Creating flexible service-oriented business solutions with WebSphere Business Services Fabric, Part 4: Creating metadata for the business process

Skill Level: Introductory

Rohit P. Sardesai (rohitsardesai@in.ibm.com)
Software Engineer
IBM

Manojkumar P Pal (palmanojkumar@in.ibm.com)
Software Engineer
IBM

Suresh Das (dasuresh@in.ibm.com)
QA Lead
IBM

Pankuj Chachra (chpankuj@in.ibm.com)
QA Architect
IBM

12 Nov 2008

Learn how you can leverage the features of WebSphere Business Services Fabric to build composite business applications that support dynamic binding and orchestration. Part 4 of this series describes how to create metadata for a business process based on the Fabric Business Service Model. This metadata enables users to reuse and extend existing components, and enables dynamic endpoint selection at run-time.

Introduction

The WebSphere Business Services Fabric (hereafter called Fabric) Business

Service Model provides a rich domain for modeling composite business applications. You can use the Fabric Business Services Composition Studio (Composition Studio) and the administrative console to create instances of types defined in the Business Service Model and store them in the Business Services Repository as metadata. This metadata can be used by other developers to understand and reuse existing components or refine and extend them for their specific needs. The metadata can also be used by the Business Services Dynamic Assembler (Dynamic Assembler) to make routing decisions at run-time. For more information on the Business Service Model, refer to the WebSphere Business Services Fabric Version 6.1 Information Center.

In previous articles in this series, we designed and implemented an auto loan procurement business process. In this article, you'll learn how to create the metadata from the business process.

Prerequisites

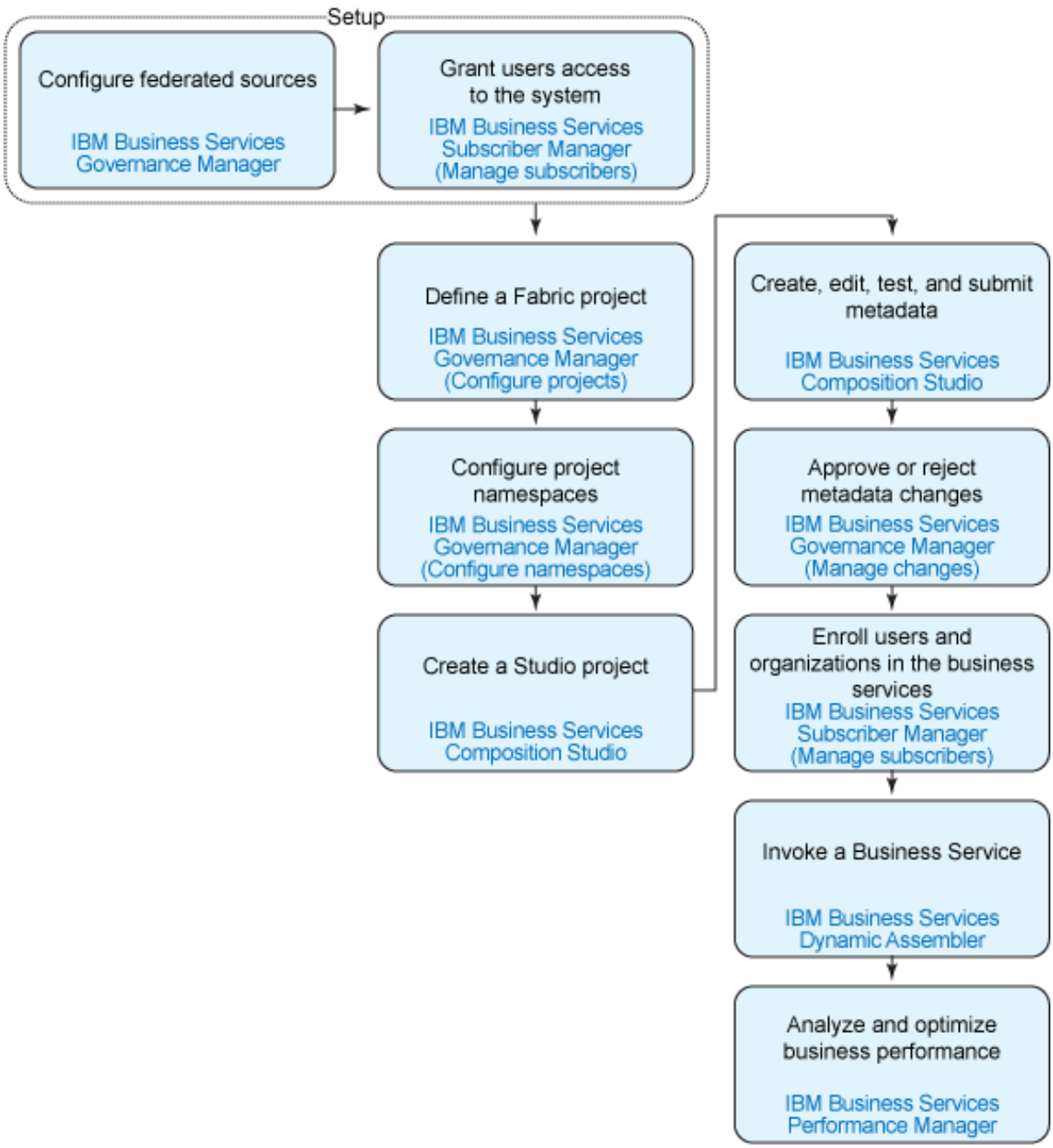
In order to complete the steps in this article, you need the following software installed on your system:

- IBM WebSphere Integration Developer V6.1
- IBM WebSphere Business Services ToolPack V6.1 (Unit Test Environment)

The Fabric assembly process

Fabric provides a governed environment that enables administrators, architects, business analysts, and developers to collaborate, assemble and deliver business services in an SOA system. Figure 1 shows a high-level view of the Fabric assembly process.

Figure 1. Fabric assembly process



We'll follow this process for our auto loan application, starting with the set-up stage. We'll skip the set-up of federated sources since we won't be using any. We'll start by setting up the organization and users of the system.

Create an organization and a user

Fabric supports multiple development teams working together to build business services using the Business Service Fabric model to create SOA applications. You

can use the Fabric model to organize business applications to make SOA governance easier. In this section, we'll create an organization and a user.

An organization can have employees with varying responsibilities. You can form various teams within an organization, and assign different roles within a team. You can then create users with the appropriate roles and assign them to different teams. This enables administrators to provide the appropriate privileges to users based on role. Complete the following steps to create an organization and a user.

To define an organization entity that will own the Fabric project where you'll store all the metadata for the application, complete the following steps:

1. Log in to the Fabric administrative console by pointing your browser to the following URL: `http://localhost:<port>/fabric`. For example, `http://localhost:9081/fabric/`, if the Unit Test Environment (UTE) is running on port 9081.
2. Select **My Services => Subscriber Manager => Manage Subscribers**.
3. Click **Create an Organization**.
4. Specify **Auto-org** as the **Organization name** and click **Create Organization**.
5. Click the **Users** tab, then click **Create a User**.
6. Specify **aut-admin** for the **User ID**, and fill in the remaining required fields. Add the **Administrator** role , and click **Create User**.

