NaturalONE

Business Services

Version 8.2.7

March 2013

This document applies to NaturalONE Version 8.2.7.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ONE-SERVICES-DOC-827-20130320

Table of Contents

Preface

This documentation describes how to create and maintain business services with the optional Business Service component of NaturalONE. It is organized under the following headings:

Release Notes	Information on new features and enhancements.
What is the Business Service Component	Brief description of this NaturalONE component.
Before You Start	Information on how to set up a Natural Business Services environment and a testing environment.
Getting Started	Step-by-step instructions on how to create a business service, steplib and domain, and how to download business services from the server.
Service Development	Detailed descriptions of all service development tasks.

I Release Notes

These *Release Notes* pertain to the Business Services component of NaturalONE version 8.3. The following topics are covered:

1 What's New in Version 8.2.1

This section describes the changes to the Business Services component in version 8.2.1. The following topics are covered:

Enhancements

This section describes the changes in this release of the Business Services component. The following topics are covered:

- Java Client Generation
- Natural Server View
- Security View

Java Client Generation

The style of Java clients generated for business services has changed. Previous clients will continue to function correctly, but new clients will be generated using a different style that uses EntireX RPC directly instead of requiring the generation of an EntireX RPC wrapper. All Java client generation details are now provided in this guide; they are no longer documented in the Java Wrapper/Web Service Wrapper sections of the EntireX documentation. For information, see ***Generate a Java Client***.

Natural Server View

The Add from Server wizard has been replaced with standard Natural Server view functionality. You can now add business services to a local project directly from the Natural Server view. For information, see ***Downloading Business Services from the Server***.

Security View

The **NBS Security** view is now available and you can apply security at a domain, business service, and/or method level. The **NBS Security** view is linked to the **Navigator** and **Natural Server** views; as you select different domains or children of domains, the views will change accordingly. For information, see ***Setting Security Privileges***.

Known Issues

This section describes the known issues in this release of the Business Services component. The following topics are covered:

- Option to Move Business Services between Domains
- Support for Redefined Fields
- Java Client Generation when PDA Fields Contain Direction Modifier Comments

Option to Move Business Services between Domains

Currently, you cannot move or copy a business service from one domain to another. If you do, an error will occur and the service will appear to be in the old domain.

Support for Redefined Fields

When creating Java classes or Web services, redefined fields are not supported and the original field will be used. This behavior is not consistent with Natural Business Services 5.*n* installations, which support redefined fields.

Java Client Generation when PDA Fields Contain Direction Modifier Comments

When generating a Java client from a business service (.bsrv file), ensure the underlying Natural subprogram data areas do not contain direction modifier comments in the parameter fields, such as "/* in" or "/* out" and the "BY VALUE" (IN direction) option. These are currently not supported. If they are used, the resulting generated code will contain compile errors.

2 What's New in Version 8.2.2

This section describes the changes to the Business Services component in version 8.2.2. The following topics are covered:

Enhancements

This section describes the changes in this release of the Business Services component. The following topics are covered:

- Perform Actions on Business Service Resources
- Redesign the Interface during Java Client Generation

Perform Actions on Business Service Resources

You can now use the **Natural Server** view to perform standard actions on business service resources on the server. For information, see ***Perform Standard Actions on Business Service Resources***.

Redesign the Interface during Java Client Generation

When using the Java client generation wizard, you can now redesign the interface (for example, select which portion of a redefined field to use for a Web service). For information, see ***Generate a Java Client***.

3 What's New in Version 8.2.3

This version contains several error corrections. New functionality is not provided.

4 What's New in Version 8.2.4

This section describes the changes to the Business Services component in version 8.2.4. The following topics are covered:

Enhancements

This section describes the changes in this release of the Business Services component. The following topics are covered:

- New Preferences for Business Services
- Support for WS-Security in Generated Java Code
- Run a Dynamic RPC Java Class

New Preferences for Business Services

Using the new **Installation** preference page, you can control whether or not UI functions are visible (for example, context menu actions and **Natural Server** view nodes), based on the availability of Business Services on the Natural server. For information, see ***Set Installation Preferences***.

Support for WS-Security in Generated Java Code

The generated Java code now supports WS-Security. When WS-Security is enabled for a Web service and credentials are passed in the SOAP header, they will be used to issue the call to Broker. For information, see ***Configure the Web Service to Use WS-Security***.

Run a Dynamic RPC Java Class

You can now set the nbs.properties file to run a dynamic RPC Java class. For information, see ***Set the nbs.properties File to Run a Dynamic RPC Java Class***.

In addition, you can define different settings in the nbs.properties file for individual services and/or groups of services. For information, see ***Define Different User IDs and Passwords for Web Services***.

5 What's New in Version 8.2.5

This section describes the changes to the Business Services component in version 8.2.5. The following topics are covered:

Enhancements

This section describes the changes in this release of the Business Services component. The following topics are covered:

- Changes to the Test Editors
- Changes to the Steplib Wizard When Natural Security is Installed

Changes to the Test Editors

All toolbar controls for the Test editors are now available in the editor toolbar. These controls were previously located in the Eclipse toolbar.

Changes to the Steplib Wizard When Natural Security is Installed

When Natural Security is installed, you can now use the steplib wizard to copy the Natural Security steplib definitions for a selected library when creating a new steplib chain. By default, the steplib definitions for a project were copied to the NBS steplib when the corresponding project was not secured and was not using Natural Security.

Notes:

1. For information, see ***Generating a New Steplib***.
2. To change the default steplib options, see ***Set Business Service Preferences***.

6 What's New in Version 8.2.6

This version contains several error corrections. New functionality is not provided.

7 What's New in Version 8.2.7

This version contains several error corrections. New functionality is not provided.

II What is the Business Service Component

8 What is the Business Service Component

Similar to how NaturalONE allows you to work with Natural objects in Eclipse, the Business Service component for NaturalONE allows you to create and maintain business services in Eclipse. You can then upload the generated components to the server.

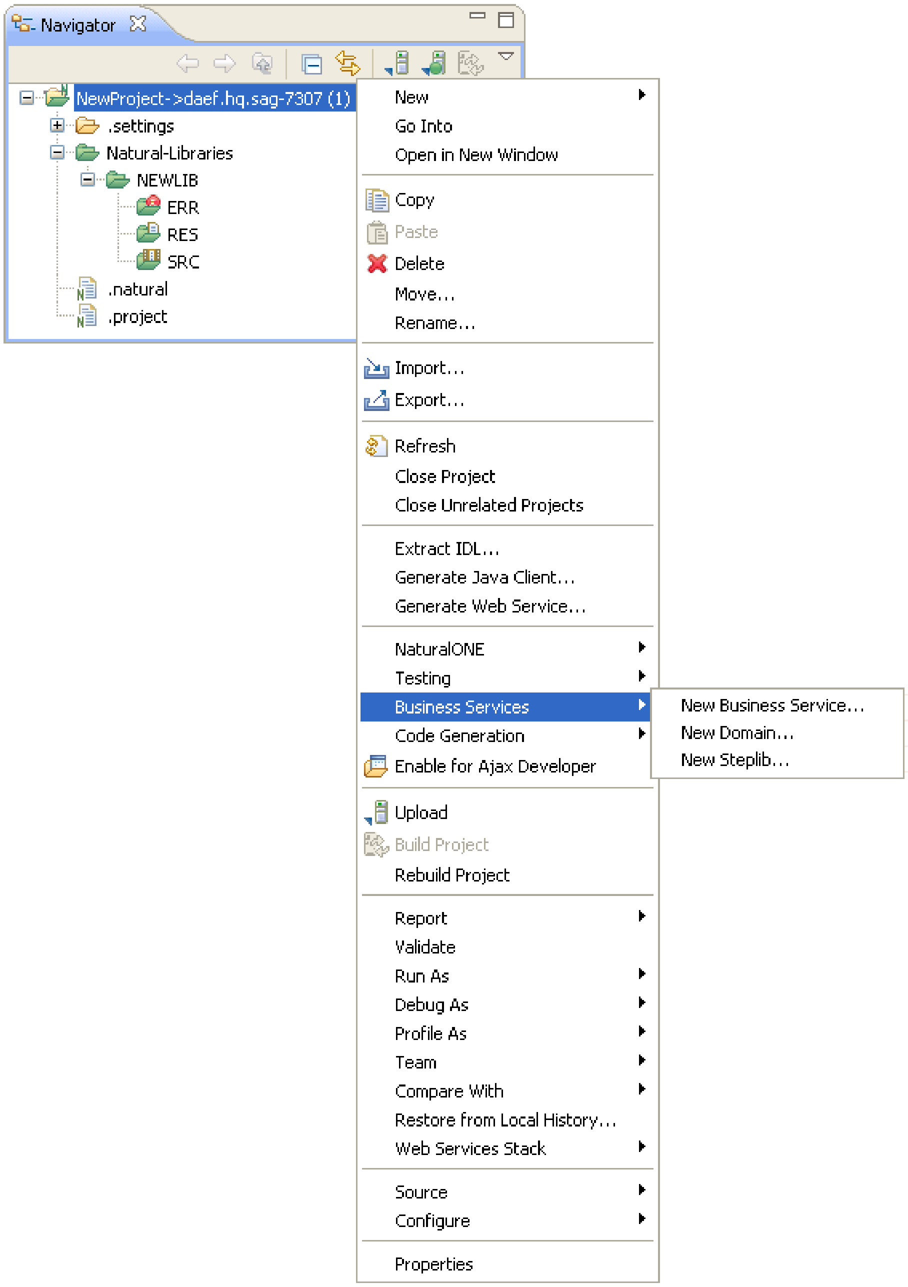
Business services are grouped by domains, which define the following settings:

- Security permissions for one or more users
- A steplib chain file indicating where the Natural objects for a business service are stored

The services are created locally in a NaturalONE project using the business service wizard or downloaded from a Natural Business Services installation by defining server connections in NaturalONE and then using a wizard to select the services. You can also use a wizard to generate clients for business services, such as Java clients and Web services.

Note: To install the Business Service component, you must select **Designer > NaturalONE > Service Development** in the installation tree for the installer.

The Business Service component supplied with NaturalONE provides the following functions:



Note: The Business Service component must be initiated from a NaturalONE project in the NaturalONE perspective. For information, see the NaturalONE documentation.

Using the Business Service component, you can:

Task	Procedure
Download business services data from a Natural Business Services installation on the server.	See *Downloading Business Services from the Server*.
Create a new business service in NaturalONE.	See *Creating a New Business Service*.
Generate a new domain.	See *Generating a New Domain*.
Generate a new steplib.	See *Generating a New Steplib*.

III

9 Before You Start

This part contains information on the prerequisites for using the Business Service component. The following topics are covered:

10 Setting Up a Natural Business Services Environment

To use the Business Service component of NaturalONE, a Natural Business Services (NBS) environment must be available in the **Natural Server** view. Although business services can be generated into a project connected to the local Natural runtime environment, they cannot be executed in the NBS runtime environment.

NaturalONE only supports RPC servers. Natural Business Services version 5.3.1 service pack 4 or higher is required to execute business services via the Natural RPC server.

Notes:

1. Business services created on an RPC server can be used in other NBS environments (for example, the Dispatch server environment).
2. For information on installing and configuring a Natural Business Services environment, see *Natural Business Services Installation on Mainframes*.

11 Setting Up a Test Environment

To test subprograms and business services directly, and to create unit tests for subprograms and business services, a Natural RPC server is required. The Natural Development Server cannot be used in this context. If you are testing items in a project connected to the local Natural runtime environment, a special connection via RPC must be made.

Note: For information on testing business services and subprograms, see *Application Testing*.

IV Getting Started

This part contains step-by-step instructions for the following topics:

12 Creating a New Business Service

Create a Natural Project

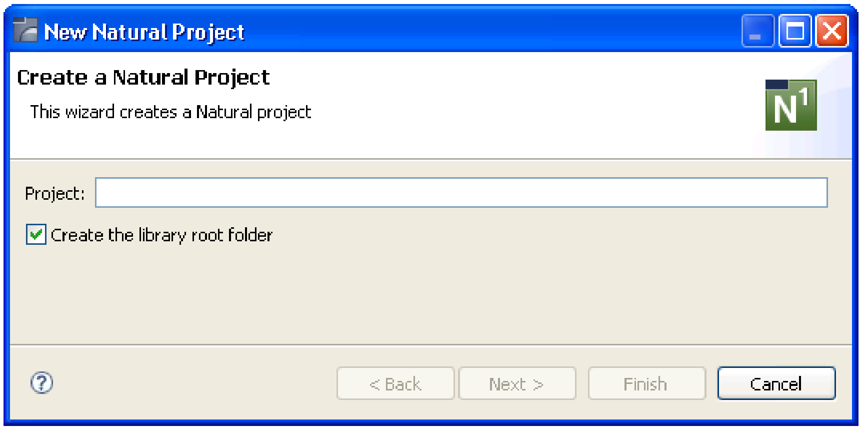
Before creating a new business service, you must create a local project in which to generate the business service data.

To create a Natural project:

1 Open the NaturalONE perspective.

2 Select **New > Natural Project** on the **File** menu.

The **Create a Natural Project** window is displayed. For example:



New Natural Project

Create a Natural Project

This wizard creates a Natural project

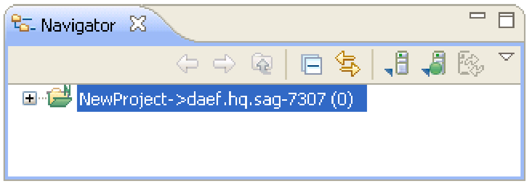
Project:

☑ Create the library root folder

3 Type "NewProject" in **Project**.

4 Select **Finish**.

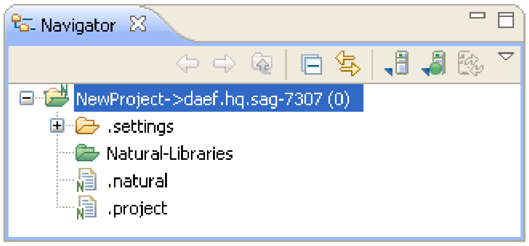
The new project is displayed in the **Navigator** view. For example:



NewProject->daef.hq.sag-7307 (0)

5 Expand the **NewProject** node.

The standard project components are displayed. For example:



Note: For more information on creating a Natural project, refer to the NaturalONE documentation.

Create a New Library

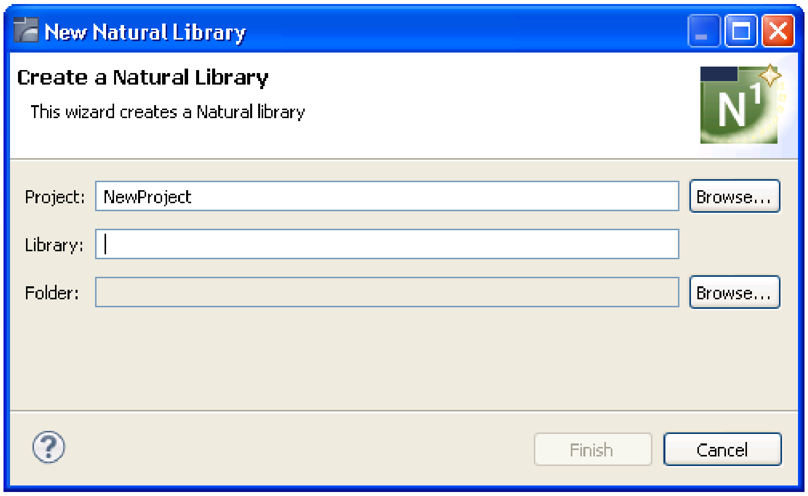
This section describes how to create a new library in which to store the new business service.

To create the library:

1 Open the context menu for **NewProject** in the **Navigator** view.

2 Select **New > Natural Library**.

The **Create a Natural Library** panel is displayed. For example:


New Natural Library

Create a Natural Library

This wizard creates a Natural library

Project: NewProject

Library:

Folder:

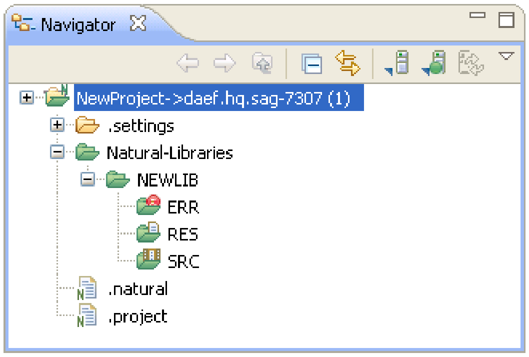
3 Type "NEWLIB" in **Library**.

4 Select **Finish**.

The library is created locally.

5 Expand the **Natural-Libraries** node in the **Navigator** view.

The new library and sub-folders are displayed. For example:



Access the Business Services Functions

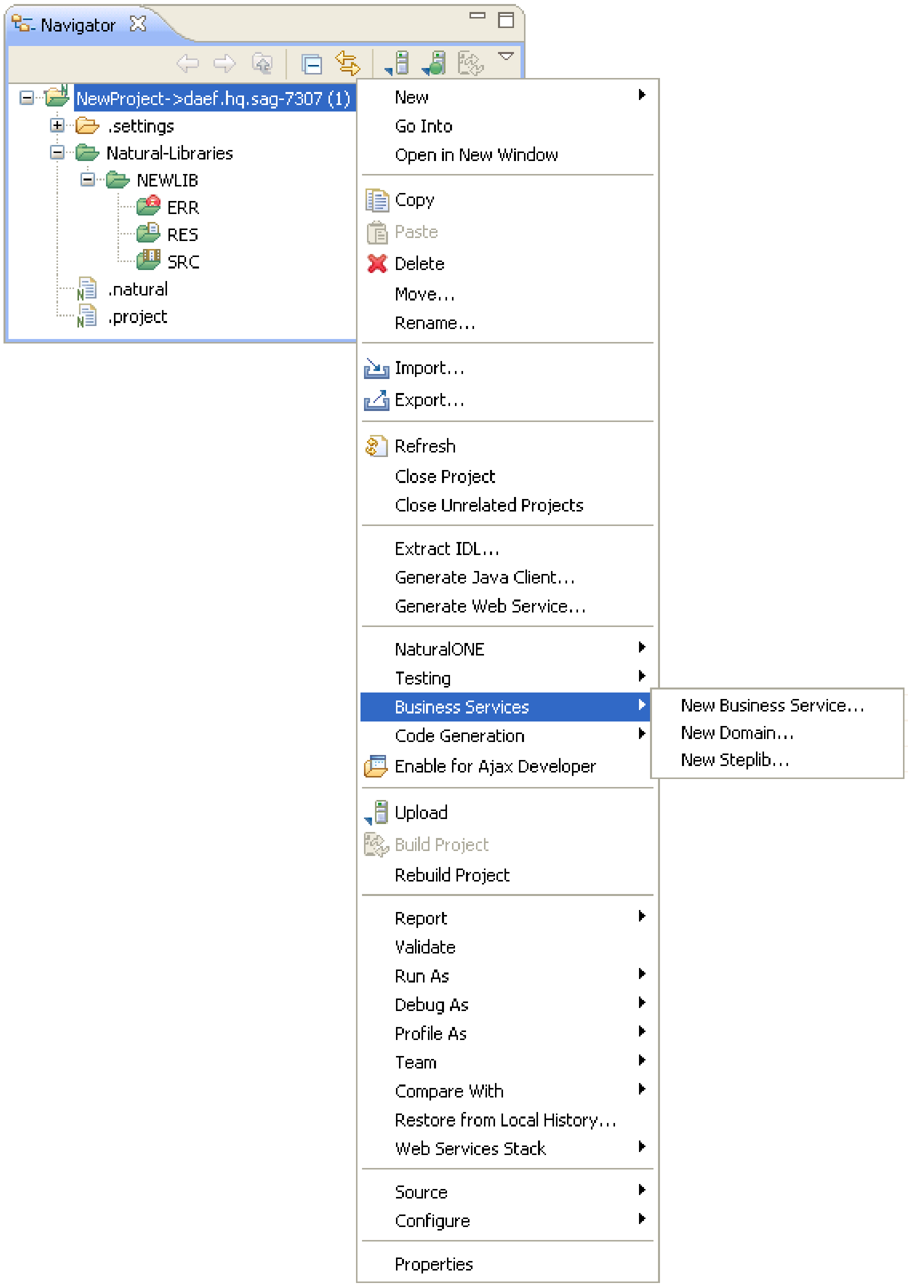
This section describes how to access the Business Services functions.

To access the Business Services functions:

1 Open the context menu for **NewProject** in the **Navigator** view.

2 Select **Business Services**.

The Business Service functions are displayed. For example:


Navigator
NewProject->daef.hq.sag-7307 (1)
.settings
Natural-Libraries
NEWLIB
ERR
RES
SRC
.natural
.project
New
Go Into
Open in New Window
Copy
Paste
Delete
Move...
Rename...
Import...
Export...
Refresh
Close Project
Close Unrelated Projects
Extract IDL...
Generate Java Client...
Generate Web Service...
NaturalONE
Testing
Business Services
Code Generation
Enable for Ajax Developer
Upload
Build Project
Rebuild Project
Report
Validate
Run As
Debug As
Profile As
Team
Compare With
Restore from Local History...
Web Services Stack
Source
Configure
Properties
New Business Service...
New Domain...
New Steplib...

Generate the New Service

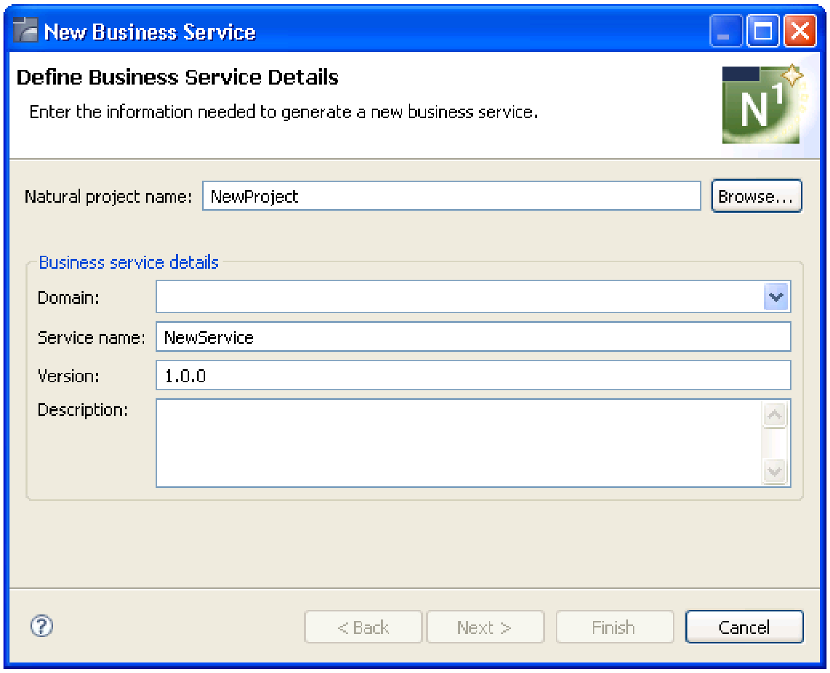
You are now ready to use the Business Service wizard to generate your business service. During generation, the wizard also creates default objects required by the service, such as the domain and steplib chain files. After generation, you can edit the default information about these objects as required.

To generate the new business service:

1 Open the context menu for **NewProject** in the **Navigator** view.

2 Select **Business-Services > New Business Service**.

The **Define Business Service Details** panel is displayed, showing the name of the project. For example:



Define Business Service Details

Enter the information needed to generate a new business service.

Natural project name: NewProject

Business service details

Domain:

Service name: NewService

Version: 1.0.0

Description:

3 Select "NEWDOM" in **Domain**.

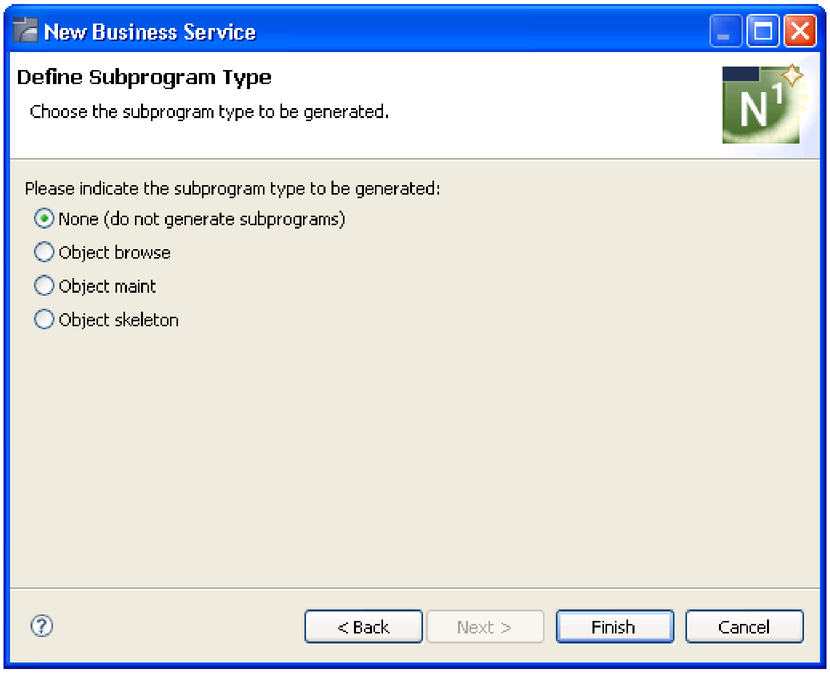
The domain will be created locally, along with an associated step library entry that points to a steplib chain with the same name as the domain.

