviewDate	5/2/2009 12:00 AM
		amountOfLoan	45000.0
		loanApplicationDate	5/2/2009 12:00 AM