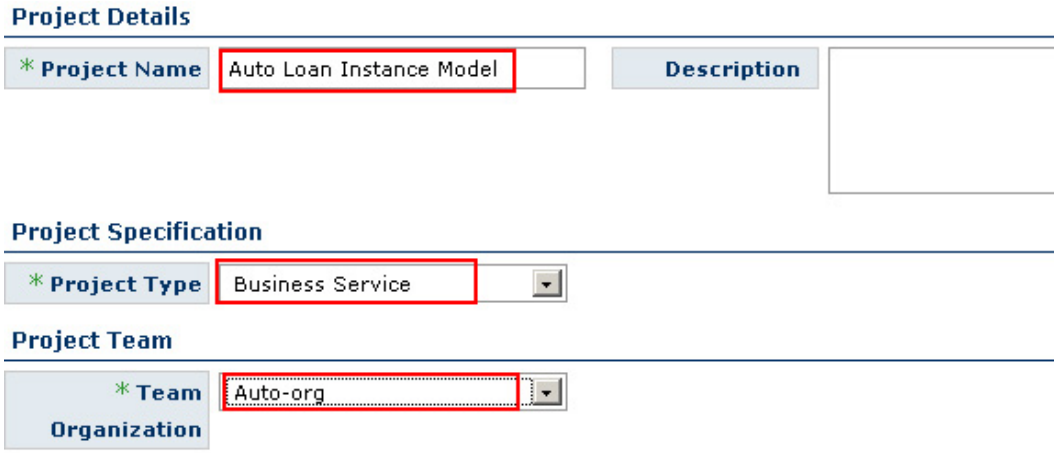
Create a Fabric project

Before you can define the metadata for the business process, you first need to define a Fabric project in the Fabric Governance Manager. A Fabric project is used to group all the metadata related to your business service. This metadata can be exported and shared with other developers. To create a Fabric project, do the following:

1. Select **Governance Manager => Configure Projects**.
2. Click **Create a Project**, and provide the following information, as shown in Figure 2:
   - **Name**: `Auto Loan Instance Model`
   - **Project Type**: **Business Service**

- **Team Organization**: `Auto-org`

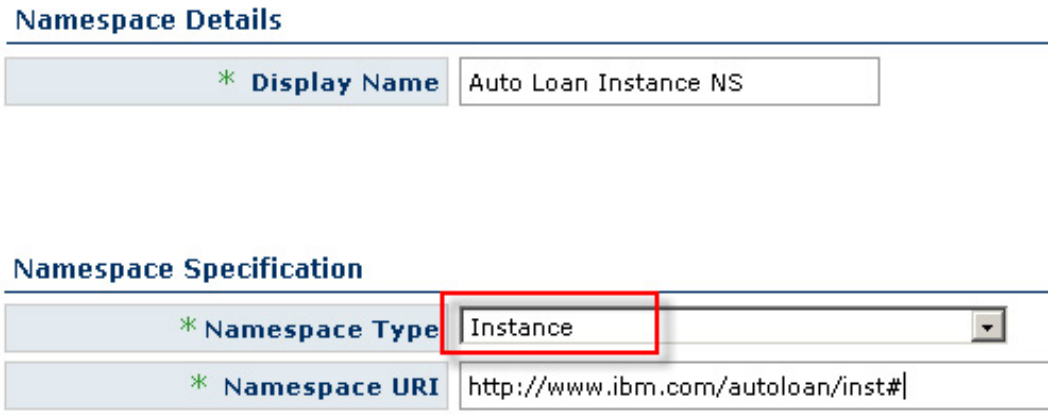
3. Click **Create Project**.
Figure 2. Create a project



4. Now you need to create a namespace for holding service instance metadata and associate it with the project created above. Click the **Namespaces** tab and click **Create a Namespace**, then provide the following information as shown in Figure 3:

- **Display Name>**: `Auto Loan Instance NS`
- **Namespace Type**: **Instance**
- **Namespace URI**: `http://www.ibm.com/autoloan/inst#`

5. Click **Create Namespace**.
Figure 3. Create a namespace instance for the project

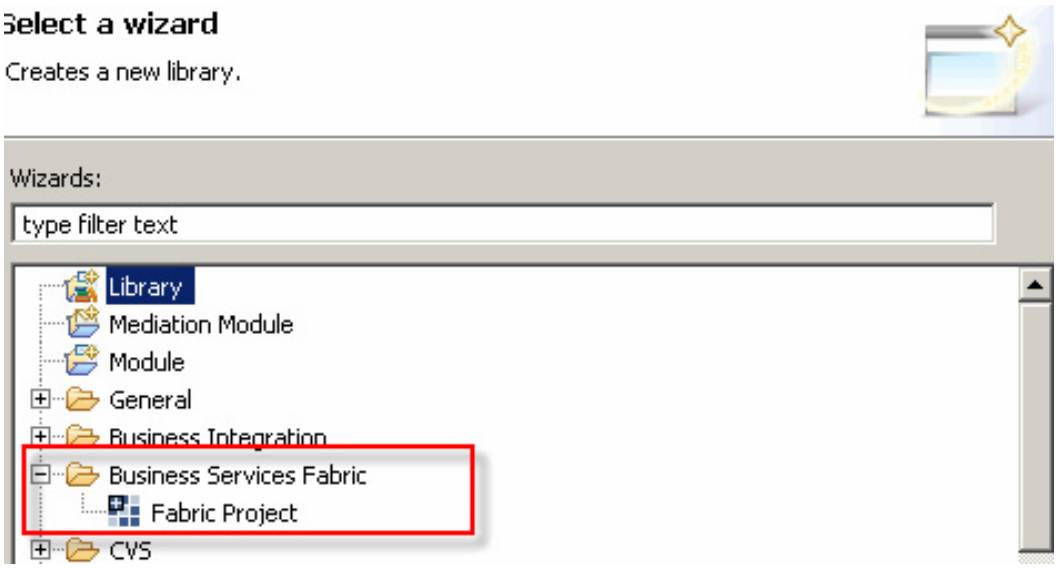


Create metadata for the composite service and interfaces

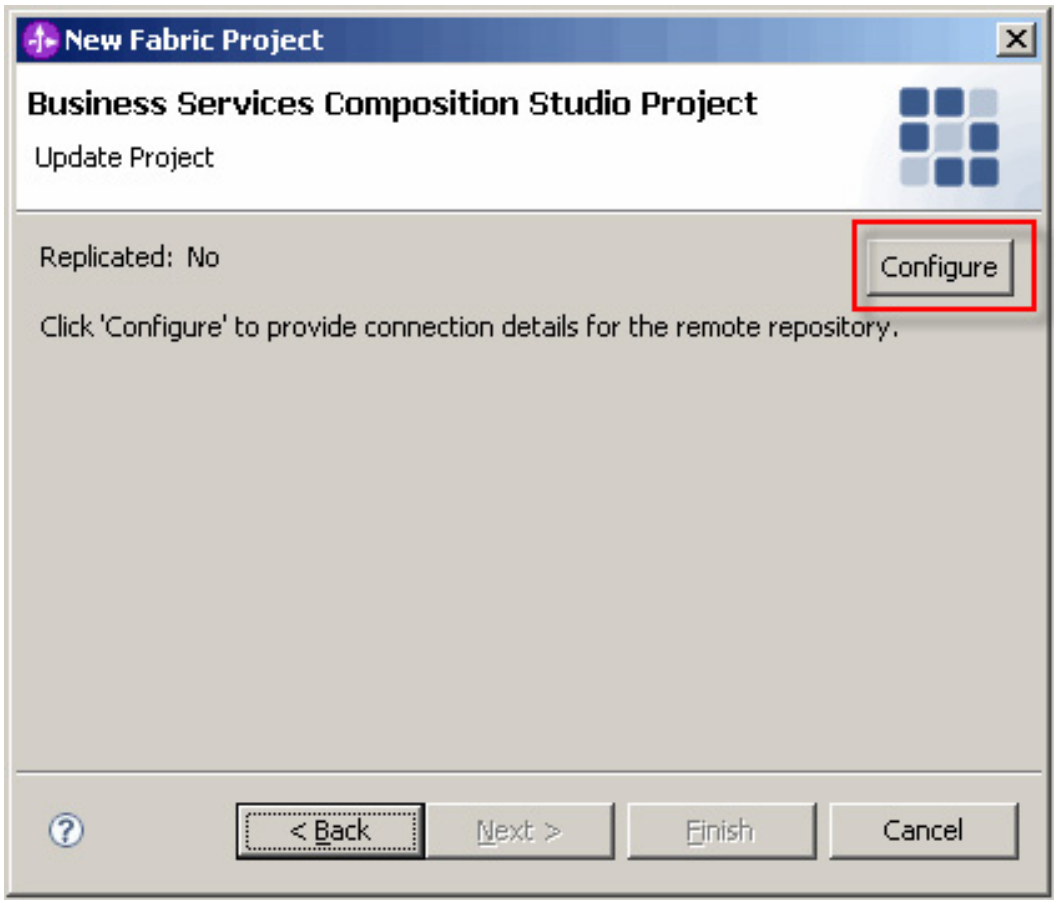
Now that you've created users, projects and namespaces, the next step, as shown in the assembly process in Figure 1, is to create a project. You can use the Composition Studio, an Eclipse-based plug-in, to create metadata for your applications. The metadata is then stored in the Business Services Repository and used by the Dynamic Assembler for dynamic endpoint selection.