Note: If you specify the name of an existing domain, the wizard will check the server connection and download any required data from the NBS repository. **Domain** only contains domain entries that have already been downloaded.

4 Type "This is a new business service." in **Description**.

5 Select **Next**.

The **Define Subprogram Type** panel is displayed. For example:



New Business Service

Define Subprogram Type

Choose the subprogram type to be generated.

Please indicate the subprogram type to be generated:

● None (do not generate subprograms)

○ Object browse

○ Object maint

○ Object skeleton

Note: By default, no subprograms are generated. For more information about creating the subprogram types on this panel, see *Code Generation*.

6 Select **Object browse**.

7 Select **Next**.

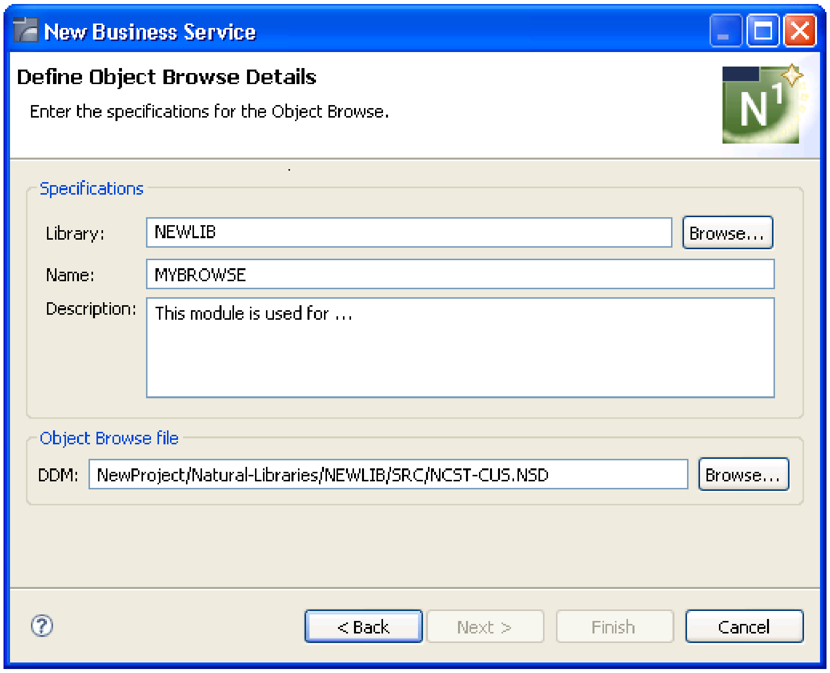
The **Define Object Browse Details** panel is displayed.

8 Select "NEWLIB" in **Library**.

9 Type "MYBROWSE" in **Name**.

10 Select "NewProject/Natural-Libraries/NEWLIB/SRC/NCST-CUST.NSD" in **DDM**.

For example:



Library: NEWLIB

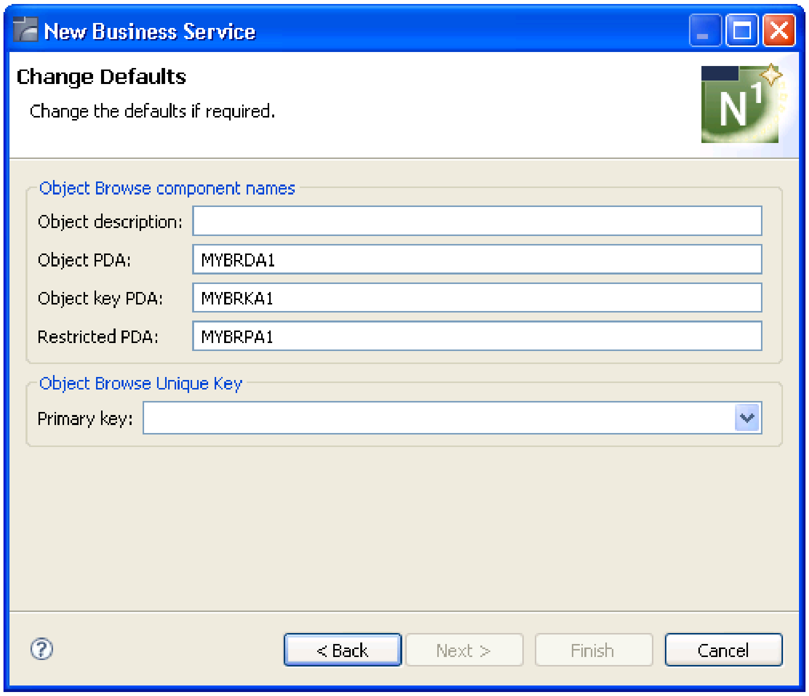
Name: MYBROWSE

Description: This module is used for ...

DDM: NewProject/Natural-Libraries/NEWLIB/SRC/NCST-CUS.NSD

11 Select **Next**.

The **Change Defaults** panel is displayed. For example:



Object description:

Object PDA: MYBRDA1

Object key PDA: MYBRKA1

Restricted PDA: MYBRPA1

Primary key:

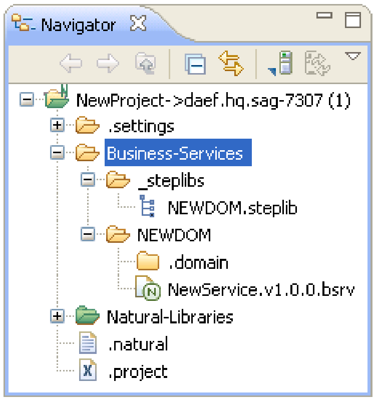
This panel shows the default specification values for the browse subprogram to be generated. Using this panel, you can:

Task	Procedure
Change the name of the parameter data area (PDA) for the object.	Type the name in **Object PDA**.
Change the name of the local data area (LDA) for the object.	Type the name in **Object LDA**. **Note:** The local data area is only required when the hash-locking option is used for record locking.
Change the name of the restricted parameter data area (PDA) for the object.	Type the name in **Restricted PDA**.
Define the primary key field used for browse operations.	Select the field in **Primary key**.

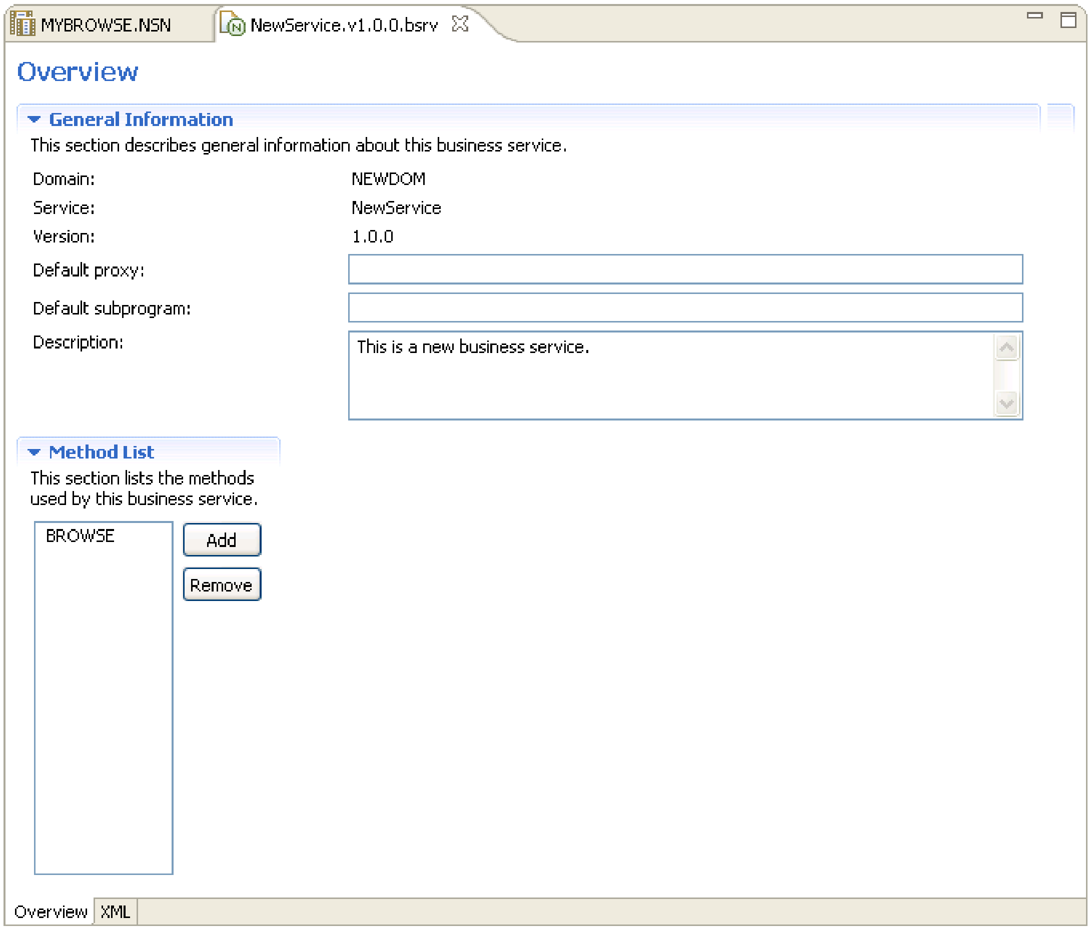
12 Type a description of the browse subprogram in **Object description**.

13 Select **Finish** to generate the business service file, the associated domain and steplib files, and the corresponding Natural objects.

The generated items are added to NewProject and the new business service is now listed in the **Navigator** view. For example:



Details about the new service are displayed in the business service editor. For example:



Overview

General Information

This section describes general information about this business service.

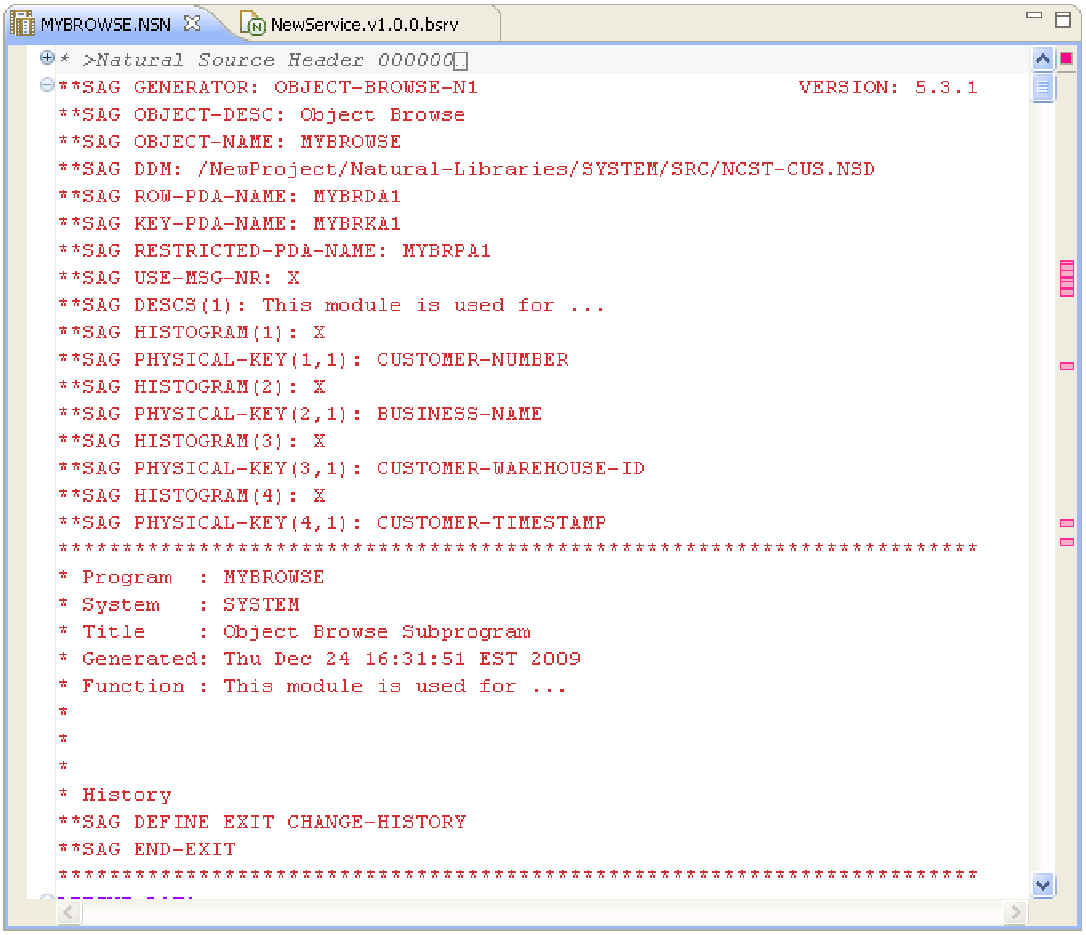
Domain: NEWDOM
Service: NewService
Version: 1.0.0
Default proxy:
Default subprogram:
Description: This is a new business service.

Method List

This section lists the methods used by this business service.

BROWSE

The generated object-browse subprogram is displayed in the editor view. For example:



```
* >Natural Source Header 000000
**SAG GENERATOR: OBJECT-BROWSE-N1                         VERSION: 5.3.1
**SAG OBJECT-DESC: Object Browse
**SAG OBJECT-NAME: MYBROWSE
**SAG DDM: /NewProject/Natural-Libraries/SYSTEM/SRC/NCST-CUS.NSD
**SAG ROW-PDA-NAME: MYBRDA1
**SAG KEY-PDA-NAME: MYBRKA1
**SAG RESTRICTED-PDA-NAME: MYBRPA1
**SAG USE-MSG-NR: X
**SAG DESCS(1): This module is used for ...
**SAG HISTOGRAM(1): X
**SAG PHYSICAL-KEY(1,1): CUSTOMER-NUMBER
**SAG HISTOGRAM(2): X
**SAG PHYSICAL-KEY(2,1): BUSINESS-NAME
**SAG HISTOGRAM(3): X
**SAG PHYSICAL-KEY(3,1): CUSTOMER-WAREHOUSE-ID
**SAG HISTOGRAM(4): X
**SAG PHYSICAL-KEY(4,1): CUSTOMER-TIMESTAMP
************************************************************************
* Program  : MYBROWSE
* System   : SYSTEM
* Title    : Object Browse Subprogram
* Generated: Thu Dec 24 16:31:51 EST 2009
* Function : This module is used for ...
*
*
*
* History
**SAG DEFINE EXIT CHANGE-HISTORY
**SAG END-EXIT
************************************************************************
```

14 Save the business service data.

Notes:

1. By default, the settings defined in **Properties > Natural > Steplibs** on the context menu for the project will be used as the default steplib chain.
2. You can upload the business service data to the server using standard NaturalONE functionality. For information, see the NaturalONE documentation.

At this point, the business service has been created along with a default domain and steplib chain (list of libraries that includes the step library containing the subprogram that implements the service method). If required, you will now edit the default information for the domain and steplib files:

- Edit the Default Domain Information

- Edit the Default Steplib Information

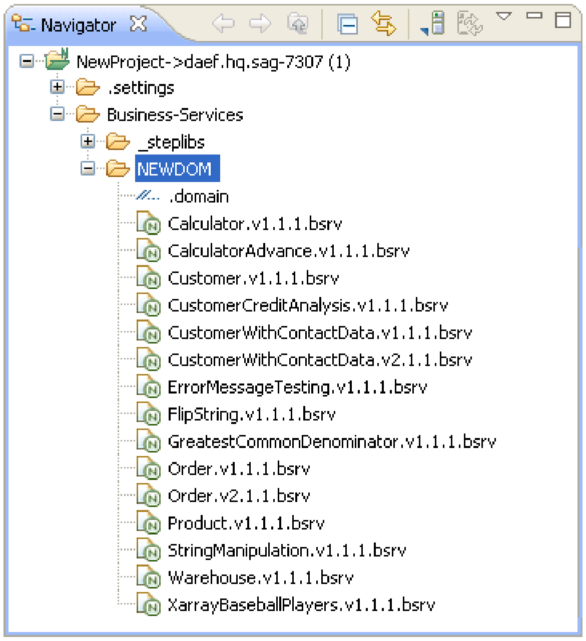
Edit the Default Domain Information

In this example, you will provide a brief description of the default domain generated by the Business Service wizard.

To edit the default domain information:

1 Expand the **NEWDOM** node in the **Navigator** view.

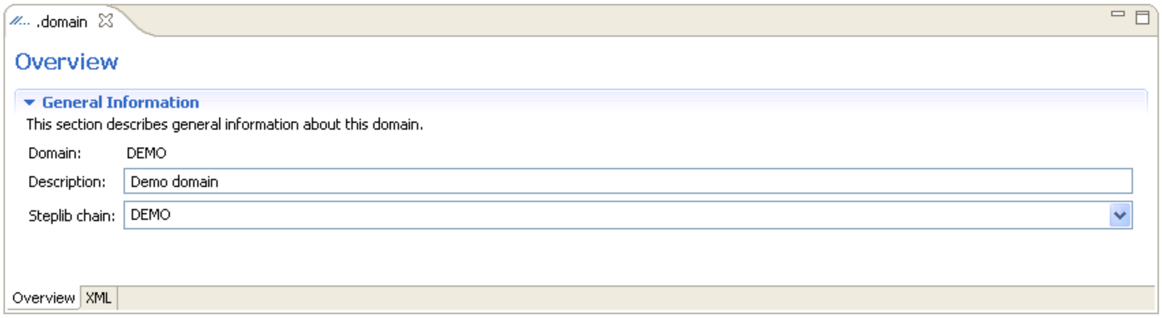
For example:



```
NewProject->daef.hq.sag-7307 (1)
  .settings
  Business-Services
    _steplibs
    NEWDOM
      .domain
      Calculator.v1.1.1.bsrv
      CalculatorAdvance.v1.1.1.bsrv
      Customer.v1.1.1.bsrv
      CustomerCreditAnalysis.v1.1.1.bsrv
      CustomerWithContactData.v1.1.1.bsrv
      CustomerWithContactData.v2.1.1.bsrv
      ErrorMessageTesting.v1.1.1.bsrv
      FlipString.v1.1.1.bsrv
      GreatestCommonDenominator.v1.1.1.bsrv
      Order.v1.1.1.bsrv
      Order.v2.1.1.bsrv
      Product.v1.1.1.bsrv
      StringManipulation.v1.1.1.bsrv
      Warehouse.v1.1.1.bsrv
      XarrayBaseballPlayers.v1.1.1.bsrv
```

2 Open the *.domain* file.

The domain editor is displayed. For example:



General Information

This section describes general information about this domain.

Domain: DEMO

Description: Demo domain

Steplib chain: DEMO

3 Type "This domain contains the sample business service." in **Description**.

4 Save the domain file.

Note: You can upload the domain file to the server using standard NaturalONE functionality. For information, see the NaturalONE documentation.

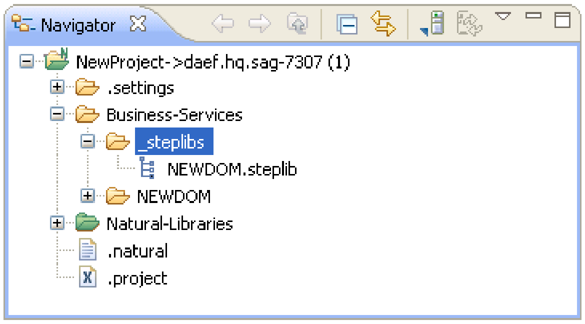
Edit the Default Steplib Information

In this example, you will provide the database ID and file number for the default steplib chain generated by the Business Service wizard.

To edit the default steplib information:

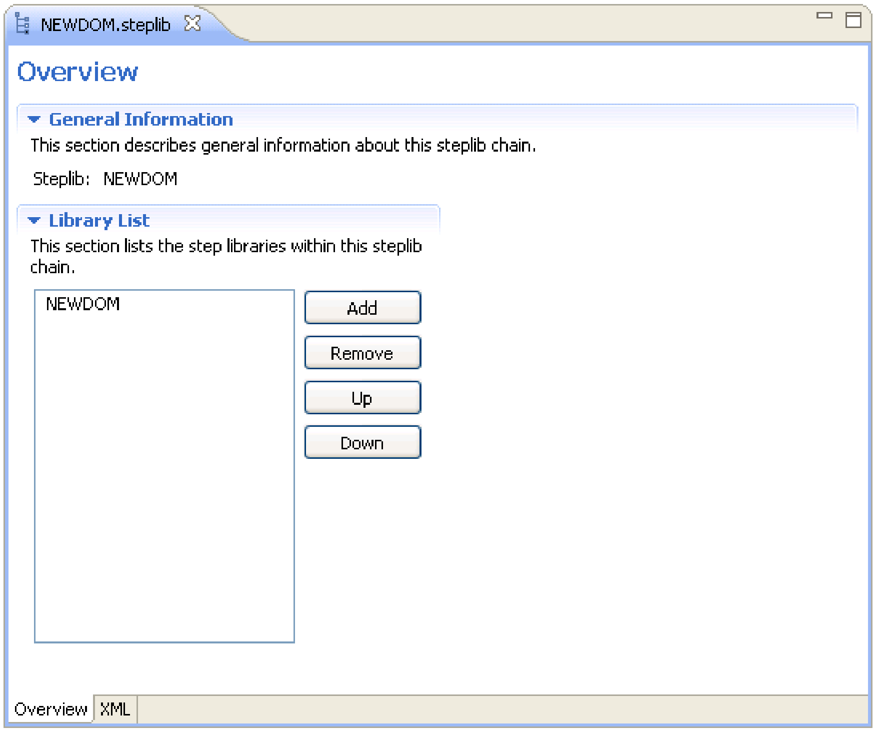
1 Expand the **_steplibs** node in the **Navigator** view.

For example:



2 Open the *NEWDOM.steplib* file.

The steplib editor is displayed. For example:



This section describes general information about this steplib chain.

Steplib: NEWDOM

This section lists the step libraries within this steplib chain.

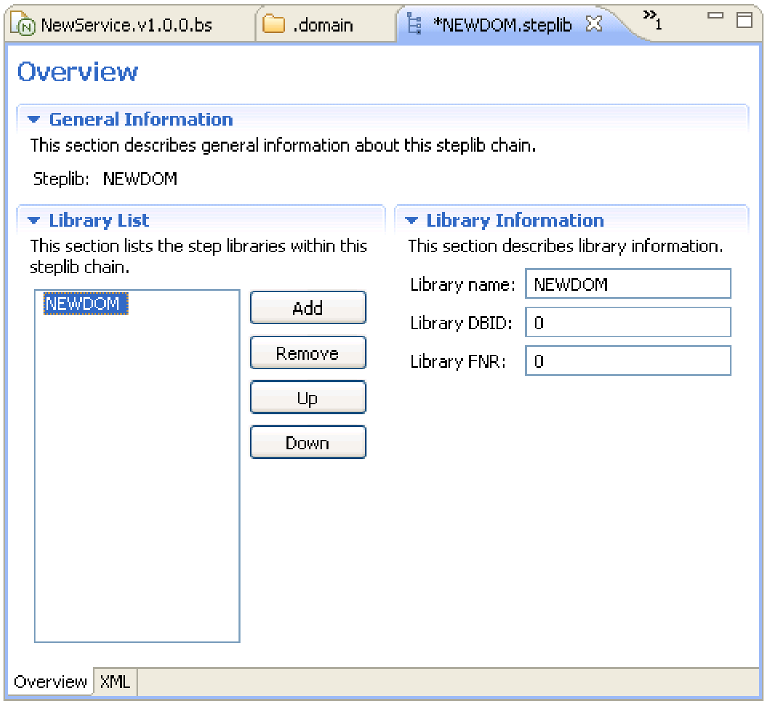
NEWDOM

A step library with the same name as the steplib chain is displayed in the **Library List** section.

Tip: The settings defined in **Properties > Natural > Steplibs** on the context menu for the project can be used to populate the **Library List**.

3 Select **NEWDOM** in the **Library List** section.

The **Library Information** section is displayed. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWDOM

Library List

This section lists the step libraries within this steplib chain.

NEWDOM

Library Information

This section describes library information.

Library name: NEWDOM

Library DBID: 0

Library FNR: 0

Note: The DBID and FNR values are not required; they are only used in advanced configurations when uploading to the server.

4 Save the steplib file.

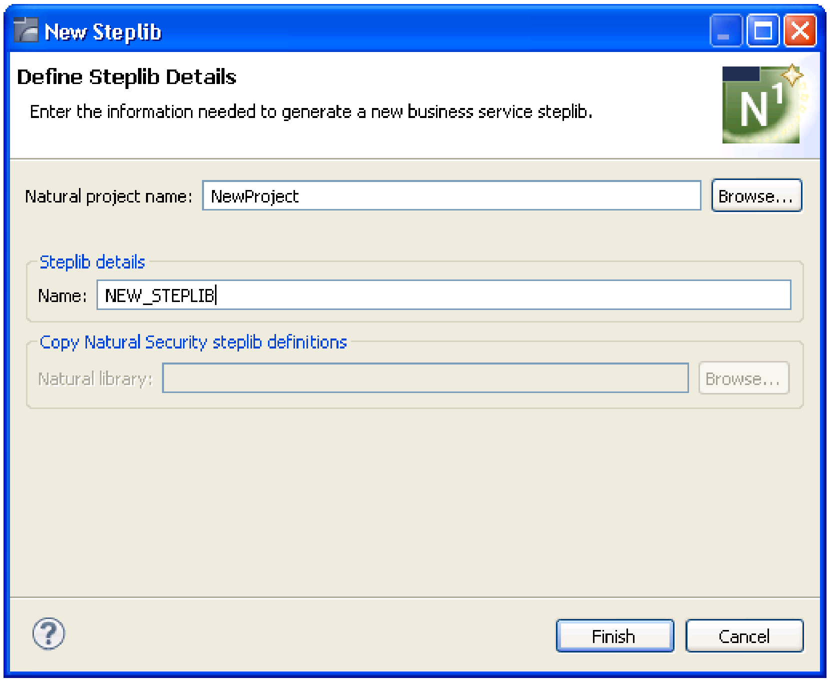
13 Generating a New Steplib

This section describes how to generate a new steplib chain for your business services. This file defines a set of Natural libraries that a business service dispatch server must access to call a subprogram or proxy. The dispatch server accesses the libraries in the order they are listed in the steplib file.

To create a new steplib:

1. Open the context menu for **NewProject** in the **Navigator** view.
2. Select **Business-Services > New Steplib**.

   The **Define Steplib Details** panel is displayed. For example:



New Steplib

Define Steplib Details

Enter the information needed to generate a new business service steplib.

Natural project name: NewProject

Steplib details

Name: NEW_STEPLIB

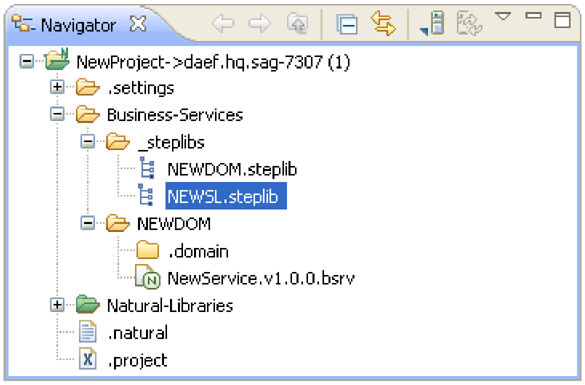
Copy Natural Security steplib definitions

Natural library:

3 Type "NEWSL" over the default name in **Name**.

4 Select **Finish** to generate the steplib file.

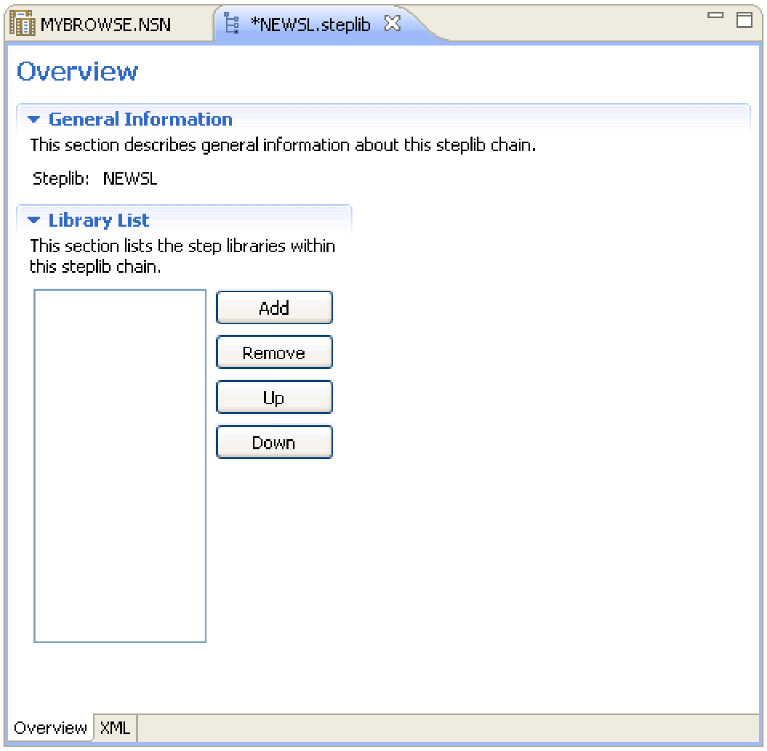
The generated items are added to your project and the new steplib is now listed in the **Navigator** view. For example:



Navigator

```
NewProject->daef.hq.sag-7307 (1)
  .settings
  Business-Services
    _steplibs
      NEWDOM.steplib
      NEWSL.steplib
    NEWDOM
      .domain
      NewService.v1.0.0.bsrv
  Natural-Libraries
  .natural
  .project
```

The steplib file is displayed in the editor view. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWSL

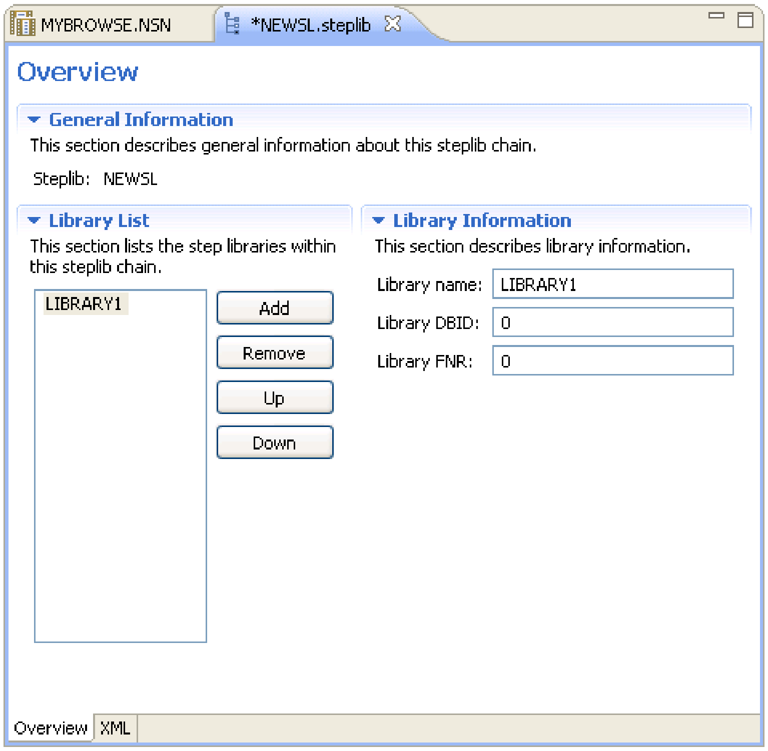
Library List

This section lists the step libraries within this steplib chain.

Tip: The settings defined in **Properties > Natural > Steplibs** on the context menu for the project can be used to populate the **Library List**.

5 Select **Add**.

The Library Information section is displayed. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWSL

Library List

This section lists the step libraries within this steplib chain.

LIBRARY1

Library Information

This section describes library information.

Library name: LIBRARY1

Library DBID: 0

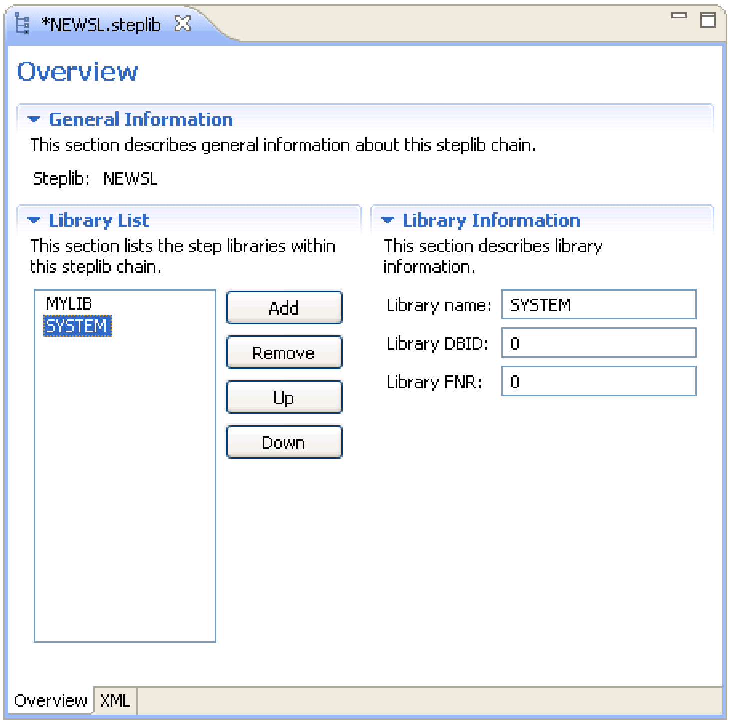
Library FNR: 0

6. Type "MYLIB" over the default name in **Library name**.

   Note: The DBID and FNR values are not required; they are only used in advanced configurations when uploading to the server.

7. Select **Add**.
8. Type "SYSTEM" over the default name in **Library name**.

   The MYLIB and SYSTEM step libraries are now part of the NEWSL steplib chain. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWSL

Library List

This section lists the step libraries within this steplib chain.

MYLIB
SYSTEM

Library Information

This section describes library information.

Library name: SYSTEM

Library DBID: 0

Library FNR: 0

9 Save the steplib file.

14 Generating a New Domain

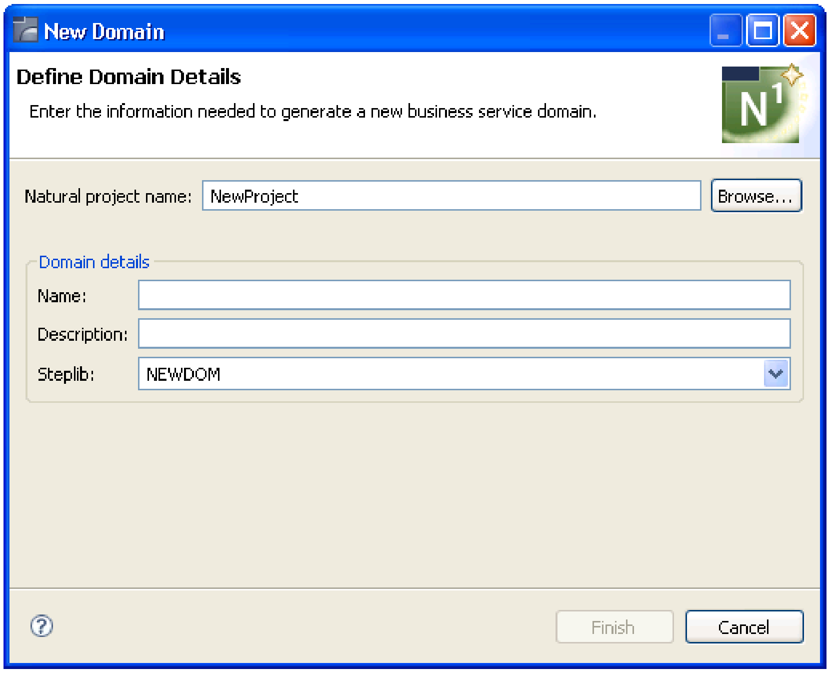
This section describes how to generate a new domain for your business services. Each business service is assigned to a domain, which groups the services according to access privileges and user-defined criteria.

To create a new domain:

1 Open the context menu for **NewProject**.

2 Select **Business-Services > New Domain**.

The **Define Domain Details** panel is displayed. For example:



New Domain

Define Domain Details

Enter the information needed to generate a new business service domain.

Natural project name: NewProject

Domain details

Name:

Description:

Steplib: NEWDOM

3 Type "MYDOM" in **Name**.

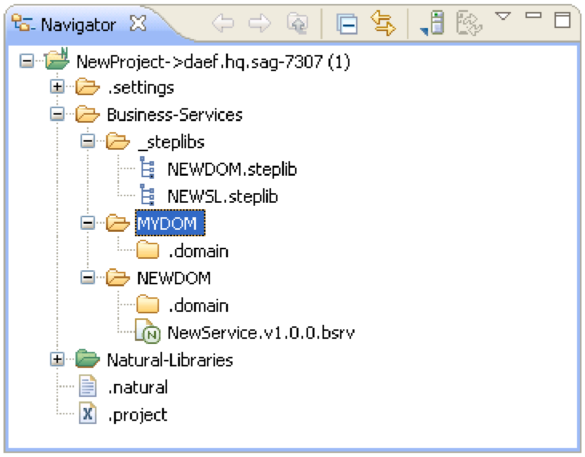
Note: The wizard will check the server connection to verify whether the domain currently exists. If it does, a message is displayed. You can either overwrite the existing domain or enter a new name.

4 Type "This is the new domain for my business service." in **Description**.

5 Select "NEWSL" in **Steplib**.

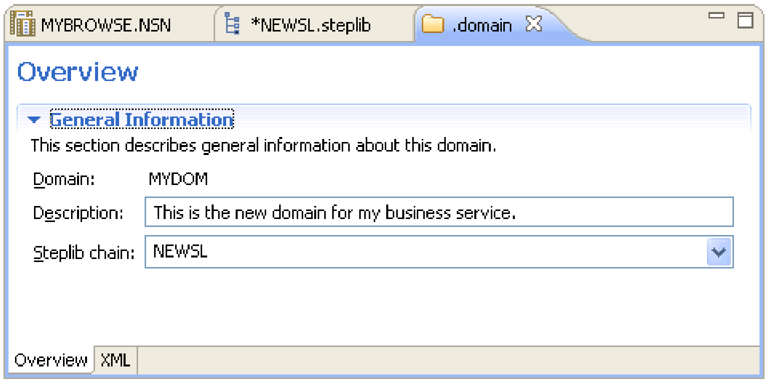
6 Select **Finish** to generate the domain file.

The generated items are added to your project and the new domain is now listed in the **Navigator** view. For example:



```
NewProject->daef.hq.sag-7307 (1)
    .settings
    Business-Services
        _steplibs
            NEWDOM.steplib
            NEWSL.steplib
        MYDOM
            .domain
        NEWDOM
            .domain
            NewService.v1.0.0.bsrv
    Natural-Libraries
    .natural
    .project
```

The domain is displayed in the editor view. For example:



This section describes general information about this domain.

Domain: MYDOM

Description: This is the new domain for my business service.

Steplib chain: NEWSL

7 Save the domain file.

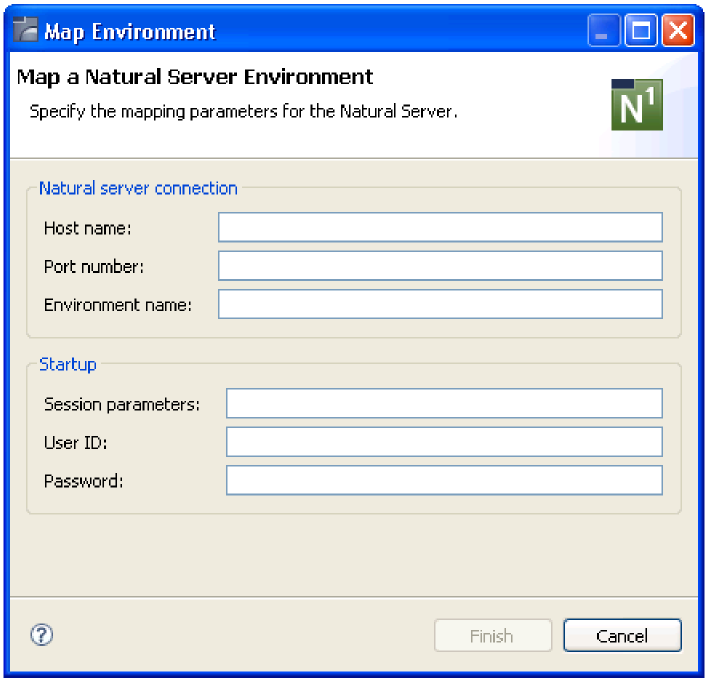
15 Downloading Business Services from the Server

This section describes how to download business services data from a Natural Business Services installation on the server to the local environment. This procedure is performed in two parts: first map to an existing business service installation on the server and then download the services from the **Natural Server** view to the **Navigator** view.

To map to an existing Natural Business Services installation:

1 Select on the toolbar in the **Natural Server** view.

The **Map a Natural Server Environment** panel is displayed. For example:



Map a Natural Server Environment

Specify the mapping parameters for the Natural Server.

2 Type the name of the host for the Natural Business Services installation in **Host name**.

3 Type the port number in **Port number**.

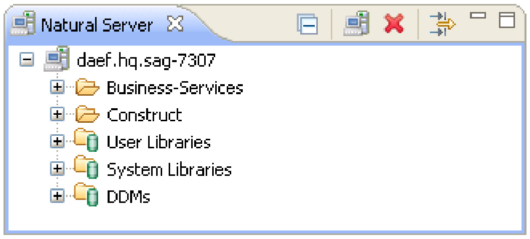
The name of the environment is derived from the host and port values and displayed in **Environment name** when you select the field. You can change this name if desired.

4 Type the Natural profile name for the environment in **Session parameters**.

5 Type the user credentials for the server in **User ID** and **Password**.

6 Select **Finish**.

The connection is displayed in the **Natural Server** view. Expand the connection node to see the contents. For example:



To download business services from the Natural Server view to the Navigator view:

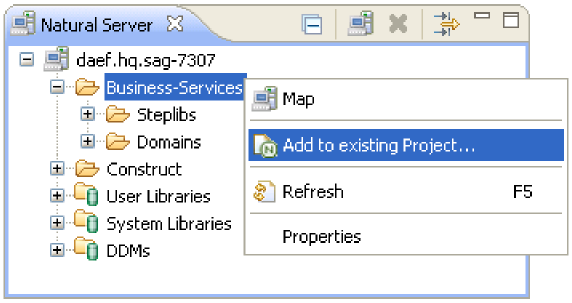
1 Open the context menu for the **Business-Services** root node in the **Natural Server** view.

Or:

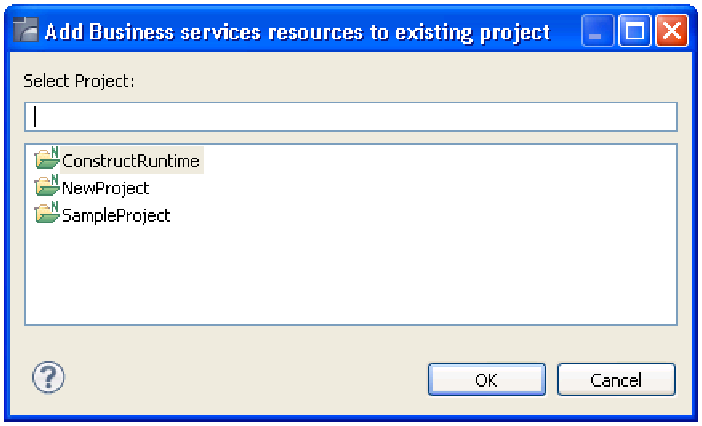
Expand the root node and select one or more steplib and/or domain nodes or files using standard selection techniques.

2 Select **Add to existing Project**.

For example:



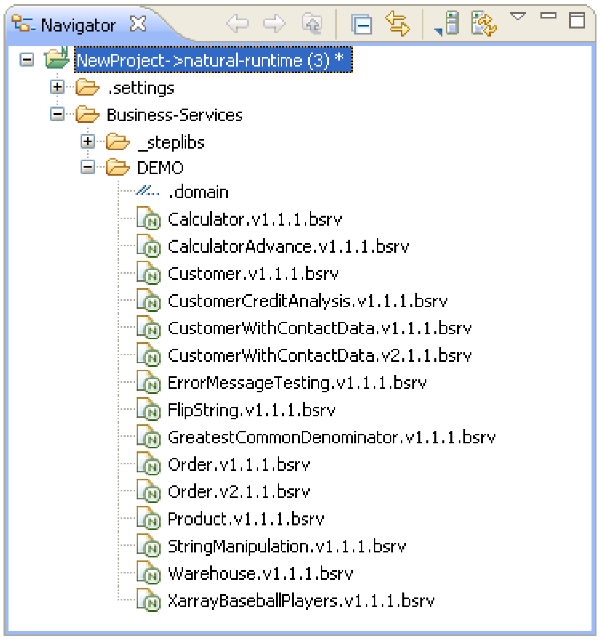
A list of available projects is displayed. For example:



3 Select **NewProject**.

4 Select **OK**.

A progress window is displayed as the business services are downloaded from the server to the local project in the **Navigator** view. Expand the **Business-Services** root node to display the downloaded business services. For example:



```
NewProject->natural-runtime (3) *
  .settings
  Business-Services
    _steplibs
    DEMO
      .domain
      Calculator.v1.1.1.bsrv
      CalculatorAdvance.v1.1.1.bsrv
      Customer.v1.1.1.bsrv
      CustomerCreditAnalysis.v1.1.1.bsrv
      CustomerWithContactData.v1.1.1.bsrv
      CustomerWithContactData.v2.1.1.bsrv
      ErrorMessageTesting.v1.1.1.bsrv
      FlipString.v1.1.1.bsrv
      GreatestCommonDenominator.v1.1.1.bsrv
      Order.v1.1.1.bsrv
      Order.v2.1.1.bsrv
      Product.v1.1.1.bsrv
      StringManipulation.v1.1.1.bsrv
      Warehouse.v1.1.1.bsrv
      XarrayBaseballPlayers.v1.1.1.bsrv
```

16 Testing a Business Service

You can test a business service or subprogram directly and/or create unit tests for business services and subprograms. For information, see *Application Testing*.

V Service Development

This part covers the following topics:

17 Creating a New Business Service

This section describes how to create a new business service. The Business Service wizard quickly generates the business service based on a few simple input values. In addition, the wizard generates default objects required by the service. These include:

Object Generated	Description
Domain file	Groups business services and defines the boundaries of a business service. Each business service is assigned to a domain.
Steplib chain file	Defines a set of Natural libraries that a business service dispatch server must access to call a subprogram or proxy. The dispatch server accesses the libraries in the order they are listed in the steplib file. **Note:** By default, the generated steplib will contain the libraries listed in **Properties > Natural > Steplibs** on the context menu for the project.
Methods list	Lists the methods used for this business service. The type of methods listed depends on the type of service generated.

After generating a business service, you can edit the default information that was generated for the objects.

Notes:

1. You cannot change the key fields, such as the file or domain names.
2. Always use the corresponding editors (business service, domain, and steplib) to edit business services files. Manually editing these files outside of the editors or NaturalONE perspective can potentially cause problems. The XML tab attached to these editors can only be used to view the underlying XML file.

This section covers the following topics:

Create a NaturalONE Project

All business service components are stored locally in a NaturalONE project. If you do not want to use an existing NaturalONE project to store the new business service, you can create a new project. For information, see the NaturalONE documentation.

The following example shows the Business Services folder structure within a NaturalONE project:

- NaturalONE_project_name
  - Business_Services
    - _steplibs
      - steplib1.steplib
      - steplib2.steplib

- DOMAIN1
  - .domain
  - BusinessService1.v1.0.0.bsrv
  - BusinessService2.v1.0.0.bsrv
- DOMAIN2
  - .domain
  - BusinessService3.v1.0.0.bsrv
  - BusinessService4.v1.0.0.bsrv

These folders and files are:

Folder/File	Description
Business_Services folder	Top-level folder with a static name that stores all business service-related files. This folder can be deleted (which deletes all business service-related files within it), but must not be renamed.
_steplibs folder	Top-level folder with a static name that stores all steplib chains (.steplib files). This folder can be deleted (which deletes all steplib files within it), but must not be renamed.
*.steplib file	XML representation of a steplib. Use this file to perform steplib-related operations (delete, edit using the Steplib editor, upload to the Natural server using the Upload command, etc.). This file must not be renamed.
`DOMAIN_NAME` folder	Folder with the name of the domain to make it visually easier to see the business services within the domain. The actual XML representation of a domain is the .domain file located within this folder. This folder can be deleted (which deletes all business services files within it), but must not be renamed.
.domain file	XML representation of a domain. You interact with this file to perform domain-related operations (i.e., edit the file using the domain editor and upload the file to the Natural server using the Upload command). This file must not be renamed. **Note:** To delete a domain, you must delete the domain folder (you cannot delete the .domain file only).
*.bsrv file	XML representation of a business service. You interact with this file to perform business service-related operations (delete, edit using the business service editor, upload to the Natural server using the Upload command, Test, etc.). This file must not be renamed.

Create the Business Service

This section describes how to create a new business service in a NaturalONE project. Once a business service has been generated, you can provide further details about the service, its domain, and its steplib chain. This section covers the following topics:

Note: The business service file is created locally. Use standard NaturalONE functionality to upload the file to the server.