To replicate the project you just created into the Composition Studio and create service metadata for the composite service interfaces and endpoints in this project, complete the following steps:

1. In WebSphere Integration Developer, switch to the **Business Service** perspective.
2. Select **Window => Open Perspective => Other**, and select **Business Service**.
3. Select **File => New => Project => Business Services Fabric => Fabric Project**, as shown in Figure 4.
   Figure 4. Configure a Fabric project

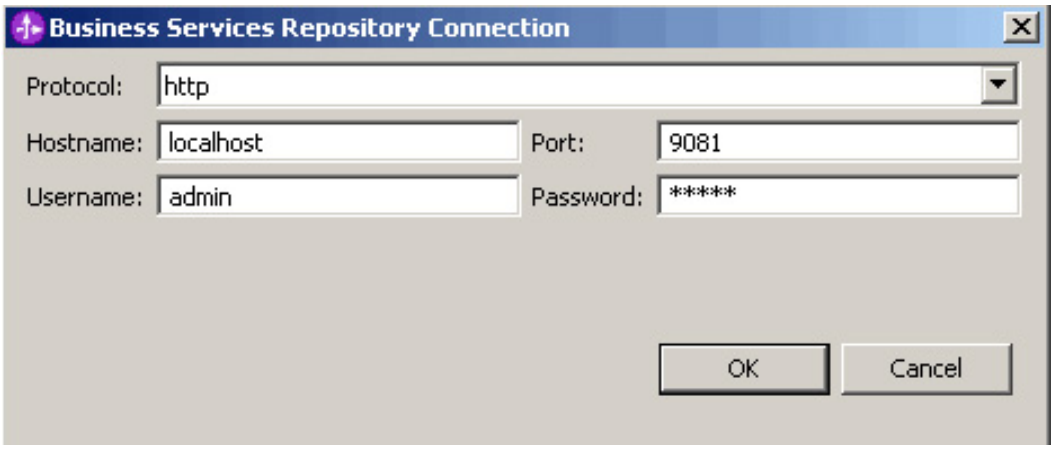


4. Click **Next**.
5. Enter the project name and click **Next**.
6. Click **Configure, as shown in Figure 5.**
   Figure 5. Configure a Fabric project



7. Specify the repository connection information, as shown in Figure 6, by specifying the following information:
   - **Hostname:** `localhost`. The host where Fabric is deployed.
   - **Port:** `portnumber`. The port on which the Fabric UTE server is running (for example, 9081).
   - **Username:** admin
   - **Password:** admin
8. Click **OK**.

Figure 6. Specify repository connection information



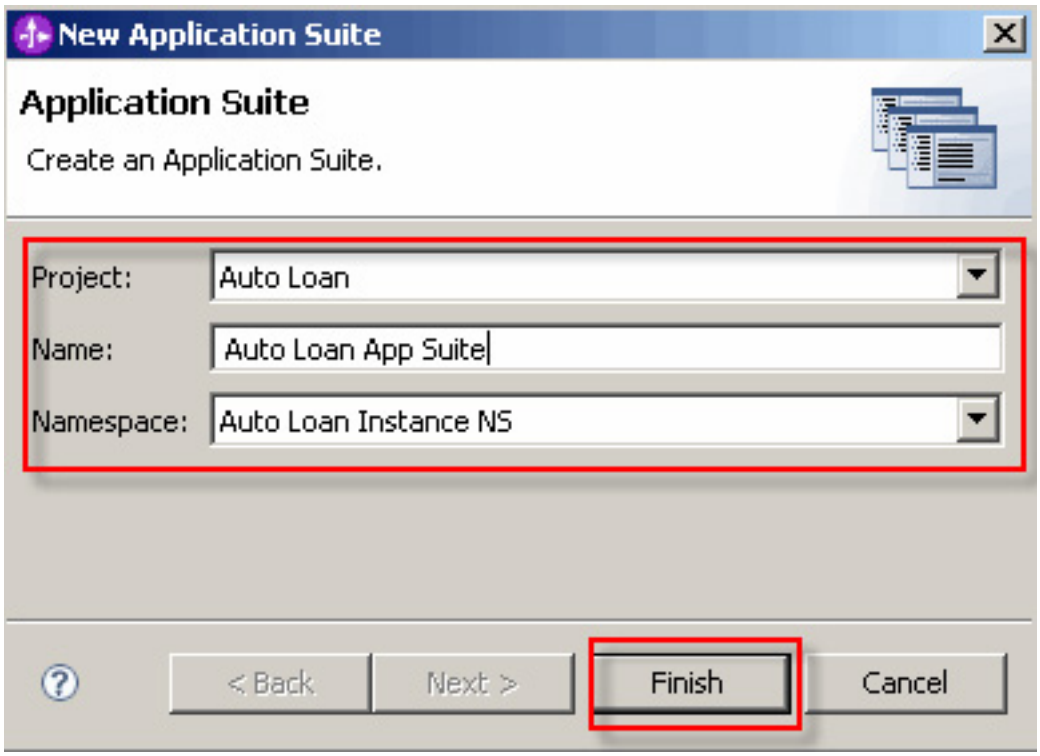
9. Click **Next**. Select the **Auto Loan Procurement** project and click **Finish**.

Now you have a Fabric project that will hold the composite service and associated interfaces for the auto loan application.