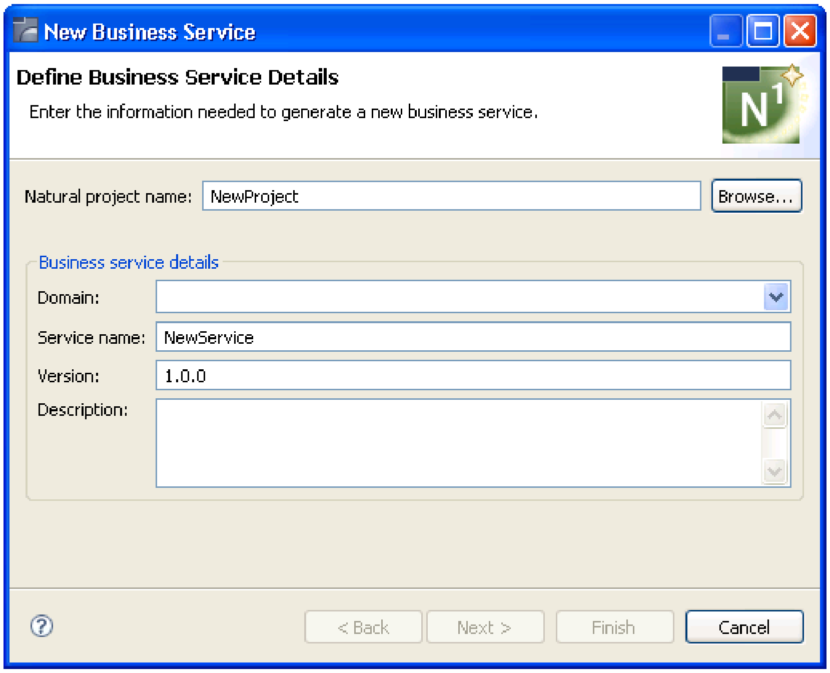
Generate the Business Service

▶ **To generate the new business service:**

1 Open the context menu for the project.

2 Select **Business-Services > New Business Service**.

The **Define Business Service Details** panel is displayed, showing the name of the project. For example:



Define Business Service Details

Enter the information needed to generate a new business service.

Natural project name: NewProject

Business service details

Domain:

Service name: NewService

Version: 1.0.0

Description:

Note: If this is not the project into which you want to generate the service, select **Browse** to choose a different NaturalONE project.

3 Provide the name of the domain that will contain your business service in **Domain** as follows:

- Type the name of the domain in **Domain**.
  - If the domain currently exists on the server and the NaturalONE project is associated with a server connection, the domain and steplib files will be downloaded when you select **Finish** on this panel.

    Note: This functionality can be modified in the **Preferences** window for business services. For information, see *Set Business Service Preferences*.

  - If the domain is new, it will be created locally along with an associated step library entry that points to a steplib chain with the same name as the domain.

    Note: If required, you can edit the generated entries. For information, see *Define the Domain Information* and *Define the Steplib Chain Information*.

- Select the name of a domain that is currently in the local NaturalONE project from **Domain**.

4 Type the name of the new service in **Service name**.

5 Type the version number for the service in **Version**.

In this case, the version is 1.0.0.

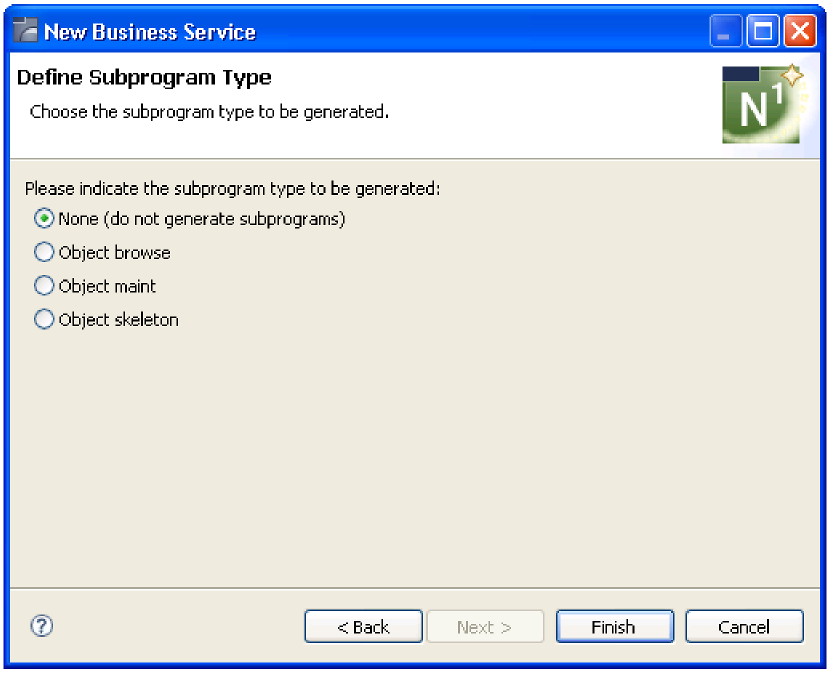
6 Type a brief description of the service in **Description**.

7 Select **Finish** to generate the business service file (and possibly the associated domain and steplib files).

Or:

Select **Next** to also generate the Natural objects for the business service.

The **Define Subprogram Type** panel is displayed.



New Business Service

Define Subprogram Type

Choose the subprogram type to be generated.

Please indicate the subprogram type to be generated:

● None (do not generate subprograms)

○ Object browse

○ Object maint

○ Object skeleton

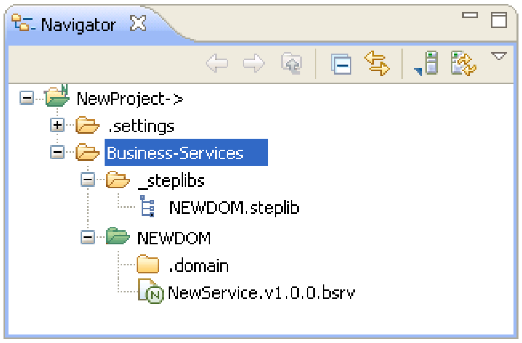
8 Select the subprogram type to be generated.

By default, no subprograms are generated. To generate a subprogram, select the type on this panel and select **Next**. The corresponding wizard is displayed to define the parameters for the subprogram.

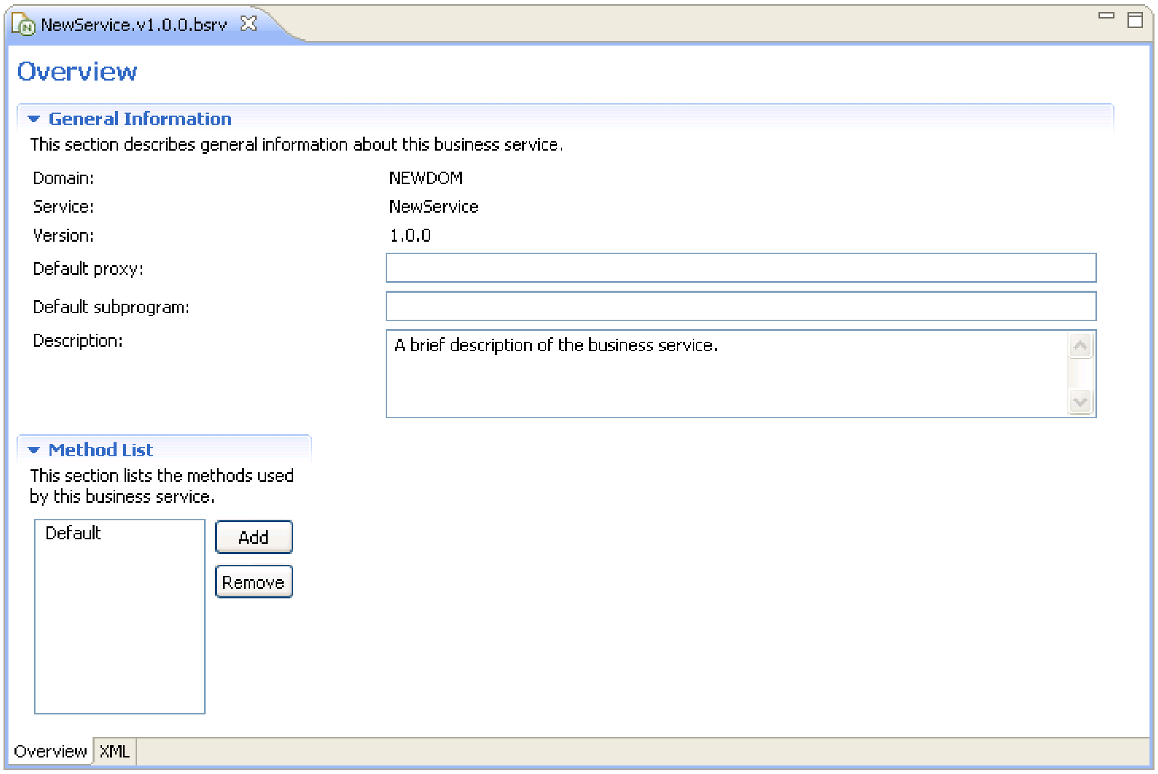
Note: For more information about the subprogram types on this panel, see *Code Generation*.

9 Select **Finish** to generate the business service file and possibly the associated domain and steplib files, along with the corresponding Natural objects.

The generated items are added to your NaturalONE project and the new business service is now listed in the **Navigator** view. For example:



Details about the new service are displayed in the editor view. For example:



Overview

General Information

This section describes general information about this business service.

Domain: NEWDOM
Service: NewService
Version: 1.0.0
Default proxy:
Default subprogram:
Description: A brief description of the business service.

Method List

This section lists the methods used by this business service.

Default

Use this editor to provide a default name for the business service subprogram and proxy.

Note: You must specify a default subprogram for the service if any of the methods do not specify a subprogram.

10 Save the business service.

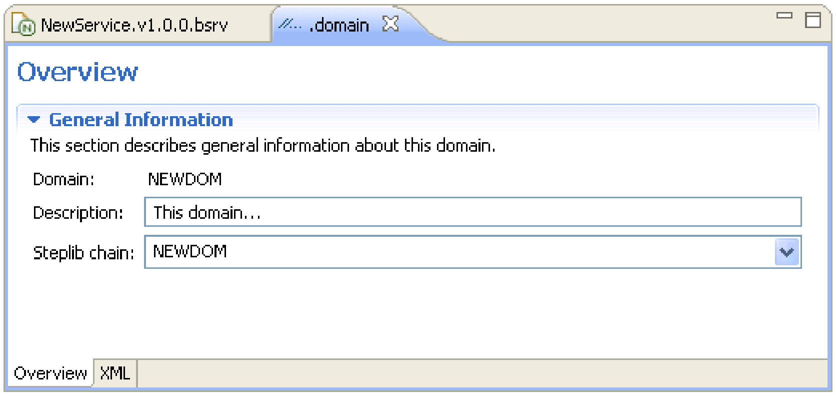
Define the Domain Information

To define the domain information:

1 Expand the domain node for the business service in the **Navigator** view.

2 Open the *.domain* file for the business service.

The domain editor is displayed. For example:



This section describes general information about this domain.

Domain: NEWDOM

Description: This domain...

Steplib chain: NEWDOM

Using this editor, you can:

Task	Procedure
Provide a description of the domain.	Type a brief description in **Description**.
Change the default name of the steplib chain (list of libraries that includes the step library containing the subprogram that implements the service method).	Type or select the name of the steplib chain in **Steplib**.

3 Save the domain.

Notes:

1. You can upload the domain file to the server using standard NaturalONE functionality.
2. For more information on domains, see *Defining Steplibs and Domains*, *Natural Business Services Administration*.

Define the Steplib Chain Information

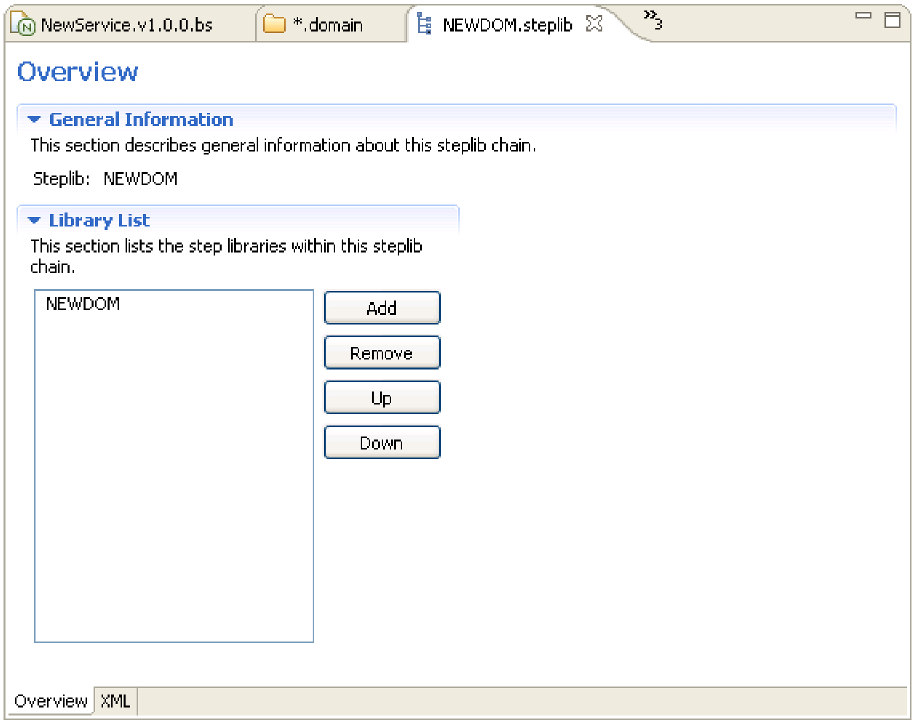
This section describes how to define the list of step libraries in the steplib chain used for the business service.

To define the steplib chain information:

1 Expand the step library node in the **Navigator** view.

2 Open the *SteplibName*.steplib file.

The steplib editor is displayed, showing the name of the steplib chain in **Steplib**. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWDOM

Library List

This section lists the step libraries within this steplib chain.

NEWDOM

The first step library in the **Library List** section has the same name as the steplib chain. You can use this step library as a starting point when defining the steplib chain.

Tip: The settings defined in **Properties > Natural > Steplibs** on the context menu for the project can be used to populate the **Library List**.

Using this editor, you can:

Task	Procedure
Add a step library to the steplib chain.	See *Add a Step Library to the Steplib Chain*. **Note:** For information on creating a new step library, see *Create a Step Library*.
Remove a step library from the steplib chain.	Select the library and then select **Remove**.
Reorder the list of step libraries within the steplib chain.	Select **Up** and **Down** to reorder the list.

3 Save the steplib.

Notes:

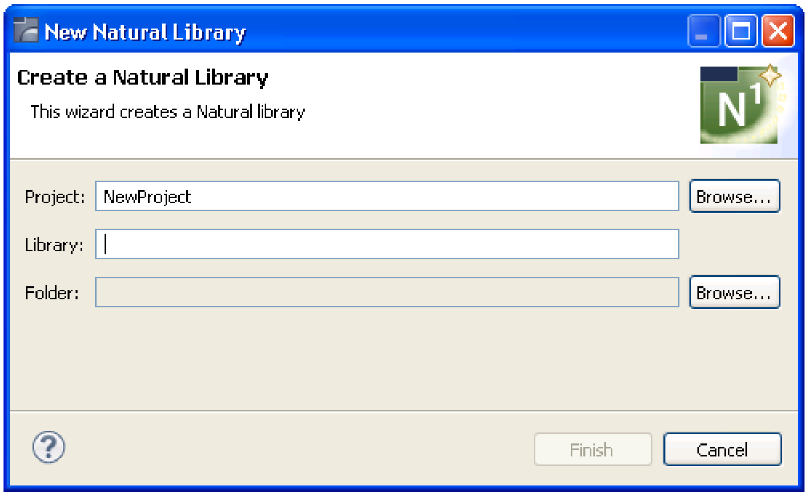
1. You can upload the steplib file to the server using standard NaturalONE functionality.
2. For more information on steplib chains, see *Defining Steplibs and Domains*, *Natural Business Services Administration*.

Create a Step Library

To create a step library:

1 Select **New > Natural Library** on the **File** menu.

The **Create a Natural Library** panel is displayed. For example:



Project: NewProject

Library:

Folder:

2 Select the name of the NaturalONE project in **Project**.

3 Type the name of the step library in **Library**.

4 Select **Finish**.

The library is created locally and displayed in the **Navigator** view.

Note: You can upload the library to the server using standard NaturalONE functionality.

Add a Step Library to the Steplib Chain

▶ To add a step library to the steplib chain:

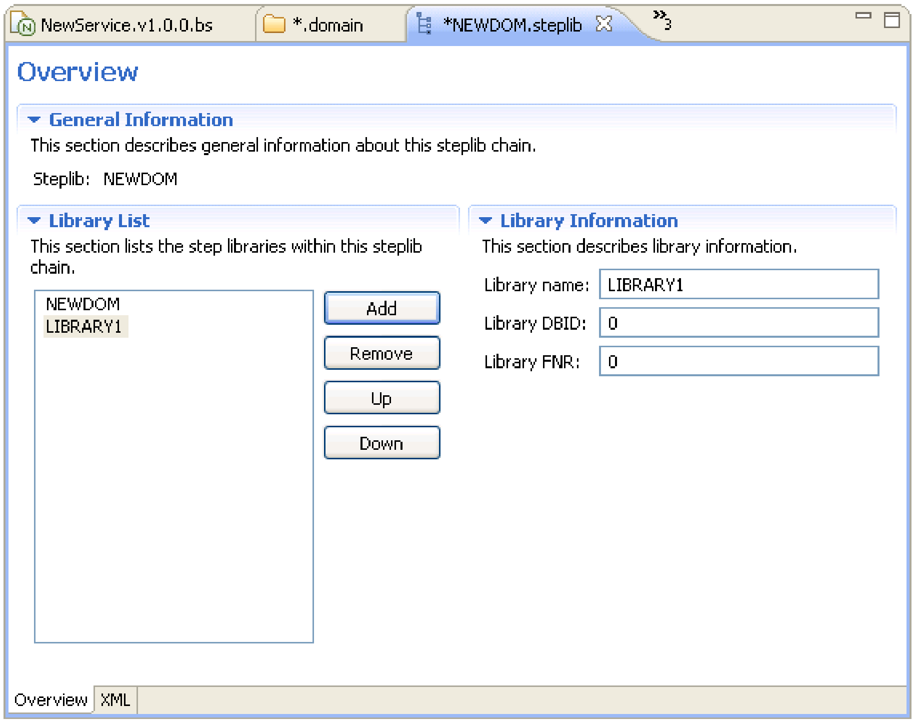
1 Expand the step library node in the **Navigator** view.

2 Open the `SteplibName`.steplib file.

The steplib editor is displayed, showing the name of the steplib chain in **Steplib**.

3 Select **Add**.

The Library Information section is displayed. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEWDOM

Library List

This section lists the step libraries within this steplib chain.

NEWDOM
LIBRARY1

Library Information

This section describes library information.

Library name: LIBRARY1

Library DBID: 0

Library FNR: 0

By default, LIBRARY1 is displayed in **Library name**.

Note: The DBID and FNR values are not required; they are only used in advanced configurations when uploading to the server.

4 Type the name of the new library over the default name in **Library name**.

5 Save the steplib file.

Note: For information on removing a step library from the steplib chain or reordering the list of step libraries, see *Define the Steplib Chain Information*.

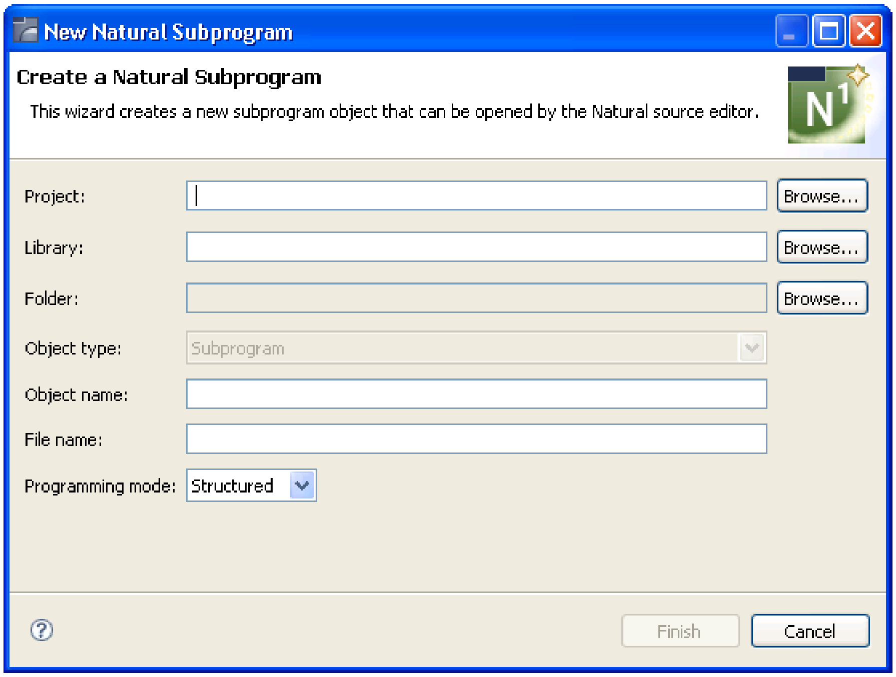
Create the Subprogram to Implement a Method

This subprogram will implement the logic for your new business service.

To create the subprogram:

1 Select **New > Subprogram** on the **File** menu.

The **Create a Natural Subprogram** panel is displayed. For example:



New Natural Subprogram

Create a Natural Subprogram

This wizard creates a new subprogram object that can be opened by the Natural source editor.

Project:

Library:

Folder:

Object type: Subprogram

Object name:

File name:

Programming mode: Structured

2 Select the name of the project in **Project**.

3 Select the name of the library in which to store the subprogram in **Library**.

4 Select the local folder in which to store the subprogram in **Folder**.

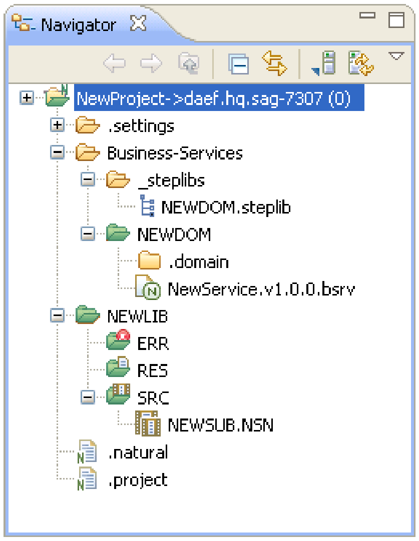
5 Type the name of the subprogram in **Object name**.

6 Type the name of the file in **File name**.

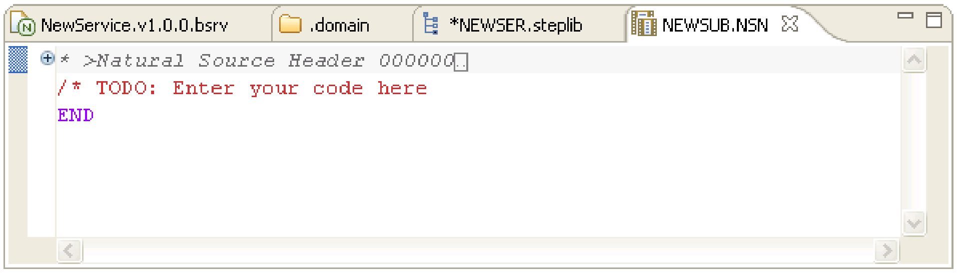
Note: By default, **Structured** is selected in **Programming mode**. If the subprogram will use reporting mode, select **Reporting**.

7 Select **Finish** to create the subprogram.

The new subprogram is listed in the **Navigator** view. For example:



The subprogram is also displayed in the Natural editor. For example:



```
* >Natural Source Header 000000
/* TODO: Enter your code here
END
```

8 Add logic to the subprogram.

9 Save the subprogram.

The subprogram is saved locally.

Note: You can upload the subprogram to the server using standard NaturalONE functionality.

You can now add a method to the business service.