Create an application suite and application

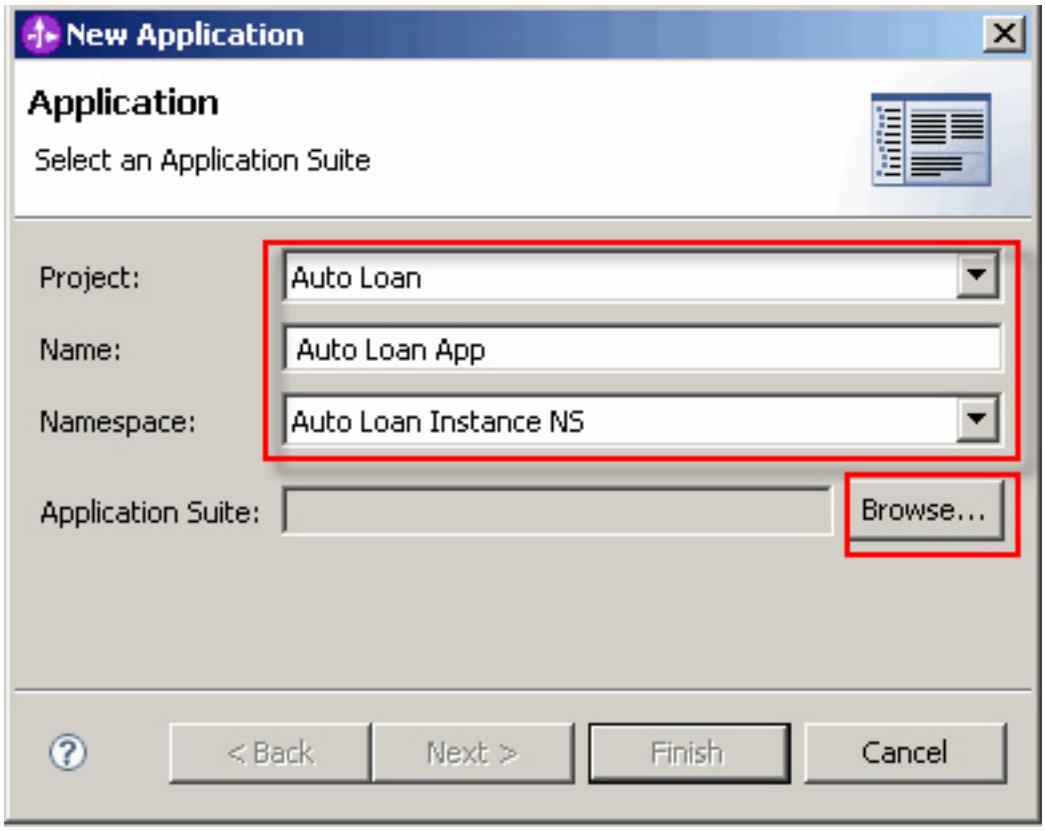
You've learned about the concepts of application suites, applications, and business services. An application suite can contain multiple applications. An application can consist of multiple business services. A business service can consist of multiple Web services and endpoints. You can use the Composition Studio to create instances of each of these types that are defined in the Fabric model and store them in the Business Services Repository.

1. Right-click the Fabric project and select **New => Application Suite**.
2. Specify the information in the New Application Suite dialog, as shown in Figure 7.
   Figure 7. Creating a new application suite



3. Right-click the Fabric Project and select **New => Application**.

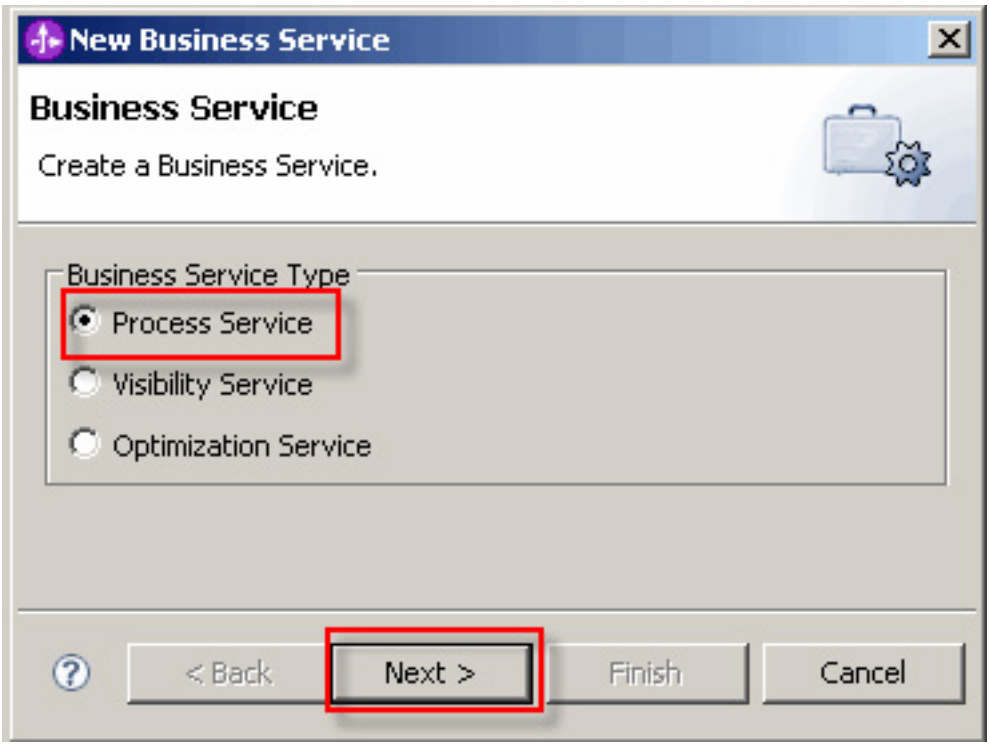
4. Specify the application information as shown in Figure 8. Click **Browse** and select the new application suite you just created (**Auto Loan App Suite**), then click **Finish**.
Figure 8. Create an application



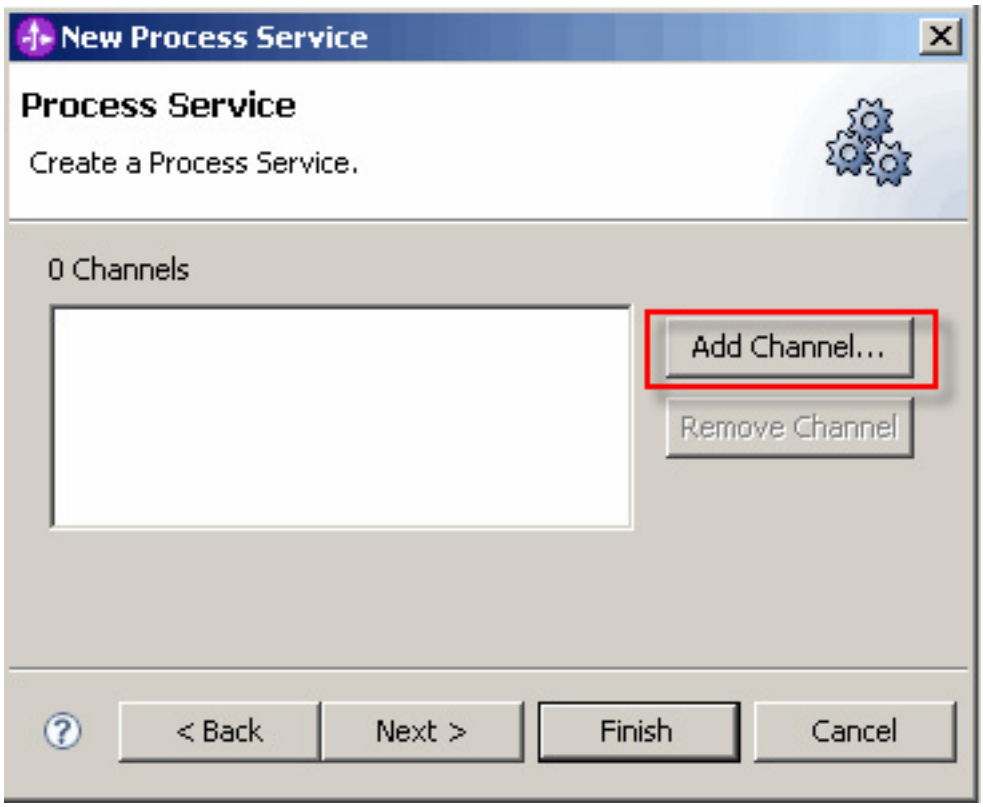
Create a process service

Complete the following steps to create a process service:

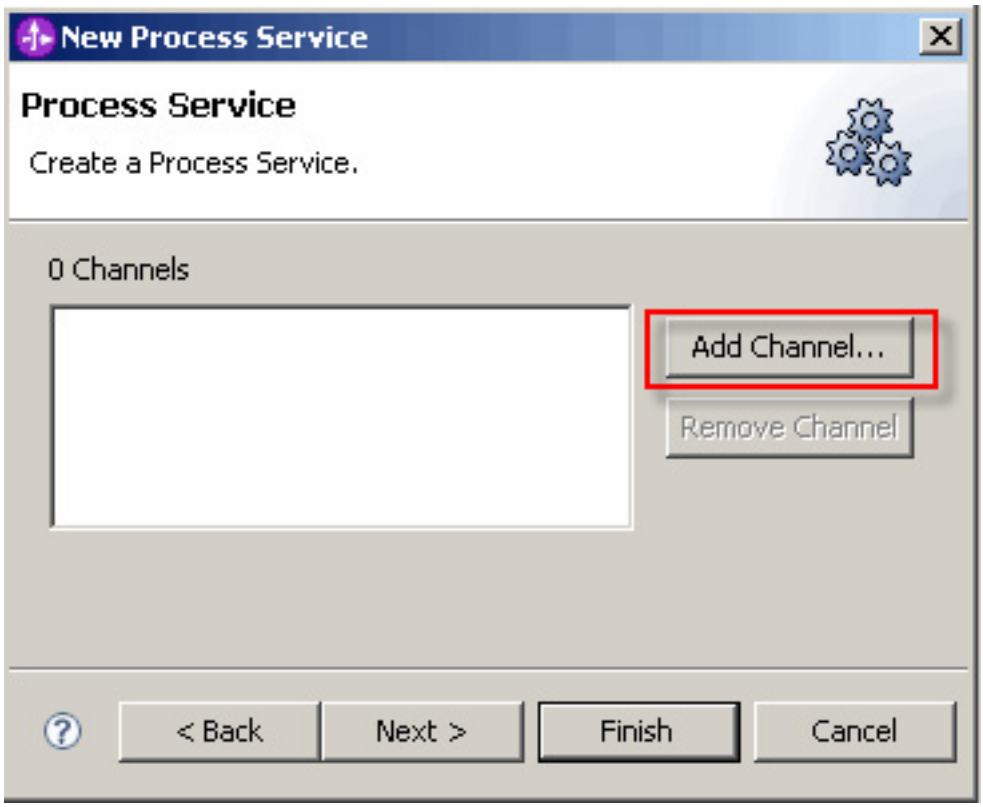
1. Right-click the project and select **New =>Business Service**, then select **Process Service**, as shown in Figure 9.
   Figure 9. Create a process service



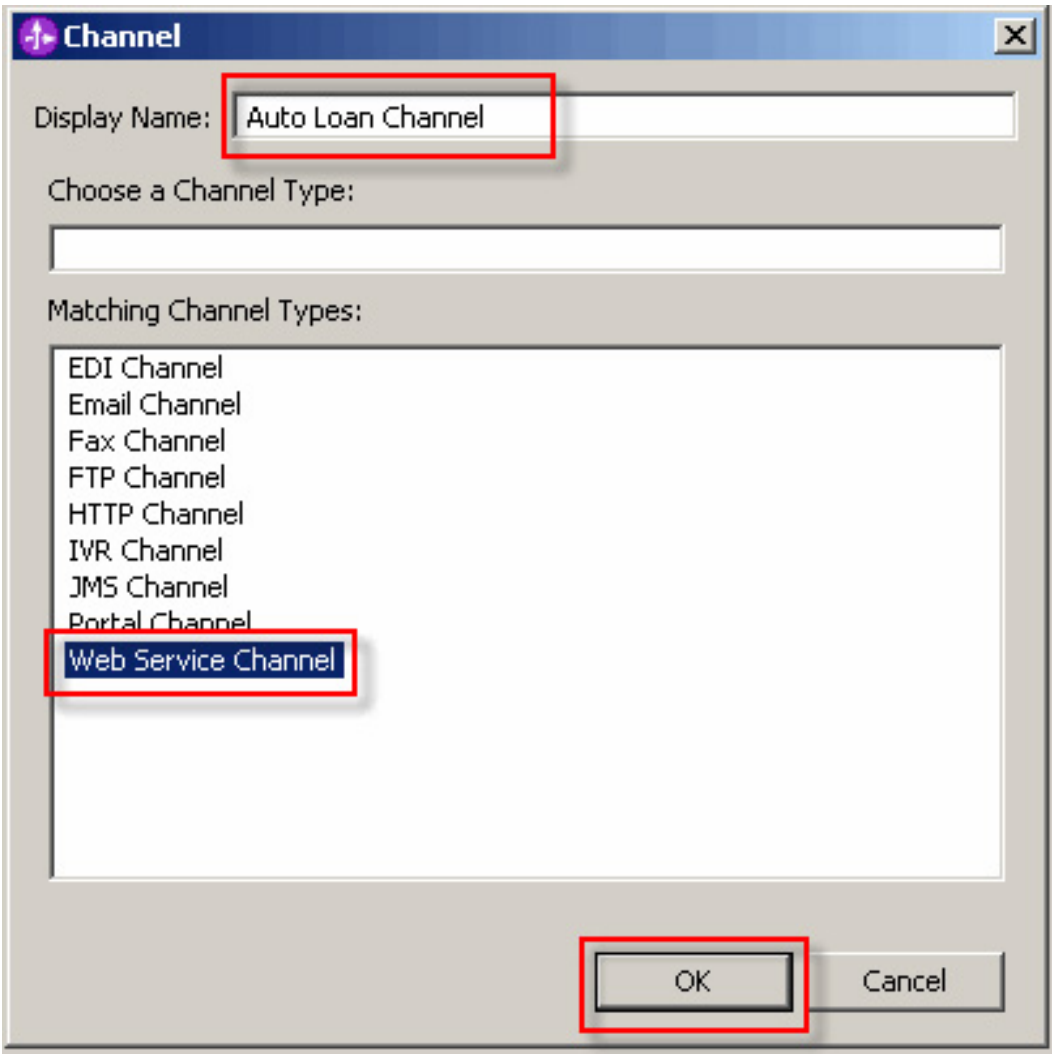
2. Specify the information for the process service as shown in Figure 10, then click **Browse** and select the application you created in the previous step (**Auto Loan App**), then click **Next**.
Figure 9. Specify process service information



3. Add a channel for the process service by clicking **Add Channel**, as shown in Figure 11.
Figure 11. Add a channel

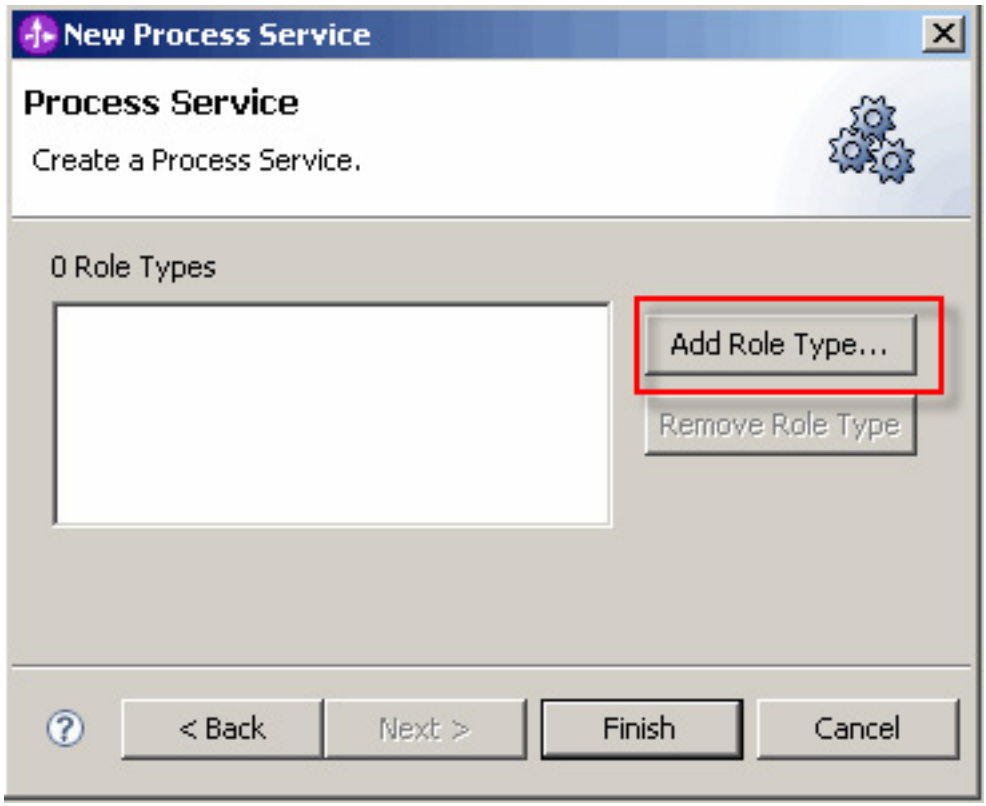


4. In the Channel dialog, specify a display name for the channel, select **Web Service Channel**, and click **OK**, as shown in Figure 12.
Figure 12. Specify channel information



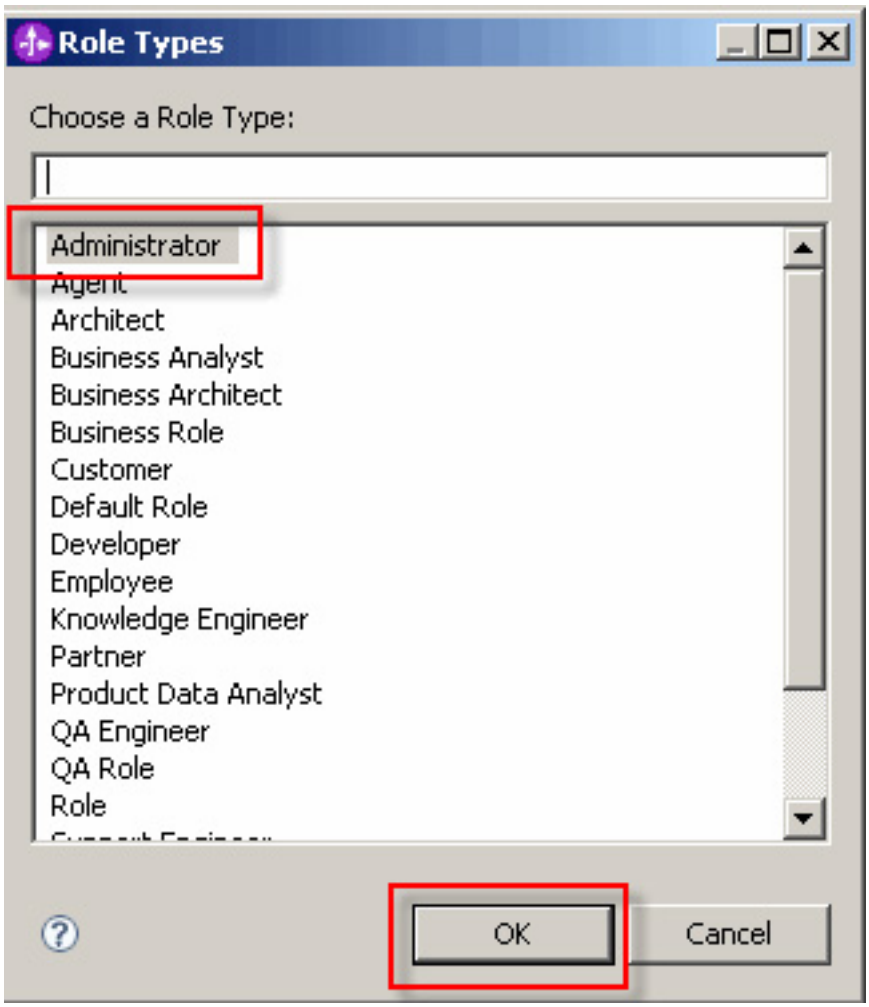
5. Click **Next**.

6. Click **Add Role Type**, as shown in Figure 13.
Figure 13. Add a role type



7. Select **Administrator** for the role type, and click **OK**, as shown in Figure 14.

Figure 14. Select Administrator role type



8. Click **Finish**.

Create composite services, interfaces and endpoints

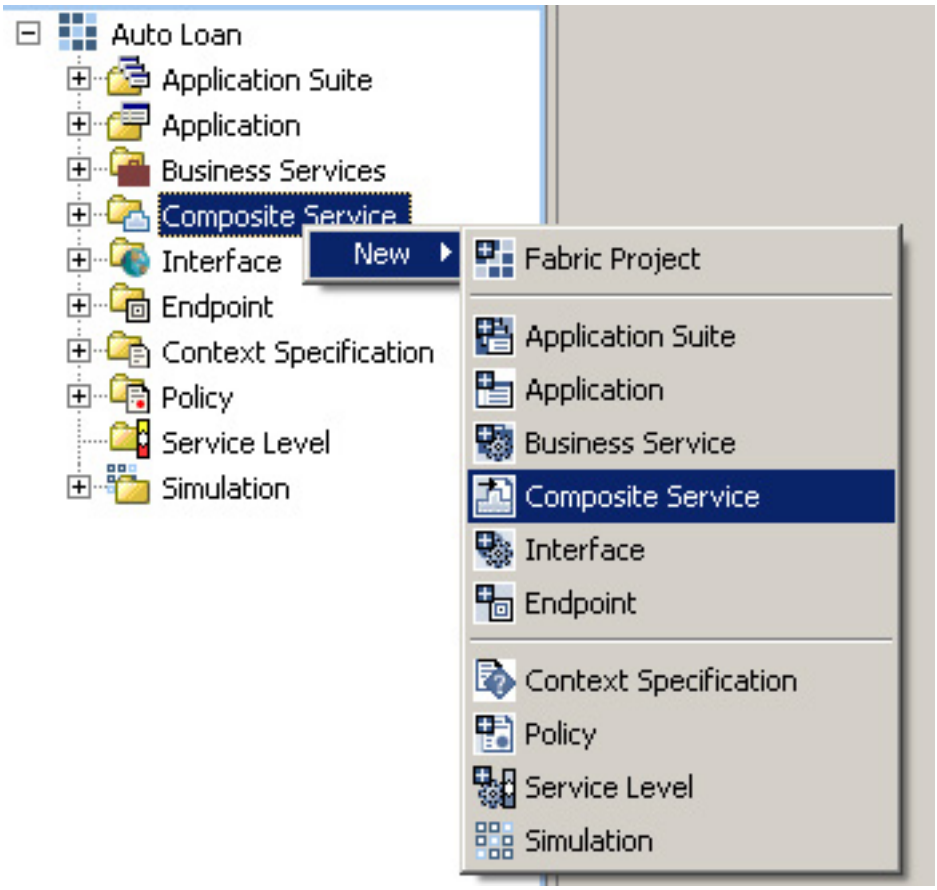
So far, you've created an auto loan application suite that holds an auto loan application, which in turn consists of an auto loan process service. Now you need to create a composite service definition for the actual BPEL process contained in the business service. Also, you need to create interface definitions for the atomic services called by the BPEL process.