Add a Method

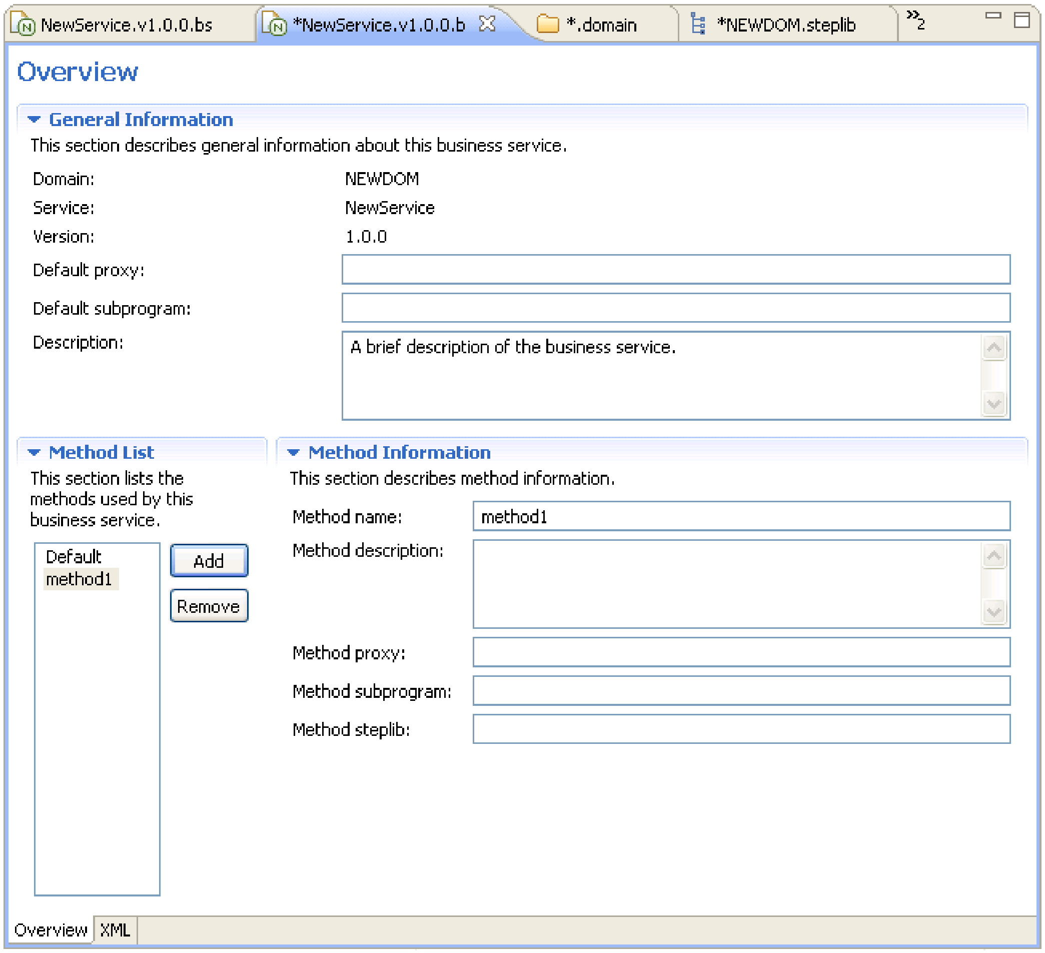
To add a method to the business service:

1 Open the *NewService*.**bsrv** file in the **Navigator** view.

The business service is displayed in the editor view.

2 Select **Add**.

The **Method Information** section is displayed. For example:



Overview

General Information

This section describes general information about this business service.

Domain: NEWDOM

Service: NewService

Version: 1.0.0

Default proxy:

Default subprogram:

Description: A brief description of the business service.

Method List

This section lists the methods used by this business service.

Default
method1

Add

Remove

Method Information

This section describes method information.

Method name: method1

Method description:

Method proxy:

Method subprogram:

Method steplib:

By default, method1 is displayed in **Method name**.

3 Type the name of the new method over the default name in **Method name**.

4 Type a brief description of the method in **Method description**.

5 Type the name of the subprogram that implements this method in **Method subprogram**.

Note: You do not need to specify a method step library unless the default step library associated with the business service/domain is not sufficient.

Test the Business Service and Subprograms

For information on testing business services and subprograms, see *Application Testing*.

Use the Dependencies View

When a business services resource (for example, a business service, domain, steplib, business service unit test) is open in the editor view, the **Dependencies** view displays dependencies between that resource and other business services resources and/or Natural resources. This section describes the nodes contributed to the view by business service resources. The following topics are covered:

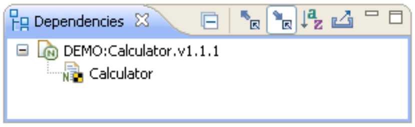
- Business Service Resources
- Natural Subprogram Resources

Notes:

1. Select to sort the resources alphabetically.
2. Select to export a textual representation of the visible nodes in the view to a file.
3. When a supporting resource cannot be found locally using the project steplib chain and project references, "<Unknown>" is displayed with the name of the resource. If the unknown module(s) is not shipped with the Construct runtime project, either manually download it from the server or create it locally. If the module(s) is shipped with the Construct runtime project, add the project. For information, see the *NaturalONE Code Generation* guide.
4. For more information about the **Dependencies** view, see the description of the source editor in *Using NaturalONE*.

Business Service Resources

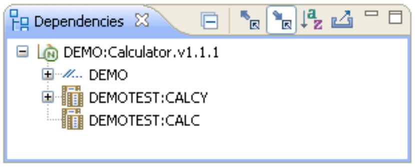
When a business service is open in the editor view, the root node displays the name of the business service, as well as the name of the domain in which it is located. In caller mode ( ), the child nodes display the name of each business service resource that depends on this business service. In the following example, the business service resource that depends on the Calculator.v1.1.1 business service in the DEMO domain is a business service unit test named Calculator:



Dependencies

- DEMO:Calculator.v1.1.1
  - Calculator

In callee mode (), the child nodes display the names of the business service resources that the parent node depends on. In the following example, the Calculator.v.1.1.1 business service in the DEMO domain depends on a domain named DEMO and two subprograms named CALC and CALCY in the DEMOTEST library:

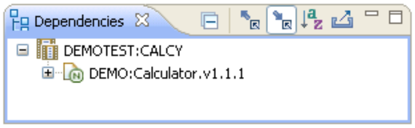


Dependencies

- DEMO:Calculator.v1.1.1
  - DEMO
  - DEMOTEST:CALCY
  - DEMOTEST:CALC

Natural Subprogram Resources

When a Natural subprogram is open in the editor view, the root node displays the name of the subprogram, as well as the name of the library in which it is located. In caller mode (), a business service child node will be displayed for each business service that depends on this subprogram. In the following example, the Calculator.v1.1.1 business service in the DEMO domain depends on the CALCY subprogram in the DEMOTEST library:



Dependencies

- DEMOTEST:CALCY
  - DEMO:Calculator.v1.1.1

In callee mode (), no child nodes are displayed because subprograms do not depend on any business service resources.

Add Business Service Definitions to CentraSite

You can register business services and their corresponding metadata with CentraSite using the **Add to CentraSite** wizard.

Note: This wizard is an optional feature for NaturalONE and is only available when the CentraSite plug-ins are installed. To install CentraSite, refer to the CentraSite documentation.

This section covers the following topics:

- Import the Business Service Asset Types

- Add Business Service Metadata to CentraSite
- Verify Metadata was Added to CentraSite

Import the Business Service Asset Types

Before you can add business service definitions to CentraSite, you must import the business service asset types to the CentraSite server.

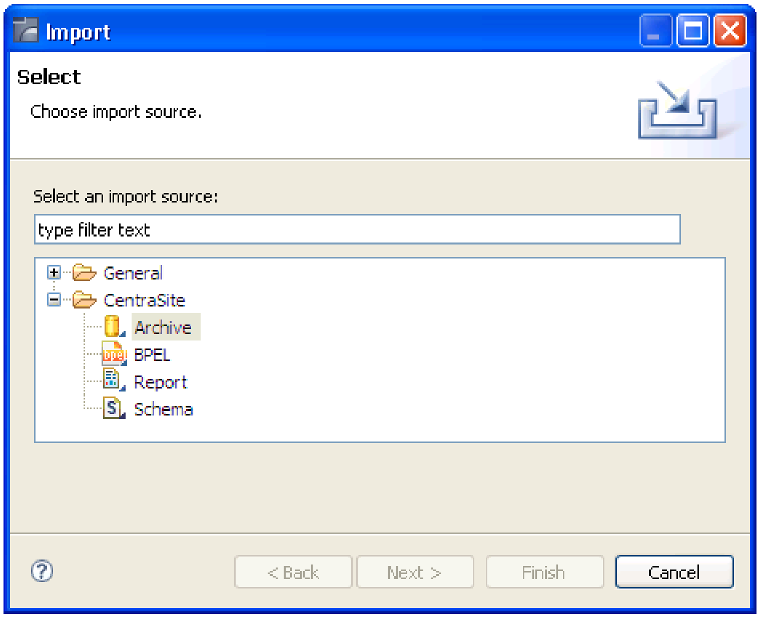
To import the business service asset types:

1 Select **Import** on the **File** menu.

The **Import** window is displayed.

2 Select **CentraSite > Archive**.

For example:



3 Search for and select the *NaturalONE install folder\eclipse\v34\plugins\com.softwareag.naturalone.gen.nbs.eclipse.version\res\com.softwareag.naturalone.gen.nbs.centrasite_asset_types.zip* file.

Note: The version number above may change.

4 Select **Finish** to import the asset types file.

Note: For more information on importing a file, see the CentraSite documentation.

Add Business Service Metadata to CentraSite

This section describes how to add the business service metadata to CentraSite.

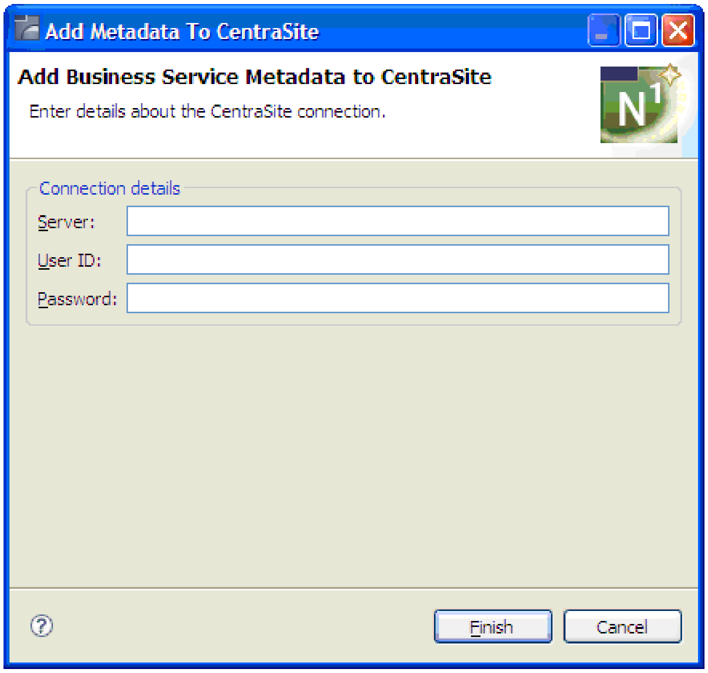
To add business service metadata to CentraSite:

1 Open the context menu for the business service in the **Navigator** view.

You can add the metadata for all business services defined in a NaturalONE project or for any folder or object within the project, depending on which business services node is selected in the **Navigator** view. For example, if you select the **Business-Services** root node, the metadata for everything defined in the project will be added to CentraSite. If you select an object or folder within the **Business-Services** root node, only this object or folder (and any items within the folder) will be added.

2 Select **Business-Services > Add to CentraSite**.

The **Add Business Service Metadata to CentraSite** panel is displayed. For example:



Add Metadata To CentraSite

Add Business Service Metadata to CentraSite

Enter details about the CentraSite connection.

Connection details

Server:

User ID:

Password:

3 Type the CentraSite connection path in **Server**.

4 Type the user credentials for CentraSite in **User ID** and **Password**.

Tip: You can set up the default **Server** and **User ID** values in the **Preferences** window for CentraSite. For information, see *Set CentraSite Preferences*.

5 Select **Finish**.

The business service definition will be added to CentraSite and the wizard will close.

Note: If there are any problems, check the **Error** view for information.

Verify Metadata was Added to CentraSite

To confirm that metadata has been added to CentraSite:

1 Select **Open Perspective > Other** on the **Window** menu.

The **Open Perspective** window is displayed.

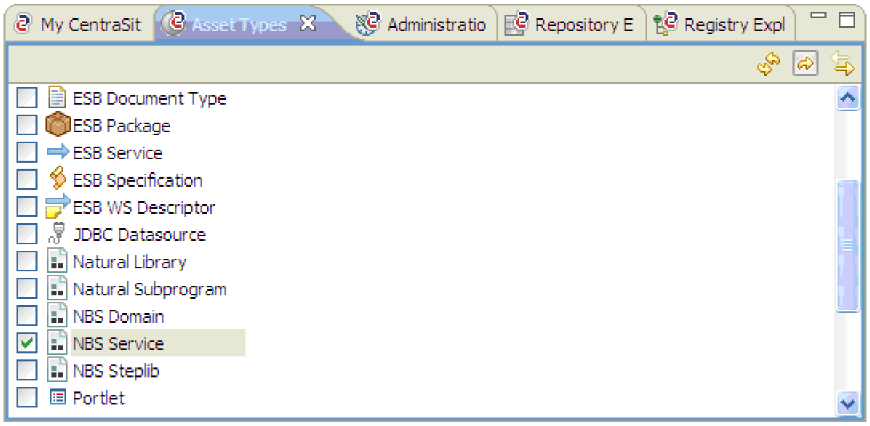
2 Select **CentraSite > Search and Browse**.

3 Select the **Asset Types** view.

If necessary, select ⟳ to refresh the view.

4 Select an NBS asset type.

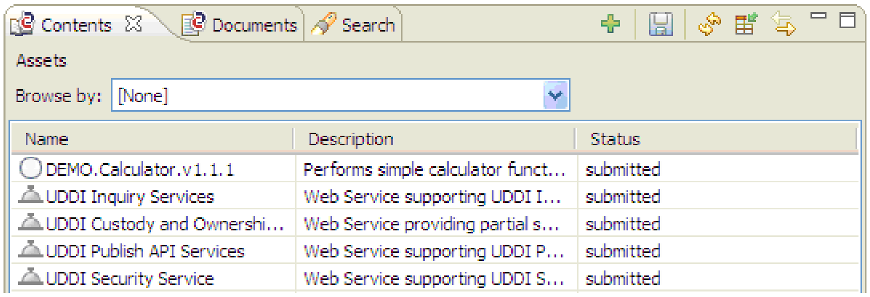
For example, select **NBS Service**:



☐ ESB Document Type
☐ ESB Package
☐ ESB Service
☐ ESB Specification
☐ ESB WS Descriptor
☐ JDBC Datasource
☐ Natural Library
☐ Natural Subprogram
☐ NBS Domain
☑ NBS Service
☐ NBS Steplib
☐ Portlet

5 Select ➩ to update the list in the **Contents** view.

For example:



Assets

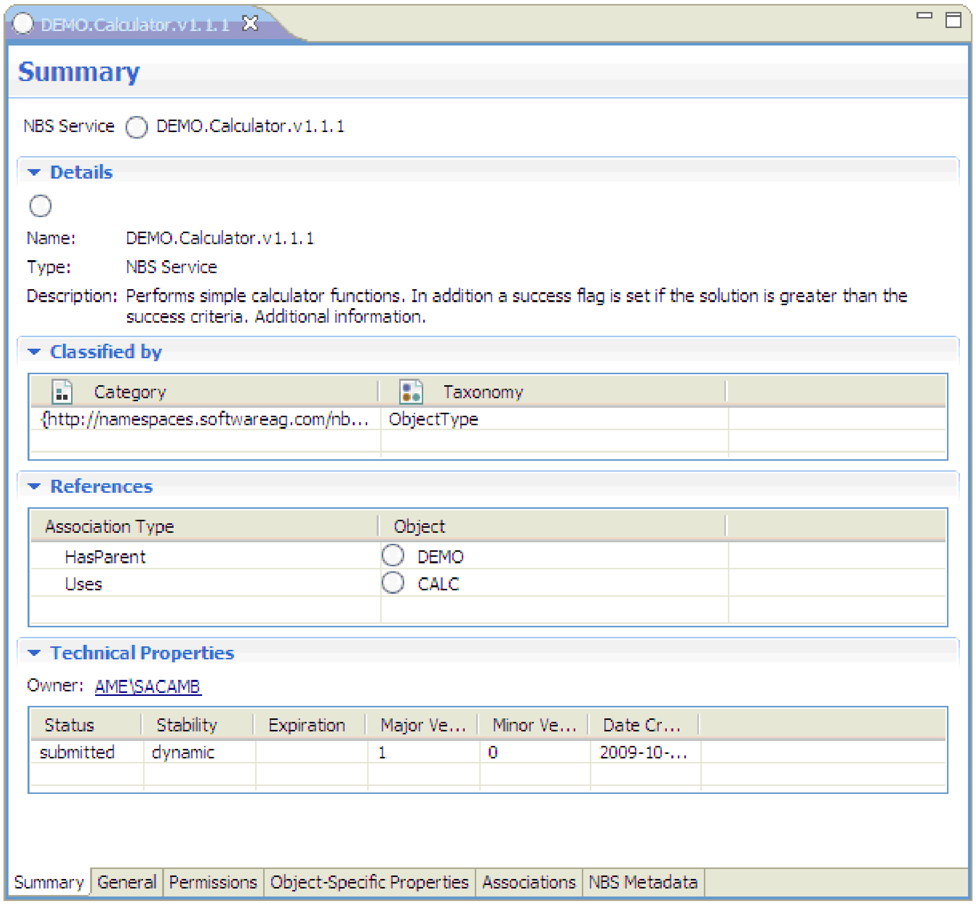
Browse by: [None]

Name	Description	Status
DEMO.Calculator.v1.1.1	Performs simple calculator funct...	submitted
UDDI Inquiry Services	Web Service supporting UDDI I...	submitted
UDDI Custody and Ownershi...	Web Service providing partial s...	submitted
UDDI Publish API Services	Web Service supporting UDDI P...	submitted
UDDI Security Service	Web Service supporting UDDI S...	submitted

Tip: To display a graphical impact analysis, open the context menu for an object in the **Contents** view.

6 Select the business service that you added to CentraSite.

The **Summary** page is displayed. For example:



Summary

NBS Service ○ DEMO.Calculator.v1.1.1

Details

Name: DEMO.Calculator.v1.1.1

Type: NBS Service

Description: Performs simple calculator functions. In addition a success flag is set if the solution is greater than the success criteria. Additional information.

Classified by

Category	Taxonomy
{http://namespaces.softwareag.com/nb...	ObjectType

References

Association Type	Object
HasParent	○ DEMO
Uses	○ CALC

Technical Properties

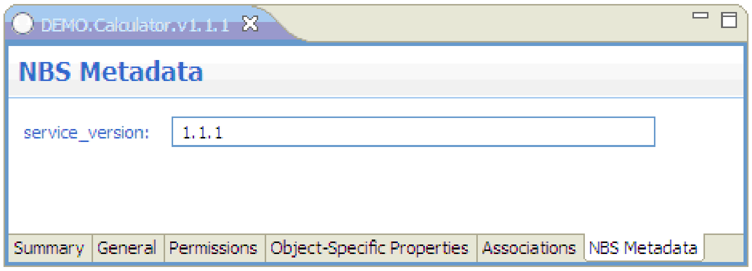
Owner: AME\SACAMB

Status	Stability	Expiration	Major Ve...	Minor Ve...	Date Cr...
submitted	dynamic		1	0	2009-10-...

7 Verify the associations in the **References** section.

8 Select the **NBS Metadata** tab.

Additional information about the service is displayed. For example:



service_version: 1.1.1

18 Generating a New Domain

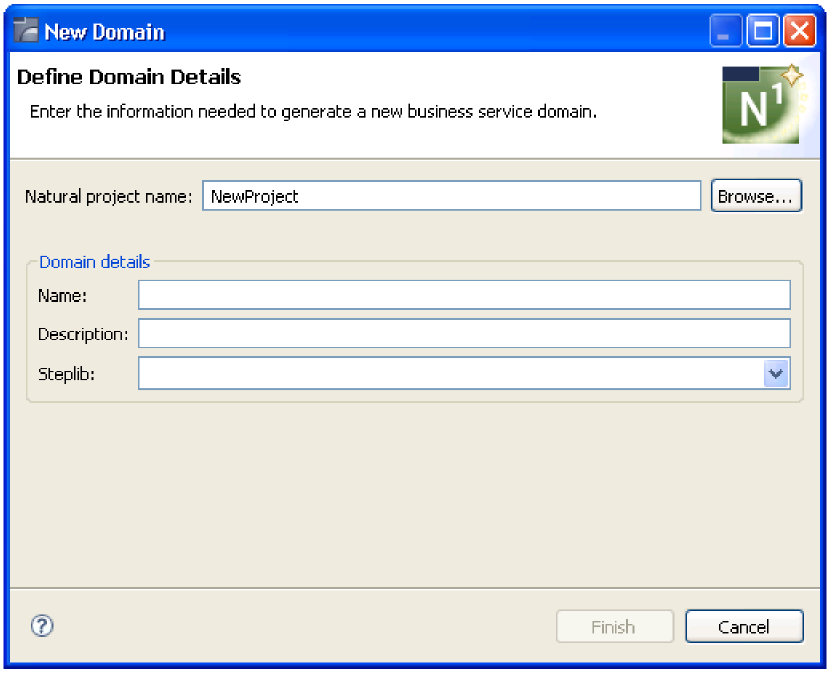
This section describes how to generate a new domain into a NaturalONE project. The domain file will be created locally.

Note: You can upload the file to the server using standard NaturalONE functionality.

▶ To generate a new domain:

1. Open the context menu for the project.
2. Select **Business-Services > New Domain**.

   The **Define Domain Details** panel is displayed, showing the name of the project. For example:



New Domain

Define Domain Details

Enter the information needed to generate a new business service domain.

Natural project name: NewProject

Note: If this is not the project into which you want to generate the domain, select **Browse** to choose a different NaturalONE project.

3 Type the name of the domain in **Name**.

Note: The wizard will check the server connection to verify whether the domain currently exists. If it does, a message is displayed. You can either overwrite the existing domain or enter a new name.

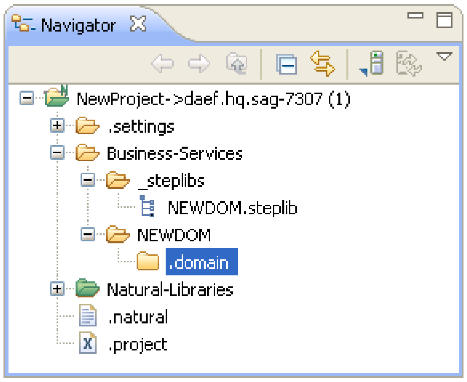
4 Type a brief description of the domain in **Description**.

5 Select the steplib in **Steplib**.

By default, the steplib name is the same as the domain name.

6 Select **Finish** to generate the domain file (and possibly the associated steplib file).

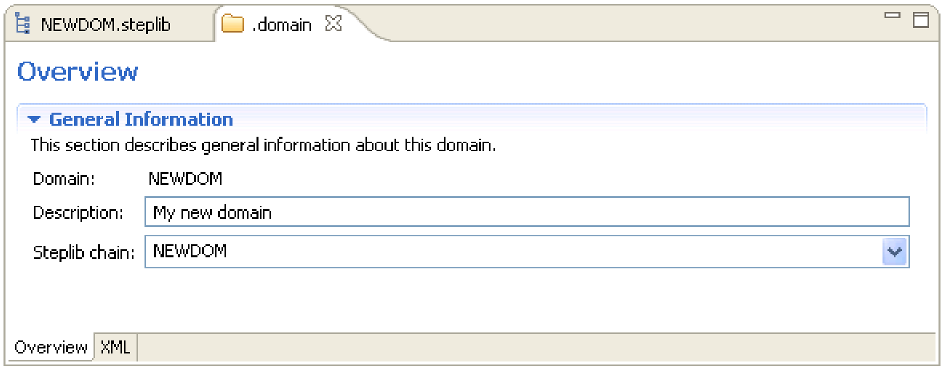
The generated items are added to your NaturalONE project and the new domain is now listed in the **Navigator** view. For example:



Note: The generated steplib file is also displayed (see NEWDOM.steplib in the example above).

7 Open the domain to display details in the editor view.

For example:



This section describes general information about this domain.

Domain: NEWDOM

Description: My new domain

Steplib chain: NEWDOM

8 Save the domain file.

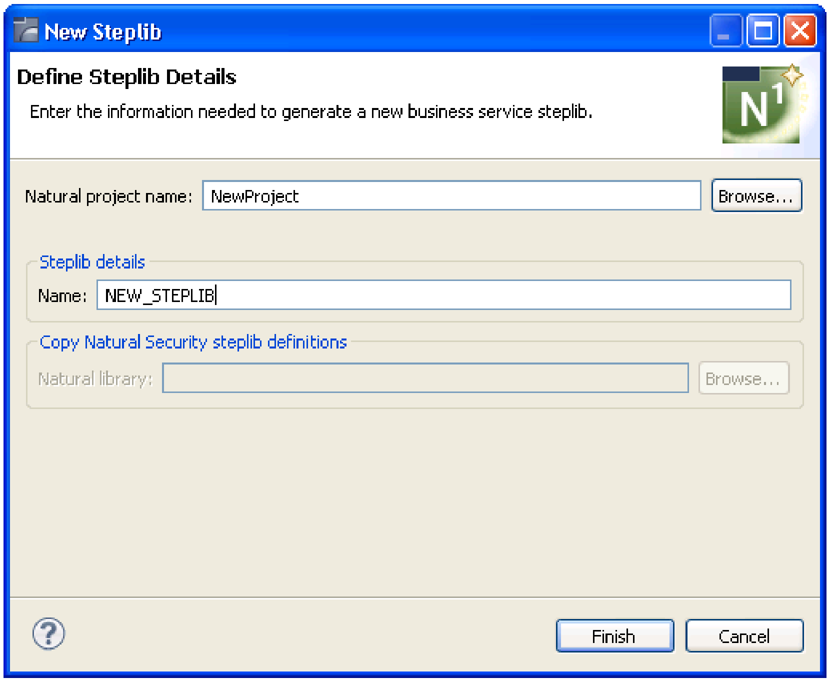
19 Generating a New Steplib

This section describes how to generate a new steplib into a NaturalONE project. The steplib file will be created locally.