In order to create this metadata, you'll need the BPEL module you created in the Part 3 of this series. If you don't already have this module, you can import the project interchange zip from the Download section into your Integration Developer workspace.

To create the composite service, do the following:

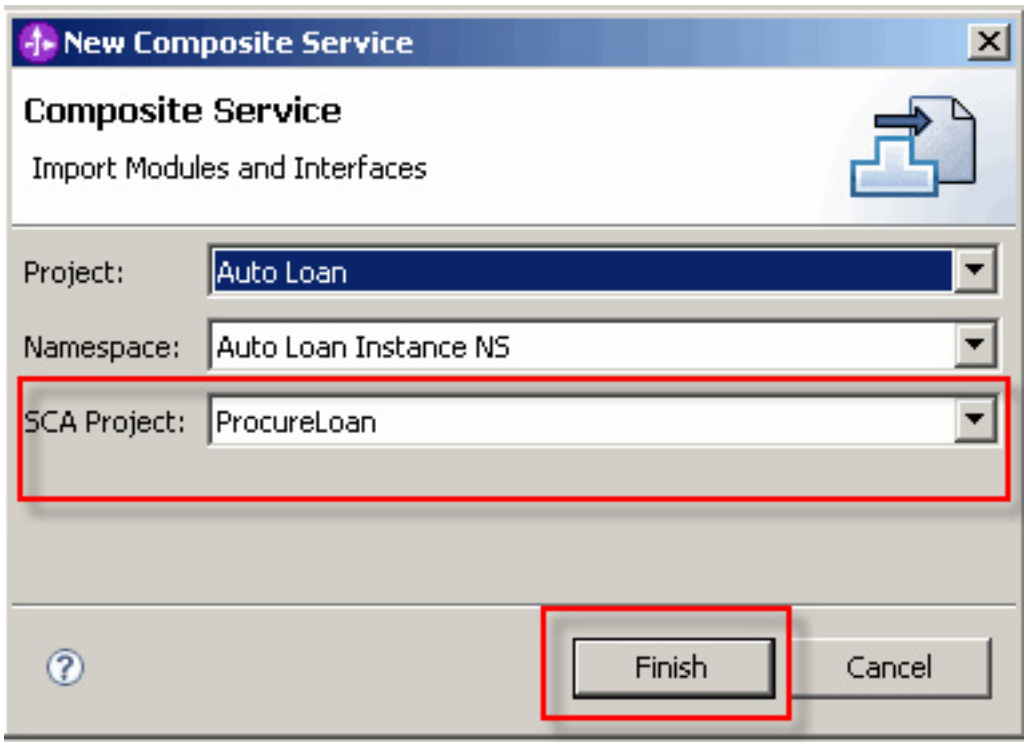
1. Right-click on the Fabric project and select **New => Composite Service**, as shown in Figure 15.

Figure 15. Create a composite service



2. The input to this wizard is the ProcureLoan module that you previously imported into the workspace. Specify **ProcureLoan** as the **SCA Project** and click **Finish**, as shown in Figure 16. The wizard creates metadata definitions for the composite service and the atomic services invoked by the composite service.

Figure 16. Specify SCA project

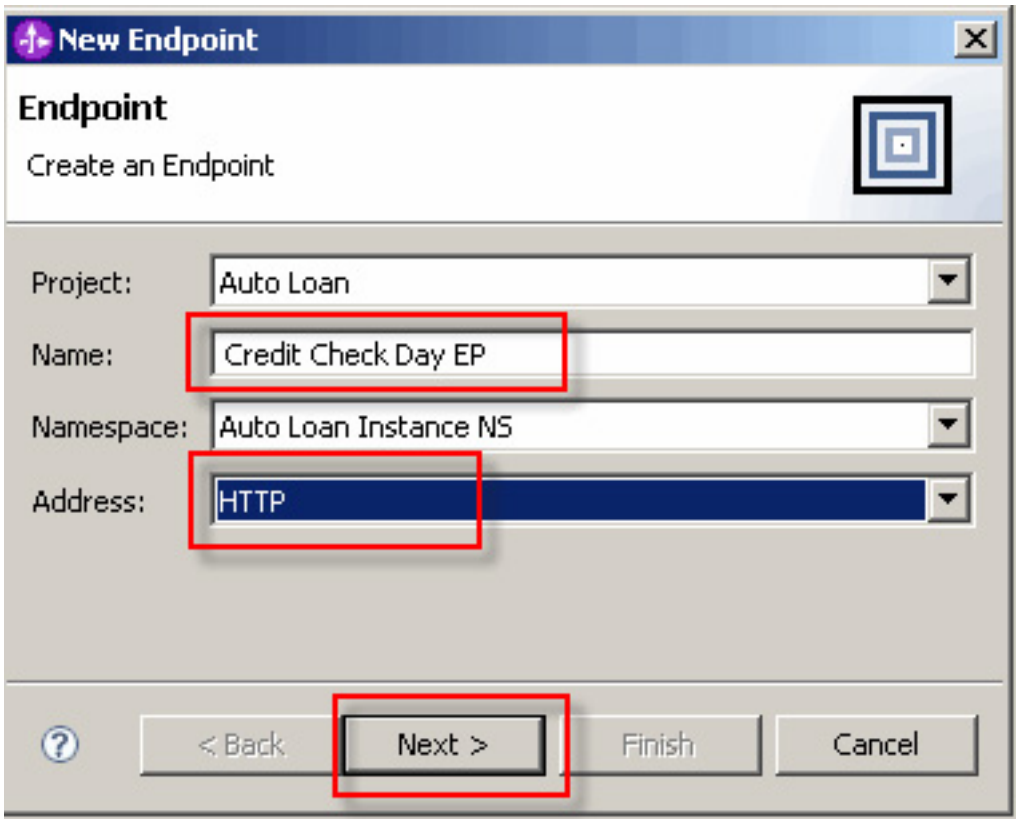


Now that you've created the service interface definitions, you next need to create service endpoint definitions which will perform the actual task of servicing the requests.

To create endpoints for the Credit Check, License Check and Loan Provider services, do the following:

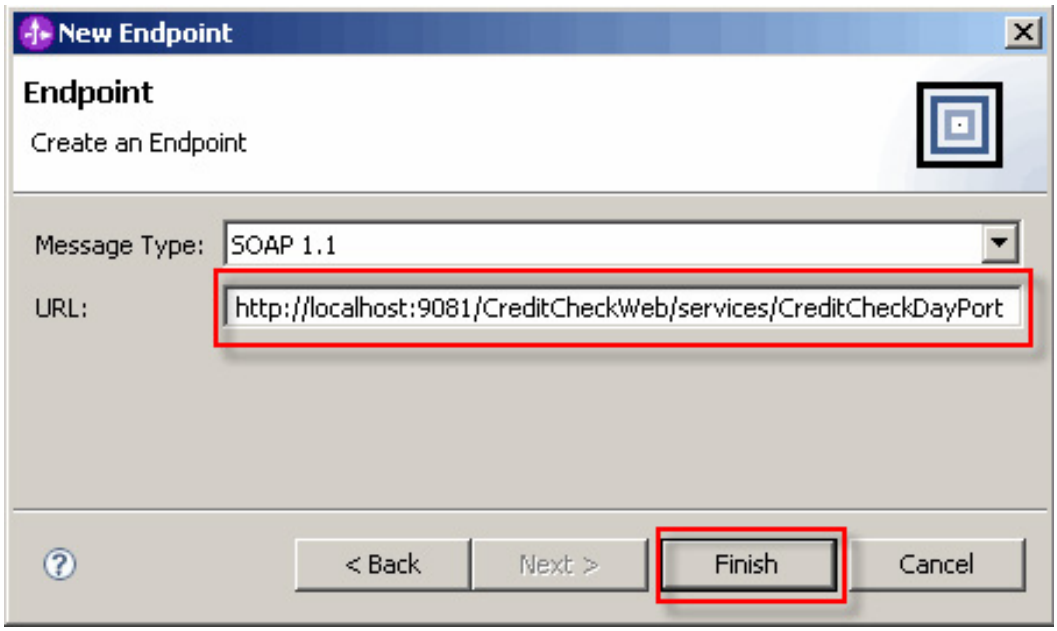
1. Right-click the project and select **New => Endpoint**.