To generate a new steplib:

1. Open the context menu for the project.
2. Select **Business-Services > New Steplib**.

   The **Define Steplib Details** panel is displayed, showing the name of the project. For example:



New Steplib

Define Steplib Details

Enter the information needed to generate a new business service steplib.

Natural project name: NewProject

Steplib details

Name: NEW_STEPLIB

Copy Natural Security steplib definitions

Natural library:

3 Type the name of the steplib in **Name**.

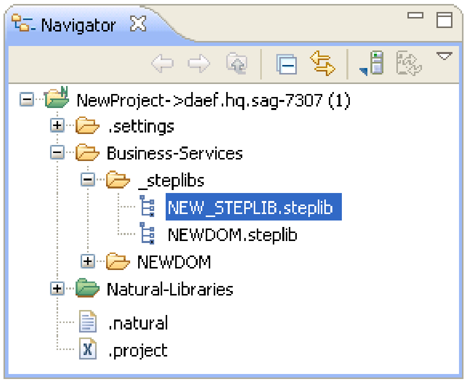
Note: If the steplib currently exists, a message will be displayed.

Optionally, you can:

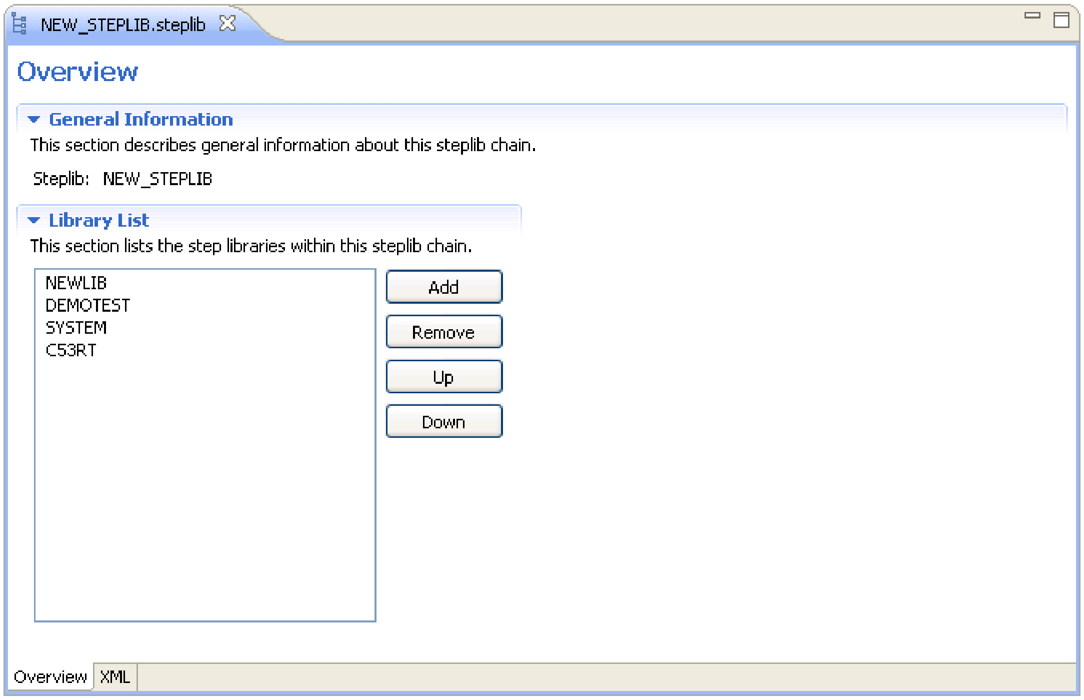
Task	Procedure
Select another NaturalONE project into which to generate the steplib.	Type the name of the project in **Project** or select **Browse** to display a window listing the existing projects for selection. The project must currently exist.
Retrieve the steplib definitions for a secured library.	Type the name of the library in **Natural library** or select **Browse** to choose the library from a list of available libraries. **Note:** 1. If you specify a secure project and a library, the steplib definitions in the specified library will be copied; if you specify a non-secure project, the steplib definitions associated with the project will be copied. 2. This option is only available when **Natural Security** is installed.

4 Select **Finish** to generate the steplib file.

The generated items are added to your NaturalONE project and the new steplib is now listed in the **Navigator** view. For example:



Open the steplib file to display details in the editor view. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: NEW_STEPLIB

Library List

This section lists the step libraries within this steplib chain.

NEWLIB
DEMOTEST
SYSTEM
C53RT

Note: By default, the settings defined in **Properties > Natural > Steplibs** on the context menu for the project are used as the default steplib chain.

5 Add step libraries to the steplib chain in **Library List**.

For information, see *Add a Step Library to the Steplib Chain*.

6 Save the steplib file.

Note: You can upload the file to the server using standard NaturalONE functionality.

20 Downloading Business Services from the Server

This section describes how to download business services data from a Natural Business Services installation on the server to the local environment. This procedure is performed in two parts: first the **Natural Server** view is mapped to an existing business service installation on the server and then the services are downloaded from **Natural Server** to the **Navigator** view.

This section covers the following topics:

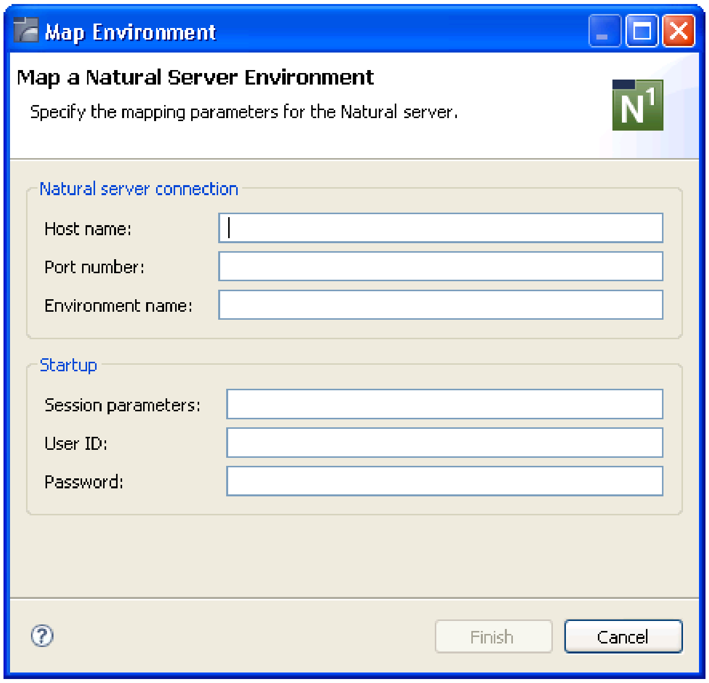
Note: All business service resources are stored in a local NaturalONE project. If you do not want to use an existing project to store the downloaded resources, you can create a new project. For information, see the NaturalONE documentation.

Map to an Existing Natural Business Services Installation

To map to an existing Natural Business Services installation:

1 Select on the toolbar in the **Natural Server** view.

The **Map Environment** panel is displayed. For example:



Map Environment

Map a Natural Server Environment

Specify the mapping parameters for the Natural server.

2 Type the name of the host in **Host name**.

3 Type the port number in **Port number**.

The name of the environment is derived from the host and port values and displayed in **Environment name** when you select the field.

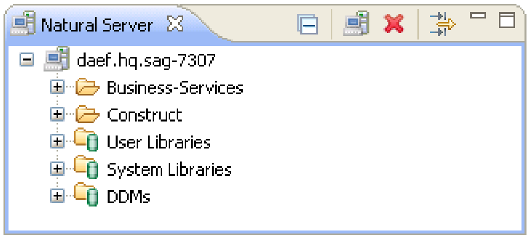
4 Type a Natural profile name in **Session parameters**.

5 Type the user credentials for the server in **User ID** and **Password**.

Note: These credentials must be defined to Natural Business Services security and have the correct permissions.

6 Select **Finish**.

The connection is displayed in the **Natural Server** view. For example:



Note: For more information about the **Natural Server** view, see the NaturalONE documentation.

Download Business Services Resources to a Local Project

To download business services to your local project:

1 Open the context menu for the **Business-Services** root node.

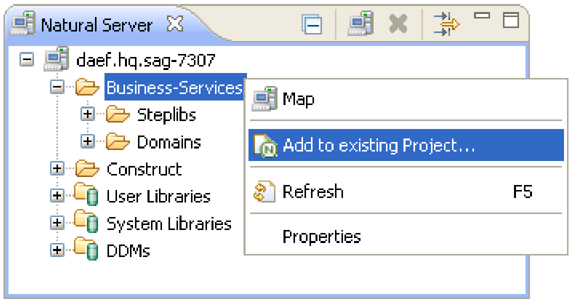
Or:

Expand the root node and select one or more steplib and/or domain nodes or files using standard selection techniques.

Note: Children of the selected nodes are automatically included in the download (for example, selecting the **Domains** node will download all domains from the server).

2 Select **Add to existing Project**.

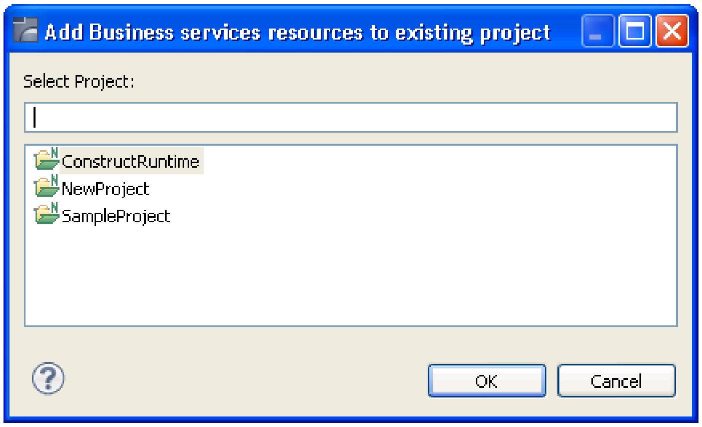
For example:



A list of available projects is displayed.

3 Select the project into which you want to download the business service data.

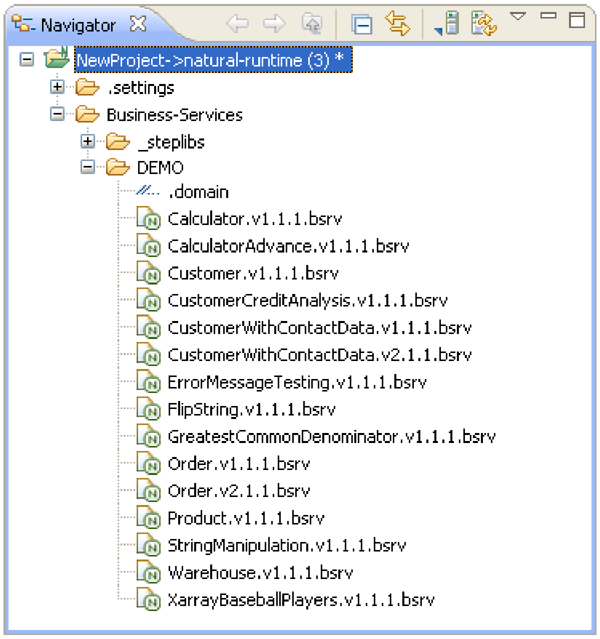
For example:



4 Select **OK**.

A progress window is displayed as the business service data is downloaded from the **Natural Server** view to the local project in the **Navigator** view.

The definitions for all selected services, domains, and steplibs are added to the specified project. Any supporting business service resource that was not selected will be automatically downloaded depending on the user preference value (for example, the supporting steplib for a selected domain will be downloaded if the user preference value is set to true). To view the downloaded resources, expand the *Project name* > **Business-Services** node in the **Navigator** view. For example:



- NewProject->natural-runtime (3) *
  - .settings
  - Business-Services
    - _steplibs
    - DEMO
      - .domain
      - Calculator.v1.1.1.bsrv
      - CalculatorAdvance.v1.1.1.bsrv
      - Customer.v1.1.1.bsrv
      - CustomerCreditAnalysis.v1.1.1.bsrv
      - CustomerWithContactData.v1.1.1.bsrv
      - CustomerWithContactData.v2.1.1.bsrv
      - ErrorMessageTesting.v1.1.1.bsrv
      - FlipString.v1.1.1.bsrv
      - GreatestCommonDenominator.v1.1.1.bsrv
      - Order.v1.1.1.bsrv
      - Order.v2.1.1.bsrv
      - Product.v1.1.1.bsrv
      - StringManipulation.v1.1.1.bsrv
      - Warehouse.v1.1.1.bsrv
      - XarrayBaseballPlayers.v1.1.1.bsrv

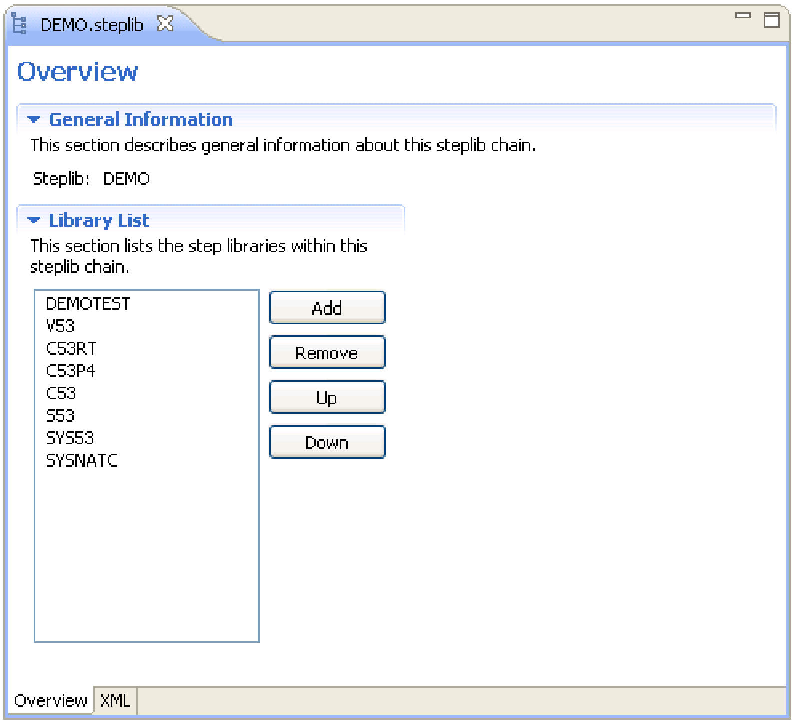
Download Additional Libraries (Optional)

This section describes how to determine whether any libraries are missing and then download the additional libraries from the server.

To download additional libraries from the server:

1 Open the **_steplibs** > *StepLibName***.steplib** file in the **Navigator** view.

The steplib editor is displayed. For example:



This section describes general information about this steplib chain.

Steplib: DEMO

This section lists the step libraries within this steplib chain.

DEMOTEST
V53
C53RT
C53P4
C53
S53
SYS53
SYSNATC

Note: For more information about the steplib editor, see *Define the Steplib Chain Information*.

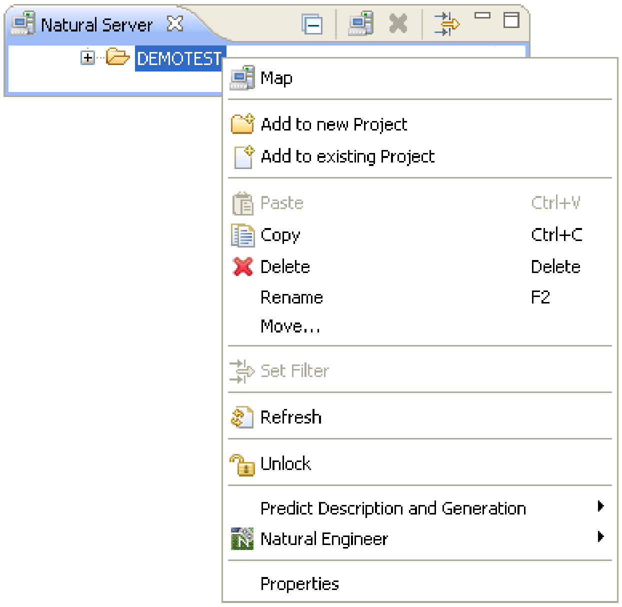
2 Review the **Library List** for any step libraries that are not available locally (i.e., not listed in the **Navigator** view).

In this example, the DEMOTEST step library is not available.

3 Open the context menu for the missing library in the **Natural Server** view.

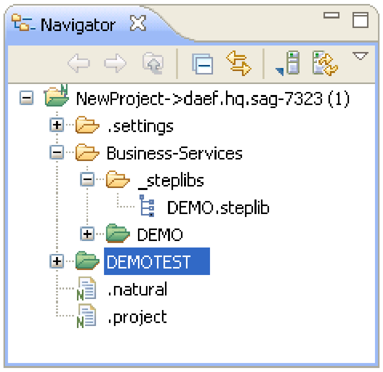
4 Select **Add to existing Project**.

For example:



Note: You can also select **Add to new Project** to create a new NaturalONE project into which you can download the library. For information about creating a project, see the NaturalONE documentation.

The selected library is downloaded from the server to the specified NaturalONE project and is now available in the **Navigator** view. For example:



Perform Standard Actions on Business Service Resources

You can use the **Natural Server** view to copy/paste, delete, or move business service resources on the server. The action will be performed in the mapped environment for the selected node(s).

Note: Although you can select resources from more than one **Business-Services** node, you cannot select nodes from different mapped environments.

This section covers the following topics:

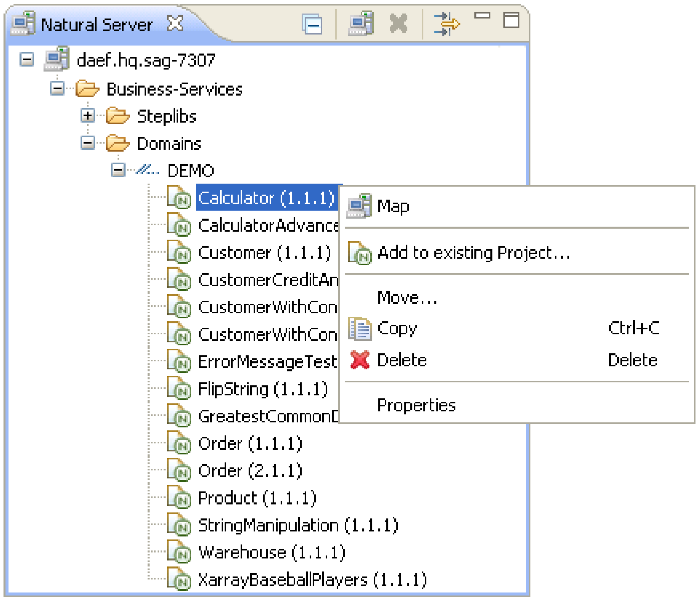
- Perform Actions on a Business Service
- Perform Actions on a Domain
- Perform Actions on a Steplib

Perform Actions on a Business Service

To perform actions on one or more business services:

1 Open the context menu for the business service(s) in the **Natural Server** view.

For example:



2 Select one of the actions listed.

The available actions are:

Action	Description
Move	Removes the selected business service(s) from the current domain and adds the service(s) to a target domain in the mapped environment for the selected node. For information, see *Move a Business Service*.
Copy	Copies the selected business service(s) to the clipboard in anticipation of a Paste action. For information, see *Copy a Business Service*.
Delete	Removes the selected business service(s) from the current domain. For information, see *Delete a Business Service*.

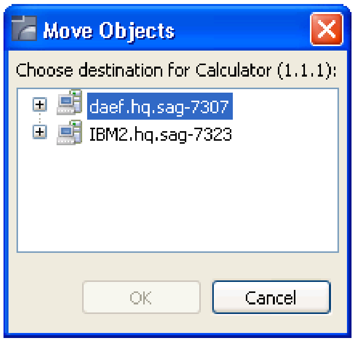
Move a Business Service

This section describes how to move one or more business services from the current domain to a target domain in the mapped environment for a selected node(s).

To move one or more business services:

1. Open the context menu for the business service(s) in the **Natural Server** view.
2. Select **Move**.

   The **Move Objects** window is displayed. For example:

   This window lists the connection nodes for the available mapped environments.
3. Expand the connection node for the environment into which you want to move the service(s).
4. Select the **Domains** root node into which you want to move the service(s).
5. Select **OK**.

   A progress window is displayed while the service(s) is removed from the previous **Domains** node and copied to the selected **Domains** node.

Copy a Business Service

This section describes how to copy one or more business services to the clipboard and then paste the service(s) into a target domain.

▶ To copy one or more business services:

1. Open the context menu for the business service(s) in the **Natural Server** view.
2. Select **Copy**.
3. Open the context menu for the **Domains** root node into which you want to copy the service(s).
4. Select **Paste**.

   The service(s) is copied to the selected **Domains** root node.

Delete a Business Service

This section describes how to remove one or more business services from the current domain.

▶ To delete one or more business services:

1. Open the context menu for the business service(s) in the **Natural Server** view.
2. Select **Delete**.

   A confirmation window is displayed to confirm the action.
3. Select **Yes**.

   The service(s) is removed from the **Domains** root node in the current mapped environment.

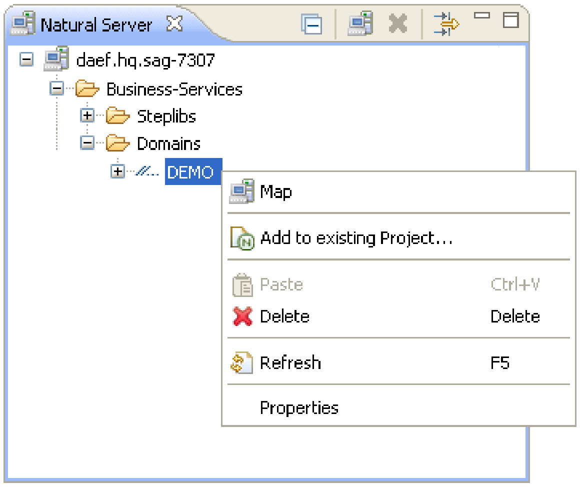
Perform Actions on a Domain

Typically, the only action available for a domain is Delete, which removes one or more domains from the current mapped environment.

▶ To remove one or more domains from the server:

1. Open the context menu for the domain in the **Natural Server** view.

   For example:



Note: The Paste action is only available when the clipboard contains one or more business services. For information, see *Copy a Business Service*.

2 Select **Delete**.

A confirmation window is displayed to confirm the action.

3 Select **Yes**.

The selected domain(s) is removed from the **Domains** root node in the current mapped environment.

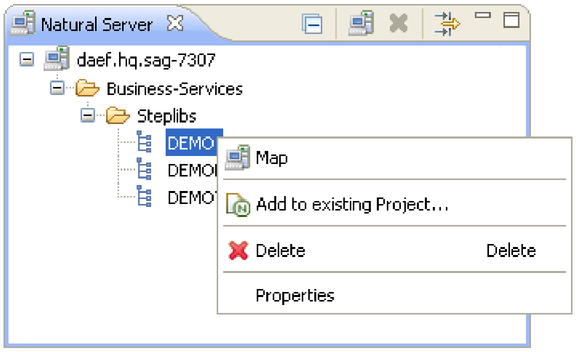
Perform Actions on a Steplib

The only action available for a steplib is Delete, which removes one or more steplibs from the current mapped environment.

To remove one or more steplibs from the server:

1 Open the context menu for the steplib in the **Natural Server** view.

For example:



2 Select **Delete**.

A confirmation window is displayed to confirm the action.

3 Select **Yes**.

The steplib(s) is removed from the **Steplibs** root node in the current mapped environment.

21 Editing Information about an Existing Business Service

This section describes how to edit the information for an existing business service. The business service editors allow you to see and edit all the information for a business service.

Note: You cannot change the key fields, such as the file or domain names.