2. Enter the endpoint information for the Credit Check as shown in Figure 16, then click **Next**,
   Figure 17. Create a Credit Check Service endpoint for daytime processing



3. Specify the URL where the endpoint for the Credit Rating Day Service implementation is deployed, as shown in Figure 18, and click **Finish**.

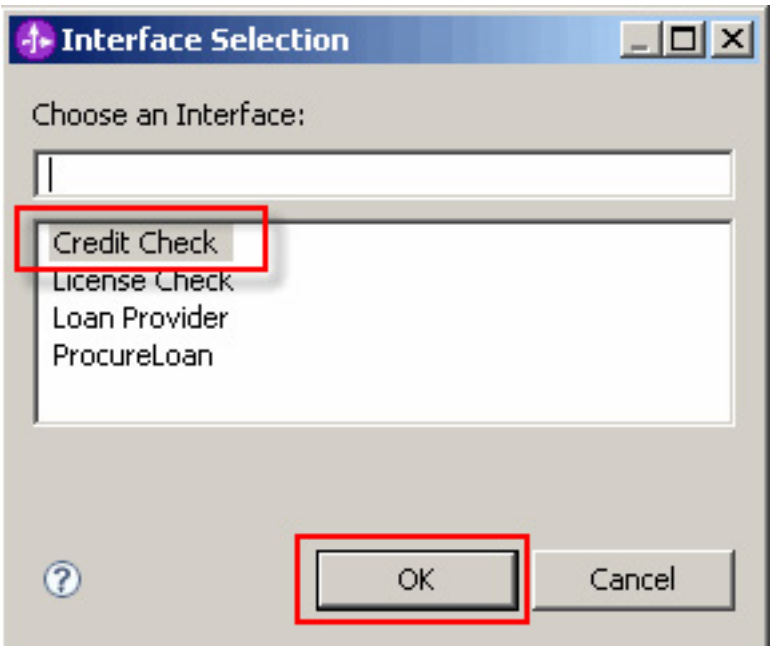
Figure 18. Specify endpoint URL



4. Click the **Interfaces** tab, and add the CreditCheck interface to the

endpoint, as shown in Figure 18, then click **OK**.

Figure 18. Add CreditCheck interface



5. Repeat steps 1-4 for the remaining endpoints in Table 1.

Table 1. Endpoints you need to create

Service interface	Endpoint name	Endpoint URL
CreditCheck	Credit Check Day EP	http://localhost:9081/CreditCheckWeb/services/CreditCheckDayPort
CreditCheck	Credit Check Night EP	http://localhost:9081/CreditCheckWeb/services/CreditCheckNightPort
LicenseCheck	License Check CA EP	http://localhost:9081/LicenseCheckWeb/services/LicenseCheckCAPort
LicenseCheck	License Check NonCA EP	http://localhost:9081/LicenseCheckWeb/services/LicenseCheckNonCA
LoanProvider	Loan Provider Gold EP	http://localhost:9081/LoanProviderWeb/services/LoanProviderGLDPort
LoanProvider	Loan Provider Silver EP	http://localhost:9081/LoanProviderWeb/services/LoanProviderSLVPort

6. Double-click **Auto Loan Process Service**.

7. Click the **Channels** tab.

8. In the **Interfaces** section, click **Add**. Select **ProcureLoan** from the **Module** drop-down menu, then click on **OK**

9. Open the **Repository Changes** view.

10. Right-click the project and select **Submit Changelist**. Since you're using the UTE environment , the changes are automatically approved and published to the repository.

Export the project to the repository

The Fabric Governance Manager provides the ability to export all the metadata that you've created in the form of a Fabric Content Archive (FCA). Importing the FCA into a blank repository automatically creates metadata, thus eliminating the need to manually create the metadata.

An FCA file is a zip file that contains Web Ontology Language (OWL) files and a special **manifest** file called content-pack.xml. The manifest file contains all the information necessary to define a Fabric project and its namespaces. For each namespace owned by the project, it can refer to the OWL file containing content for this namespace.

To export the metadata as an FCA, complete the following steps:

1. Select **Import/Export => Export by Project** .

2. Select the **Auto Loan Procurement** project and the namespace instance.

3. Click **Export to File**. This creates an FCA for your project.

Summary

In this article, you learned about the Fabric assembly process and walked through some of the steps involved in the process, including creating users, projects and namespaces in the Fabric administrative console, replicating the project in Composition Studio, and creating metadata for the application. Finally, you learned how to export these changes to the Business Services Repository.

In Part 5, we'll walk through the remaining steps in the Fabric assembly process (refer to Figure 1. You'll use the metadata created in this article and add capabilities in the form of assertions to the endpoints. You'll also create policies to address the points of variability in the business process. Finally, you'll see how the Dynamic Assembler uses all the metadata you've created and published to the repository to dynamically invoke endpoints based on customer requests.

Downloads

Description	Name	Size	Download method
Auto loan project interchange	AutoLoanPI.zip	31KB	HTTP

Information about download methods

Resources

- Creating flexible service-oriented business solutions with WebSphere Business Services Fabric, Part 1: Overview: Part 1 of this series describes the concepts of business services and composite business services and how Business Services Fabric provides an SOA platform to model, assemble, deploy, manage and govern composite business services.
- Creating flexible service-oriented business solutions with WebSphere Business Services Fabric, Part 2: Extending the ontology models: Part 2 of this series describes the use case of a sample application and extending the ontology models for capturing the points of variability in the business process.
- Creating flexible service-oriented business solutions with WebSphere Business Services Fabric, Part 2: Designing and implementing the business service: Part 3 walks you through the steps to implement a composite business process and enable it with dynamic binding and orchestration capabilities.
- WebSphere business process management zone: Get the latest technical resources for WebSphere BPM solutions, including articles, tutorials, events, downloads, and more.
- Business Process Management enabled by SOA: Get complete product information on IBM BPM software, including features and benefits, downloads, and more.
- WebSphere Business Services Fabric product information: Get product information, including features and prerequisites.
- Service Component Architecture (SCA): Get information and the specification for SCA, a technology that simplifies application development and implementation in an SOA.
- WebSphere Business Process Management Version 6.1 information center: Get complete product documentation for WebSphere BPM products.
- Getting Started with IBM WebSphere Business Services Fabric V6.1: This IBM Redbook provides a complete overview of Fabric, from an architectural introduction, to an installation guide, and a step-by-step scenario that describes how to model, assemble, deploy, and manage composite business applications.

About the authors

Rohit P. Sardesai
Rohit Sardesai works as a software engineer in the IBM India Software Lab, in Mumbai. He is part of the IBM WebSphere Business Services Fabric team.

Manojkumar P Pal
Manojkumar Pal works a Software Engineer in the India Software Lab, Mumbai. He is involved in the design and development of Websphere Business Services Fabric.

Suresh Das
Suresh Das is a Test Lead and works on the Websphere Business Services Fabric in Composition Studio. He is responsible for test planning, execution and the entire Automation for Studio.He is also responsible for GVT/TVT cordination

---

Pankuj Chachra
Pankuj Chachra is a QA architect and works on Websphere Business Services Fabric .