This section covers following topics:

Edit the Business Service Definition

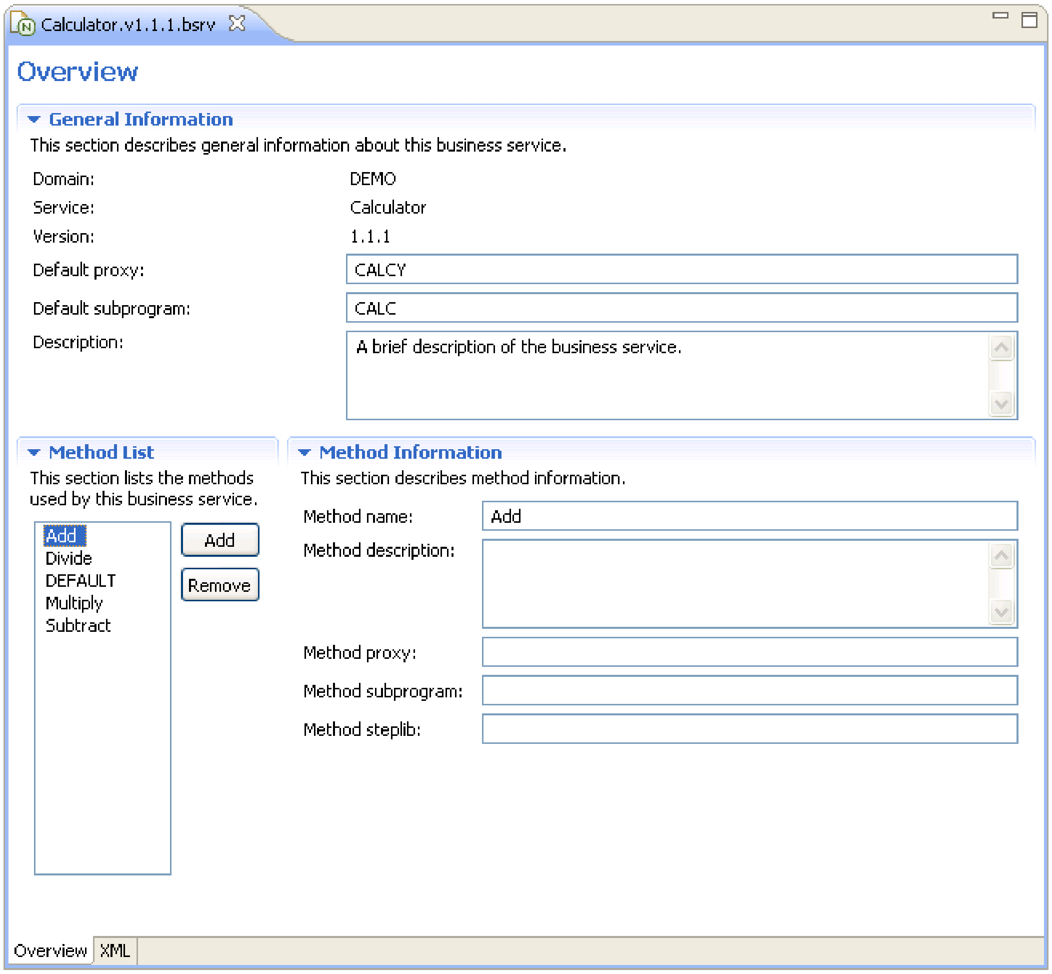
▶ **To edit the business service definition:**

1 Open the *`ExistingService`***.bsrv** file in the **Navigator** view.

The business service editor is displayed.

2 Select a method in the **Method List** section.

The **Method Information** section is displayed. For example:



Overview

General Information

This section describes general information about this business service.

Domain: DEMO

Service: Calculator

Version: 1.1.1

Default proxy: CALCY

Default subprogram: CALC

Description: A brief description of the business service.

Method List

This section lists the methods used by this business service.

Add
Divide
DEFAULT
Multiply
Subtract

Method Information

This section describes method information.

Method name: Add

Method description:

Method proxy:

Method subprogram:

Method steplib:

3 Edit information about the service.

Using the editor, you can:

Task	Procedure
Change the name of the subprogram proxy used for this service.	Type the name of the proxy in **Default proxy**.
Change the name of the subprogram that implements the service methods.	Type the name of the subprogram in **Default subprogram**.
Change the description of the business service.	Type the description in **Description**.
Add a service method.	See *Add a Method*.
Remove a service method.	Select the method in the **Methods List** section and select **Remove**.
Change details about a method.	Select the method in the **Methods List** section and change the details in the **Method Information** section.

4 Save the business service.

Note: You can upload the business service file to the server using standard NaturalONE functionality.

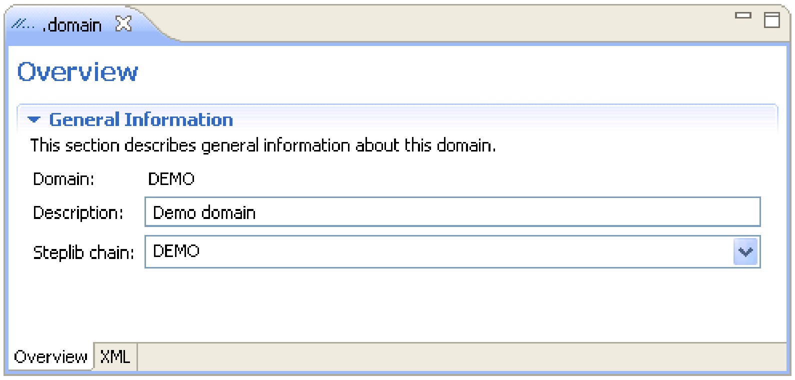
Edit the Domain Definition

To edit the domain definition:

1 Expand the domain node in the **Navigator** view.

2 Open the *.domain* file for the business service.

The domain editor is displayed. For example:



This section describes general information about this domain.

Domain: DEMO

Description: Demo domain

Steplib chain: DEMO

3 Edit information about the domain.

Using this editor, you can:

Task	Procedure
Change or provide a description of the domain.	Type a description of the domain in **Description**.
Change the default name of the steplib chain (list of libraries that includes the step library containing the subprogram that implements the service method).	Type or select the name of the steplib chain in **Steplib**.

4 Save the domain.

Notes:

1. You can upload the domain file to the server using standard NaturalONE functionality.
2. For more information on domains, see *Defining Steplibs and Domains*, *Natural Business Services Administration*.

Edit the Steplib Definition

To edit the steplib definition:

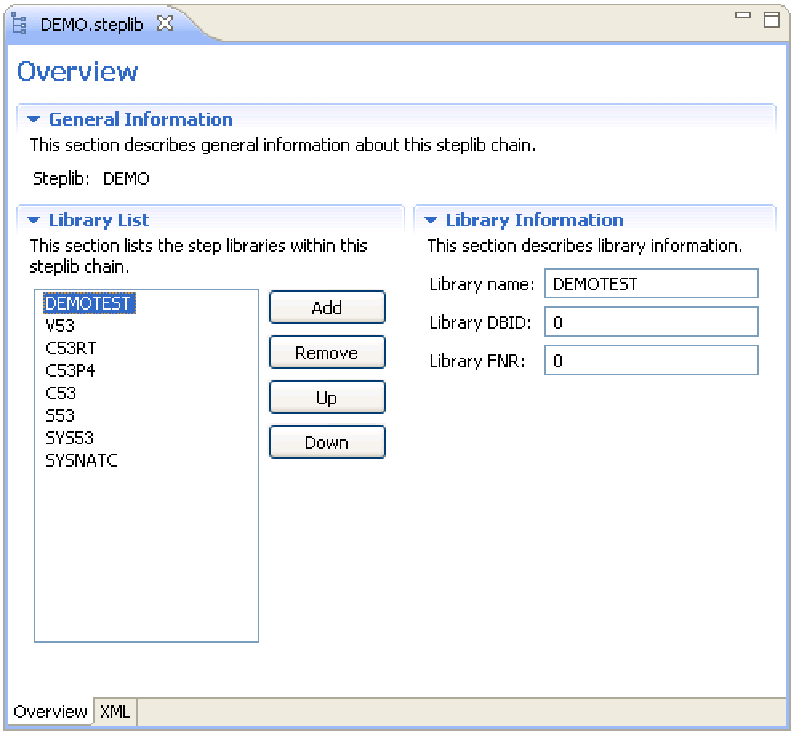
1 Expand the steplibs node in the **Navigator** view.

2 Open the `SteplibName`.steplib file for the business service.

The steplib editor is displayed.

3 Select a step library in the **Library List** section.

Information about that library is displayed. For example:



Overview

General Information

This section describes general information about this steplib chain.

Steplib: DEMO

Library List

This section lists the step libraries within this steplib chain.

DEMOTEST
V53
C53RT
C53P4
C53
S53
SYS53
SYSNATC

Library Information

This section describes library information.

Library name: DEMOTEST

Library DBID: 0

Library FNR: 0

Note: The DBID and FNR values are not required; they are only used in advanced configurations when uploading to the server.

4 Edit information about the steplib chain and step libraries used for the business service.

Using this editor, you can:

Task	Procedure
Add a step library to the steplib chain.	See *Add a Step Library to the Steplib Chain*. **Note:** For information on creating a new step library, see *Create a Step Library*.
Remove a step library from the steplib chain.	Select the library in the **Library List** section and then select **Remove**.
Reorder the list of step libraries within the steplib chain.	Select **Up** and **Down** to reorder the list.
Change the name of a step library.	Select the library in the **Library List** section and change the library name in the **Library Information** section.

Note: The DBID and FNR values are not required; they are only used in advanced configurations when uploading to the server.

5 Save the steplib file.

Notes:

1. You can upload the steplib file to the server using standard NaturalONE functionality.
2. For more information on steplib chains, see *Defining Steplibs and Domains*, *Natural Business Services Administration*.

22 Generating Java Clients and Web Services

After creating business services locally, or downloading them from an existing server installation, you can create Java clients and/or Web services for the services. This section covers the following topics:

Generate a Java Client

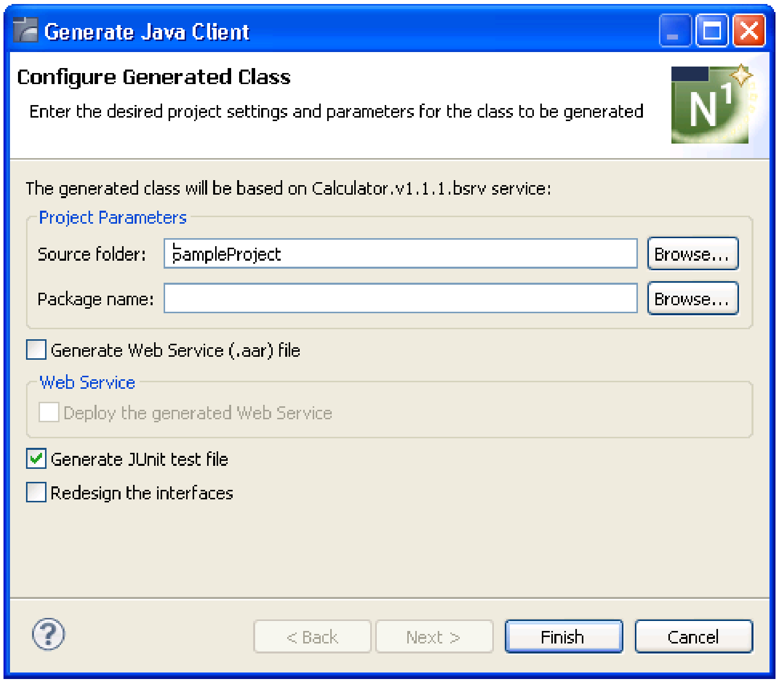
Using the **Generate Java Client** wizard, you can develop Java client applications that access Natural Business Services server components via dynamic RPC. The wizard creates Java clients from Natural Business Services objects in a NaturalONE project and then generates the Java client interface objects used by the Java client to access the server components.

To generate a Java client:

1 Open the context menu for the business service in the **Navigator** view (i.e., the `BusinessServiceName`**.bsrv** file).

2 Select **Generate Java Client**.

The **Configure Generated Class** panel is displayed. For example:



Generate Java Client

Configure Generated Class

Enter the desired project settings and parameters for the class to be generated

The generated class will be based on Calculator.v1.1.1.bsrv service:

Project Parameters

Source folder: SampleProject

Package name:

□ Generate Web Service (.aar) file

Web Service

□ Deploy the generated Web Service

☑ Generate JUnit test file

□ Redesign the interfaces

Using this panel, you can:

Task	Procedure
Generate a Web service file (file with a .aar extension).	Select **Generate Web Service (.aar) file**. The Web service file will be created when the class is generated.
Deploy the generated Web service to a WS-Stack runtime environment.	See *Deploy a Generated Web Service*.
Suppress the generation of a JUnit test file containing test functions that correspond to each method in the business service.	Deselect **Generate JUnit test file**. **Note:** For a description of the JUnit test file, see *Test the Java Class*.
Redesign the interface for Natural subprograms, such as changing the default IDL directions or selecting which portions of a redefined field to use.	See *Redesign the Interface*.

3 Type the name of the source folder in **Source folder** or select **Browse** to browse for a folder.

If you select a project as the source folder, the wizard will automatically create and use <`projectname`>/Java/src as the source folder. In addition, the wizard will set the output/compilation folder to the bin folder corresponding to the source. For example, if the source folder is set to Java/src, then the output/compilation folder will be set to Java/bin.

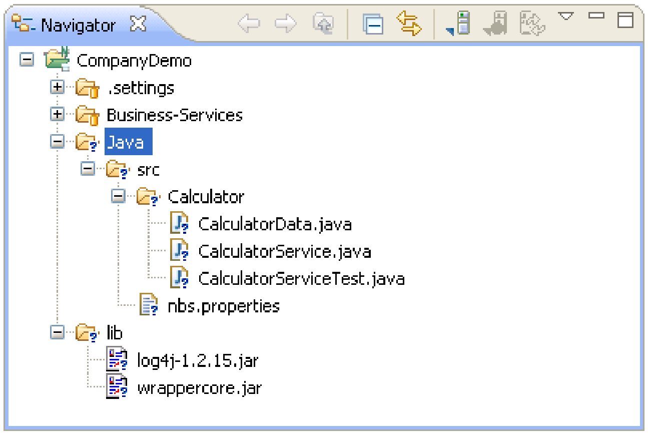
Note: The folder must currently exist.

4 Type the name of the package in **Package name** or select **Browse** to browse for a package.

The package does not have to currently exist.

5 Select **Finish** to generate the Java class.

The class and associated files are displayed in the **Navigator** view. For example:

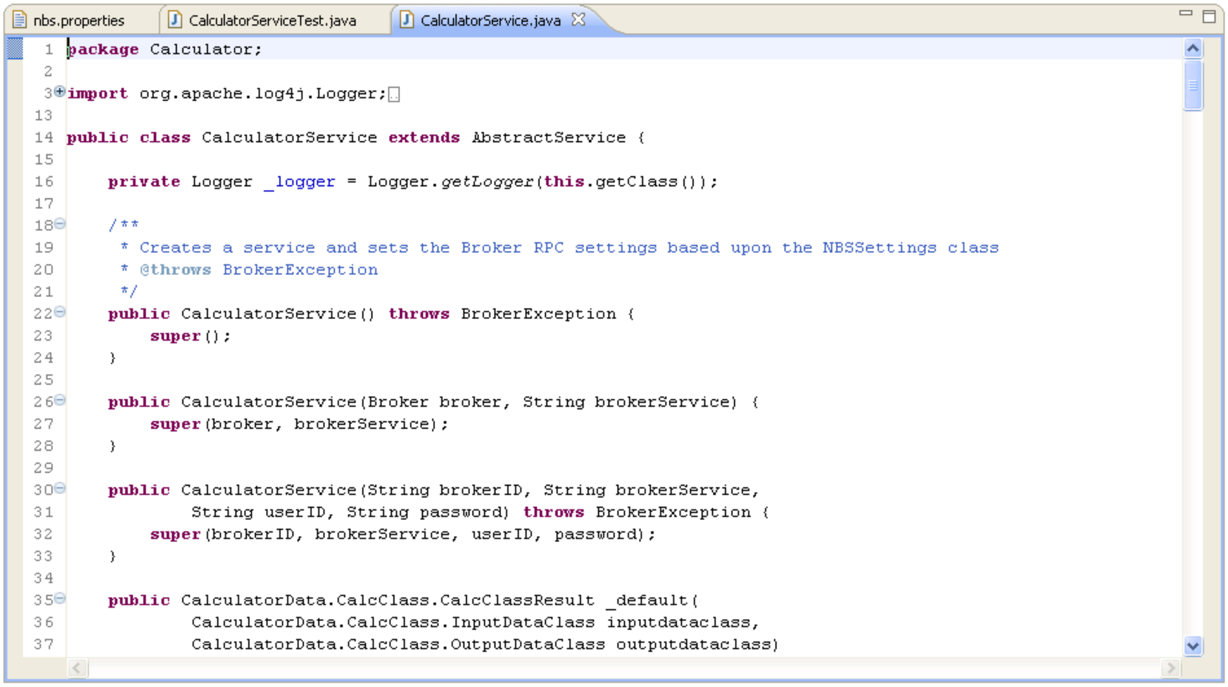


```
CompanyDemo
    .settings
    Business-Services
    Java
        src
            Calculator
                CalculatorData.java
                CalculatorService.java
                CalculatorServiceTest.java
            nbs.properties
    lib
        log4j-1.2.15.jar
        wrappercore.jar
```

In this example, the following items were generated in the **Calculator** node:

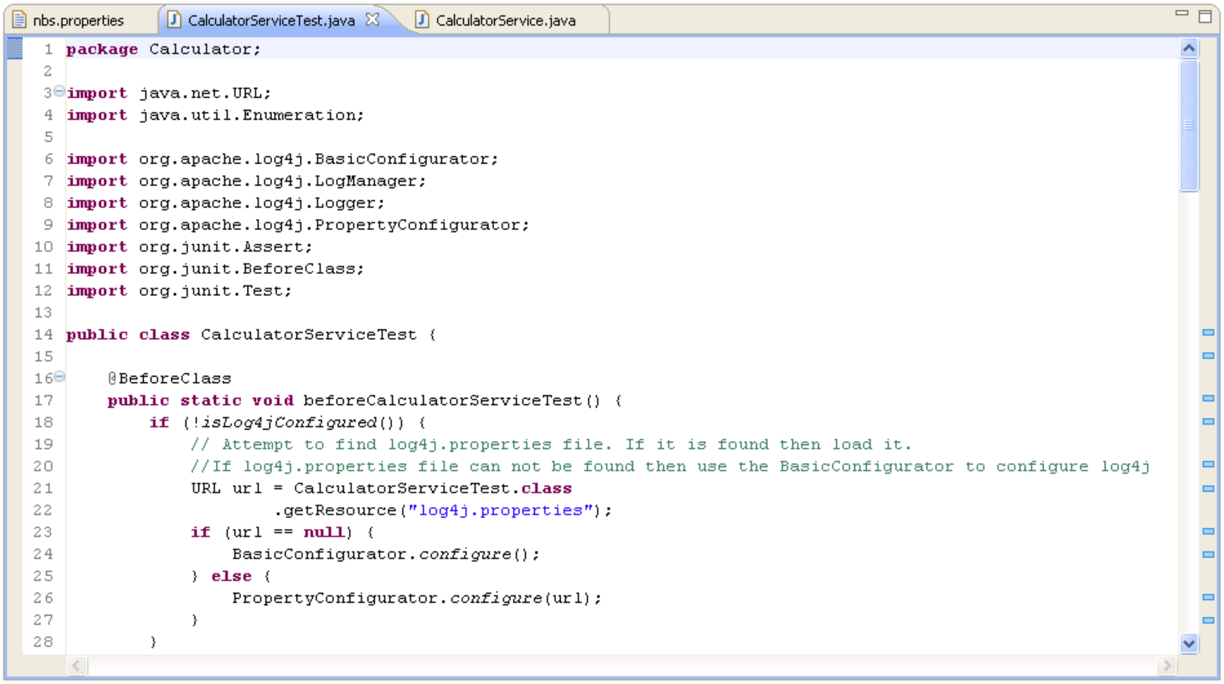
Item Generated	Description
Calculator node in src	Java package.
CalculatorData.java	Java class containing the PDA for the business service, which is mapped to Java. This class has the same name as the business service and a suffix of "Data".
CalculatorService.java	Java class containing the code that communicates with the business service, as well as the Add method. This class has the same name as the business service and a suffix of "Service".
CalculatorServiceTest.java	JUnit class with the same name as the *CalculatorService.java* class and a suffix of "Test". For more information, see *Test the Java Class*. **Note:** This file is only generated when the **Generate JUnit test file** option is selected.
nbs.properties file	A .properties file for the business service. For more information, see *Set the nbs.properties File to Run a Dynamic RPC Java Class*.
log4j-1.2.15.jar file in lib	Library containing the logging functionality.
wrappercore.jar file in lib	Library containing the helper classes and functionality for the generated Java class.

The business service class is displayed in the editor view. For example:



```
package Calculator;

import org.apache.log4j.Logger;

public class CalculatorService extends AbstractService {

    private Logger _logger = Logger.getLogger(this.getClass());

    /**
     * Creates a service and sets the Broker RPC settings based upon the NBSSettings class
     * @throws BrokerException
     */
    public CalculatorService() throws BrokerException {
        super();
    }

    public CalculatorService(Broker broker, String brokerService) {
        super(broker, brokerService);
    }

    public CalculatorService(String brokerID, String brokerService,
            String userID, String password) throws BrokerException {
        super(brokerID, brokerService, userID, password);
    }

    public CalculatorData.CalcClass.CalcClassResult _default(
            CalculatorData.CalcClass.InputDataClass inputdataclass,
            CalculatorData.CalcClass.OutputDataClass outputdataclass)
```

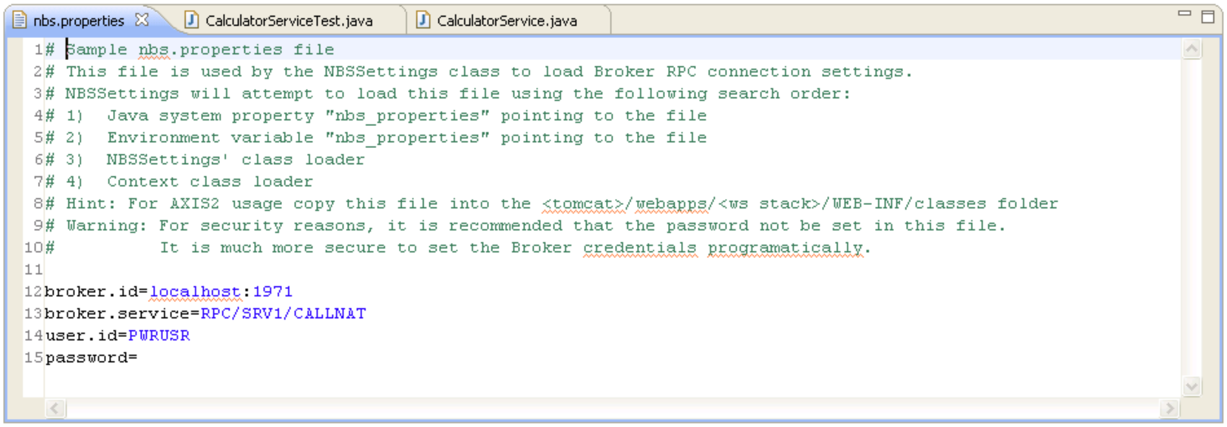
To display the JUnit class, select the **CalculatorServiceTest.java** tab. For example:



```java
package Calculator;

import java.net.URL;
import java.util.Enumeration;

import org.apache.log4j.BasicConfigurator;
import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.junit.Assert;
import org.junit.BeforeClass;
import org.junit.Test;

public class CalculatorServiceTest {

    @BeforeClass
    public static void beforeCalculatorServiceTest() {
        if (!isLog4jConfigured()) {
            // Attempt to find log4j.properties file. If it is found then load it.
            //If log4j.properties file can not be found then use the BasicConfigurator to configure log4j
            URL url = CalculatorServiceTest.class
                    .getResource("log4j.properties");
            if (url == null) {
                BasicConfigurator.configure();
            } else {
                PropertyConfigurator.configure(url);
            }
        }
```

Note: This file is only generated when the **Generate JUnit test file** option is selected.

To display the properties file for the business service, select the **nbs.properties** tab. For example:



```
# Sample nbs.properties file
# This file is used by the NBSSettings class to load Broker RPC connection settings.
# NBSSettings will attempt to load this file using the following search order:
# 1)  Java system property "nbs_properties" pointing to the file
# 2)  Environment variable "nbs_properties" pointing to the file
# 3)  NBSSettings' class loader
# 4)  Context class loader
# Hint: For AXIS2 usage copy this file into the <tomcat>/webapps/<ws stack>/WEB-INF/classes folder
# Warning: For security reasons, it is recommended that the password not be set in this file.
#          It is much more secure to set the Broker credentials programatically.

broker.id=localhost:1971
broker.service=RPC/SRV1/CALLNAT
user.id=PWRUSR
password=
```

This section covers the following topics:

- Deploy a Generated Web Service
- Redesign the Interface
- Set the nbs.properties File to Run a Dynamic RPC Java Class
- Use log4j Logging Statements in a Runtime Environment
- Test the Java Class

- Write an Application to Use the Generated Java Class

Deploy a Generated Web Service

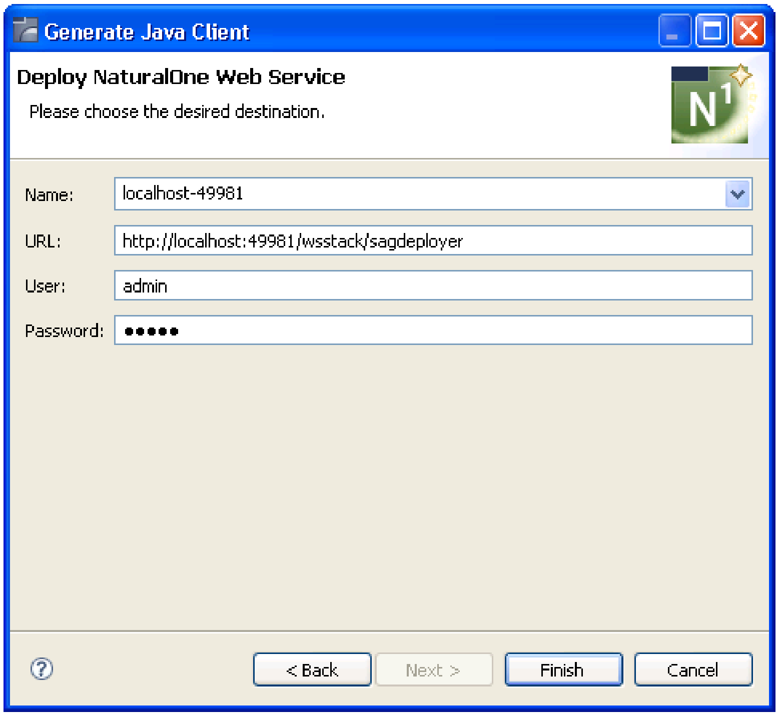
This section describes how to deploy a generated Web service to a WS-Stack runtime environment.

Note: To use this option, you must first select **Generate Web Service (.aar) file** on the **Configure Generated Class** panel.

To deploy a generated Web service:

1. Select **Deploy the generated Web Service**.
2. Select **Next**.

   The **Deploy NaturalONE Web Service** panel is displayed. For example:



Deploy NaturalOne Web Service

Please choose the desired destination.

Name: localhost-49981

URL: http://localhost:49981/wsstack/sagdeployer

User: admin

Password: ●●●●●

   This panel displays the default parameters for the WS-Stack runtime environment running in the default Tomcat servlet engine. The default user credentials are also displayed (these can be changed at first logon).

3. Confirm the name and location of the WS-Stack runtime environment.
4. Confirm the user credentials.
5. Select **Finish**.

For more information on deploying a Web service or changing logon credentials, refer to the Web Services Stack documentation.

Redesign the Interface

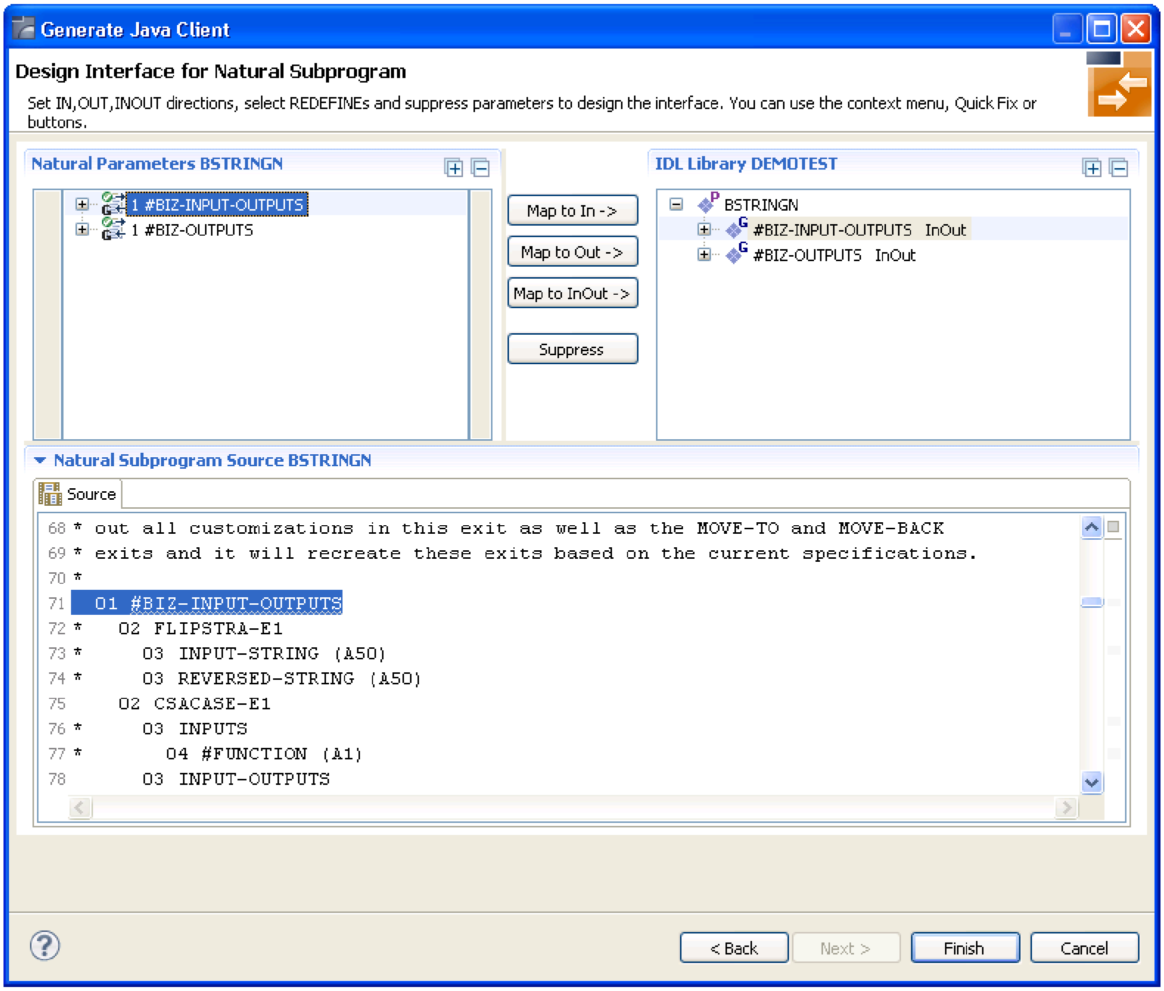
This section describes how to redesign the interface generated for a Java client.

To redesign the interface:

1 Select **Redesign the interfaces** on the **Configure Generated Class** panel.

2 Select **Next**.

The **Design Interface for Natural Subprogram** panel is displayed. For example:



Generate Java Client

Design Interface for Natural Subprogram

Set IN,OUT,INOUT directions, select REDEFINEs and suppress parameters to design the interface. You can use the context menu, Quick Fix or buttons.

```
* out all customizations in this exit as well as the MOVE-TO and MOVE-BACK
* exits and it will recreate these exits based on the current specifications.
*
  01 #BIZ-INPUT-OUTPUTS
*    02 FLIPSTRA-E1
*      03 INPUT-STRING (A50)
*      03 REVERSED-STRING (A50)
     02 CSACASE-E1
*      03 INPUTS
*        04 #FUNCTION (A1)
       03 INPUT-OUTPUTS
```

The upper left pane on this panel displays the parameters for the Natural subprogram and allows you to select operations (for example, Map to In, Map to Out, Map to InOut, or Suppress). The upper right pane displays the IDL directions that will be inserted as top-level parameters (level 1) in the interface definition for the Natural subprogram. The bottom pane is used for reference purposes; it displays the Natural subprogram source and its PDA sources, each in a separate tab.

Tip: To reserve more space for viewing the upper panes, you can close the bottom pane when it is not needed.

Using this panel, you can:

Task	Procedure
Define the direction of parameters in the interface.	See *Define the Direction of Parameters*.
Select which portions of redefined fields in the PDA to use.	See *Select Portions of Redefined Fields*.
Hide or suppress unneeded parameters in the interface.	See *Suppress Parameters*.

3 Select **Finish**.

If you selected multiple Natural subprograms, you can now redesign the next interface. The current and total number of subprograms designed so far is indicated in the title (for example, 2/3).

Note: For more information about using this panel, refer to the EntireX documentation.

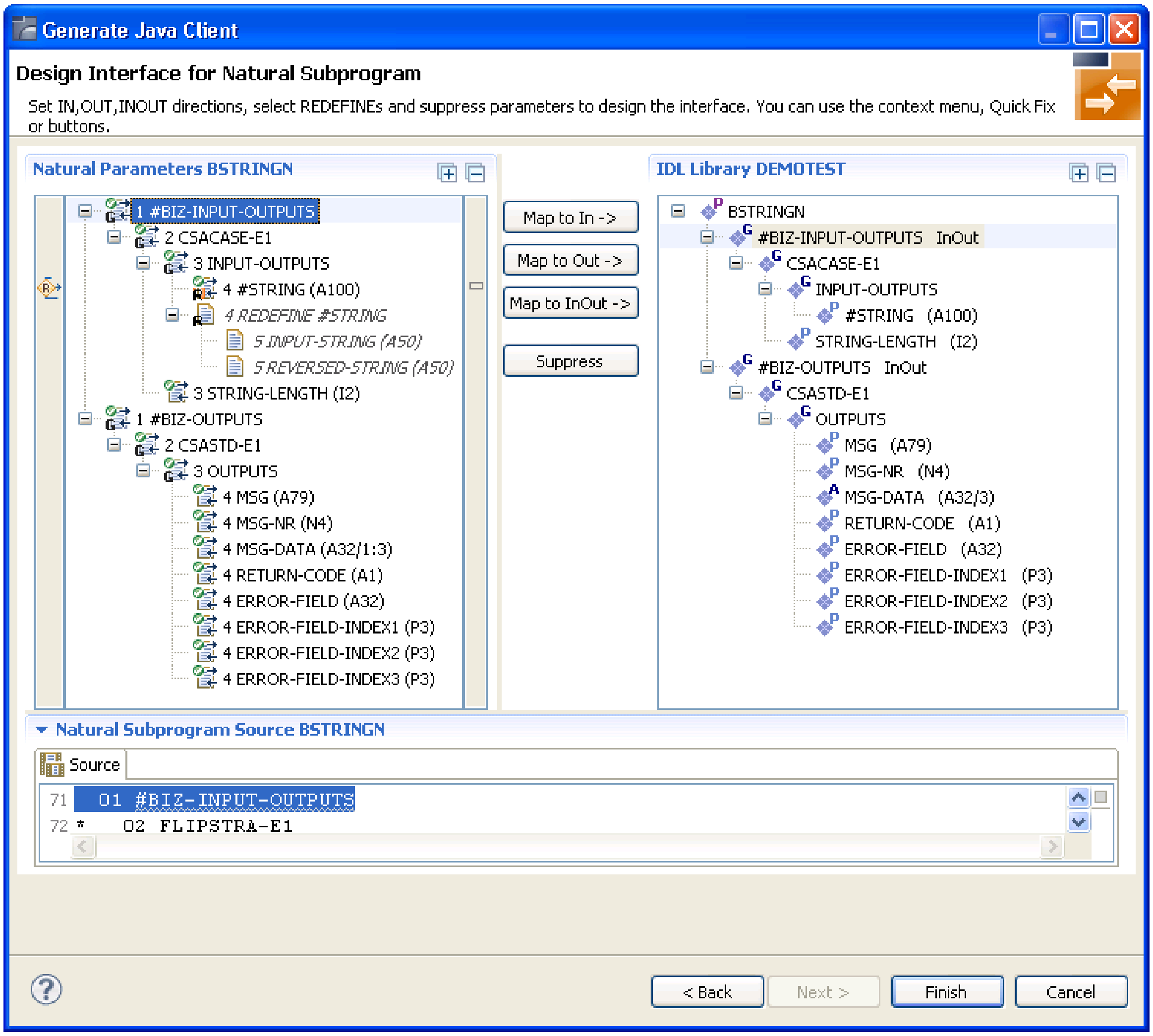
Define the Direction of Parameters

This section describes how to change the default IDL directions for the generated Java client.

▶ To define the direction of parameters:

1 Expand the INPUT-OUTPUT and OUTPUT nodes in the upper left pane of the **Design Interface for Natural Subprogram** panel.

For example:



Generate Java Client

Design Interface for Natural Subprogram

Set IN,OUT,INOUT directions, select REDEFINEs and suppress parameters to design the interface. You can use the context menu, Quick Fix or buttons.

```
71   01 #BIZ-INPUT-OUTPUTS
72 *    02 FLIPSTRA-E1
```

Note: Natural parameters that are mapped in the interface are marked with a green circle.

2 Open the context menu for the first level 1 parameter you want to define.

3 Select one of the following options:

- **Map to In**
- **Map to Out**
- **Map to InOut** (the default direction, if no other direction is specified)

The operation is reflected in the IDL Library pane.

4 Continue defining level 1 parameters as desired.

5 Select **Finish**.

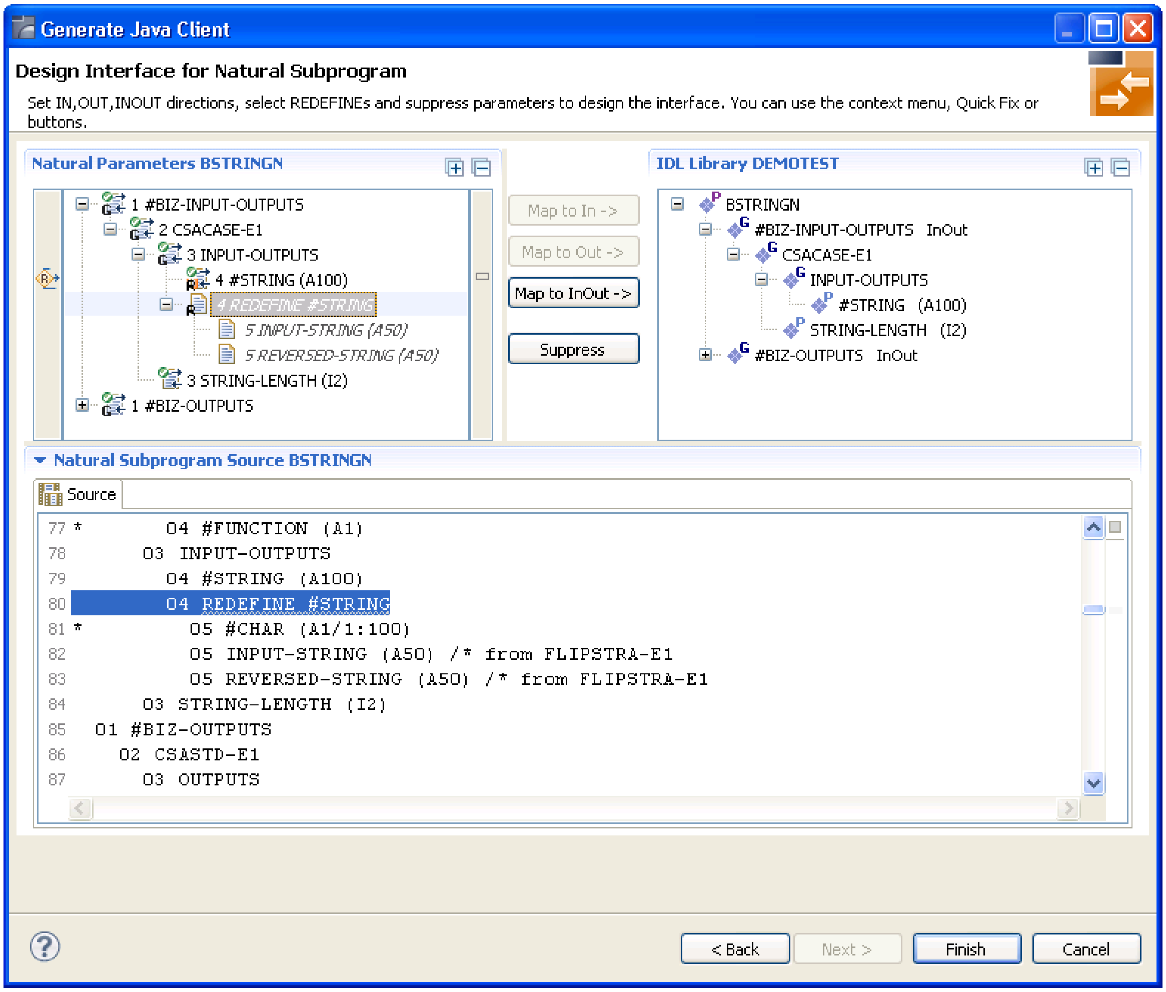
Select Portions of Redefined Fields

This section describes how to select which parameters redefined in the Natural PDA are used in the interface.

To define which portions of a redefined field to use:

1 Expand the INPUT-OUTPUT and OUTPUT nodes in the upper left pane of the **Design Interface for Natural Subprogram** panel.

For example:



Set IN,OUT,INOUT directions, select REDEFINEs and suppress parameters to design the interface. You can use the context menu, Quick Fix or buttons.

```
*   04 #FUNCTION (A1)
   03 INPUT-OUTPUTS
    04 #STRING (A100)
    04 REDEFINE #STRING
*    05 #CHAR (A1/1:100)
     05 INPUT-STRING (A50) /* from FLIPSTRA-E1
     05 REVERSED-STRING (A50) /* from FLIPSTRA-E1
   03 STRING-LENGTH (I2)
 01 #BIZ-OUTPUTS
  02 CSASTD-E1
   03 OUTPUTS
```

Redefined fields are identified by the Ⓡ icon.

2 Open the context menu for the REDEFINE base parameter or any REDEFINE path.

3 Select **Map to InOut**.

4 Select **Finish**.

Suppress Parameters

This section describes how to suppress unnecessary parameters, which creates a less cluttered IDL client interface and minimizes the amount of data transferred at runtime.

To suppress parameters in the interface:

1 Open the context menu for the parameter in the upper left pane.

2 Select **Suppress**.

3 Select **Finish**.

Natural parameters suppressed in the interface are displayed in italic type.

Set the nbs.properties File to Run a Dynamic RPC Java Class

You can set the *nbs.properties* file to run a dynamic RPC Java class. The settings to run the Java class are loaded by the NBSSettings class. This class will try to find the *nbs.properties* file using the following search order:

1. nbs_properties Java system property pointing to the file
2. nbs_properties environment variable pointing to the file
3. class loader for NBSSettings
4. class loader for context for current thread

The entries in the *nbs.properties* file are:

Setting	Description	Default
broker.id	Broker connection ID.	localhost:1971
broker.service	Broker service endpoint.	RPC/SRV1/CALLNAT
user.id	User ID to use with Broker.	GUEST
password	Password to use with Broker.	blank/empty string

For example:

```
broker.id=BKR13003:4010
broker.service=RPC/NBS53DEV/CALLNAT
user.id=PWRUSR
password=
```

Note: If the generated code will be used in AXIS2, you must copy the *nbs.properties* file to the <*WebApp*>/*classes* folder.

Caution: For security reasons, it is recommended that the password not be set in the *nbs.properties* file. It is much more secure to set the Broker credentials programmatically.

Define Different User IDs and Passwords for Web Services

Both the `user.id` and `password` properties can use a Java class suffix (for example, package.MyService), which allows the client to use different user IDs and passwords for different services and/or groups of services. For example:

```
user.id.package.name.MyService=USERIDA
password.package.name.MyService-PWDA
user.id.package.name=USERIDB
password.package.name=PWDB
user.id=PWRUSR
password=
```

When the `MyService` Web service in the `package.name` Java package is invoked, the `USERIDA` user ID and `PWDA` will be used. When any other service within the `package.name` Java package is invoked, the `USERIDB` user ID and `PWDB` password will be used. All other services will use the `PWRUSR` user ID and no password.

Use log4j Logging Statements in a Runtime Environment

The generated Java classes support log4j logging statements for advanced diagnostics at runtime. If logging is being used, there is the potential for data to be visible in the log files produced in the runtime environment. To avoid this problem, configure the com.softwareag.naturalone.gen.dynamicrpc.core logger file to only display warnings and above. For example:

```
log4j.logger.com.softwareag.naturalone.gen.dynamicrpc.core=WARN
```

Test the Java Class

When you select **Generate JUnit test file** on the **Configure Generated Class** panel before generating the class, a JUnit class is also generated. This class contains test functions that correspond to each method in the business service. For example:

```
@Test
 public void testAdd() throws Exception {
  Calculator service = createService();
  CalculatorData data = new CalculatorData();
  //TODO populate data
  CalculatorData result = service.add(data);
  Assert.assertNotNull("Result is null", result);
  //TODO check result
 }
```

//TODO comments have been added in the source to indicate where additional information is added for the test. By default, the test succeeds if no errors occur during the call to the business service and if data is returned.

Data can be added before the service method is invoked. For example:

```
  //TODO populate data
  data.getCalcInput_data().set_first_num(new BigDecimal(2));
  data.getCalcInput_data().set_second_num(new BigDecimal(2));
```

In addition, an assert clause can be included to verify that the result is correct. For example:

```
  //TODO check result
  Assert.assertEquals(new BigDecimal(4),
   result.getCalcOutput_data().get_result());
```

Tests can be invoked using standard Eclipse JUnit functionality (i.e., run as Junit test). The environment for the test is contained in this function in the test file. For example:

```
 private Broker createBroker() throws BrokerException {
  String userID = "GUEST";
  String password = "";
  Broker broker =
new Broker(Demotest.DEFAULT_BROKERID, userID,
    Demotest.DEFAULT_SERVER);
  broker.logon(password);
  return broker;
 }
```

Write an Application to Use the Generated Java Class

After generating a Java class, you can write an application that uses the class to invoke a business service. Use the following code as a template for writing the application; read the comments for additional information:

```
// Create the EntireX Broker object
Broker b = new Broker("localhost:1971","GUEST");
// Create the service class.
Calculator calcservice = new Calculator(b, "RPC/SRV1/CALLNAT");
// Create the service data class
CalculatorData calcdata = new CalculatorData();
// Set input values for the call
calcdata.getCalcInput_data().set_first_num(new BigDecimal(2));
calcdata.getCalcInput_data().set_second_num(new BigDecimal(2));
calcdata.getCalcInput_data().set_function("Add");
// Invoke a service method, passing in the data class
// make sure to catch any errors that might happen.
CalculatorData calcresult = null;
try {
 calcresult = calcservice.add(calcdata);
} catch (DataException e) {
 e.printStackTrace();
} catch (BrokerException e) {
 e.printStackTrace();
} catch (ServiceException e) {
 e.printStackTrace();
}
```

```
// Examine the result
System.out.println("Result is:"
  + calcresult.getCalcOutput_data().get_result());
```

Generate a Web Service

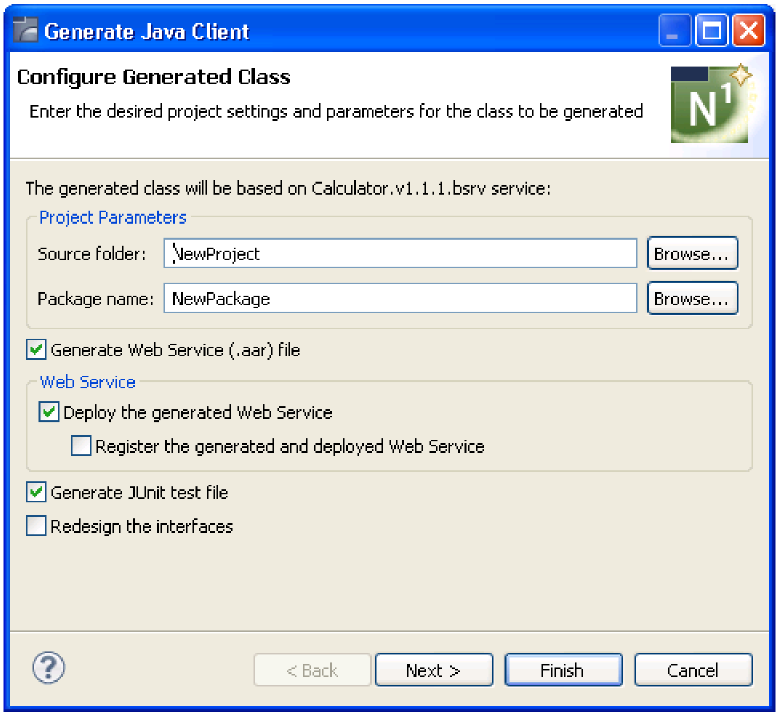
Using the **Generate Java Client** wizard, you can generate a Web service from business service object in a NaturalONE project. The generated Web service objects are WS-Stack Web service archive files (files with a .aar extension) and can be deployed in a Web services Stack runtime environment and registered in CentraSite. Web service client applications can then access these Web services and expose business logic implemented by the Natural server components.

To generate a Web service:

1 Open the context menu for the business service in the **Navigator** view (i.e., the *BusinessServiceName*.**bsrv** file).

2 Select **Generate Web Service**.

The **Configure Generated Class** panel is displayed. For example:



Enter the desired project settings and parameters for the class to be generated

The generated class will be based on Calculator.v1.1.1.bsrv service:

Project Parameters

Source folder: NewProject

Package name: NewPackage

☑ Generate Web Service (.aar) file

Web Service

☑ Deploy the generated Web Service

□ Register the generated and deployed Web Service

☑ Generate JUnit test file

□ Redesign the interfaces

Using this panel, you can:

Task	Procedure
Change the source folder in which to generate the Java class.	Type the name of the source folder in **Source folder** or select **Browse** to browse for an existing project. The folder must currently exist.
Change the name of the Java package or select another package.	Type the name of the Java package in **Package name** or select **Browse** to browse for an existing package. The package does not have to currently exist.
Suppress the deployment of the generated Web service to a WS-Stack runtime environment.	Deselect **Deploy the generated Web Service**.
Register the Web service after it has been generated and deployed.	Select **Register the generated and deployed Web Service**. **Note:** To select this option, **Deploy the generated Web Service** must also be selected.
Suppress the generation of a JUnit test file containing test functions that correspond to each method in the business service.	Deselect **Generate JUnit test file**. **Note:** For a description of the JUnit test file, see *Test the Java Class*.

3 Select **Next**.

The **Deploy NaturalONE Web Service** panel is displayed. For a description of this panel, see *Deploy a Generated Web Service*.

4 Select **Finish** to generate the Java class for the Web service.

Configure the Web Service to Use WS-Security

All generated services extend "com.softwareag.naturalone.gen.dynamicrpc.core.AbstractService". The default constructor for the generated service determines which Broker credentials to use based on the following rules:

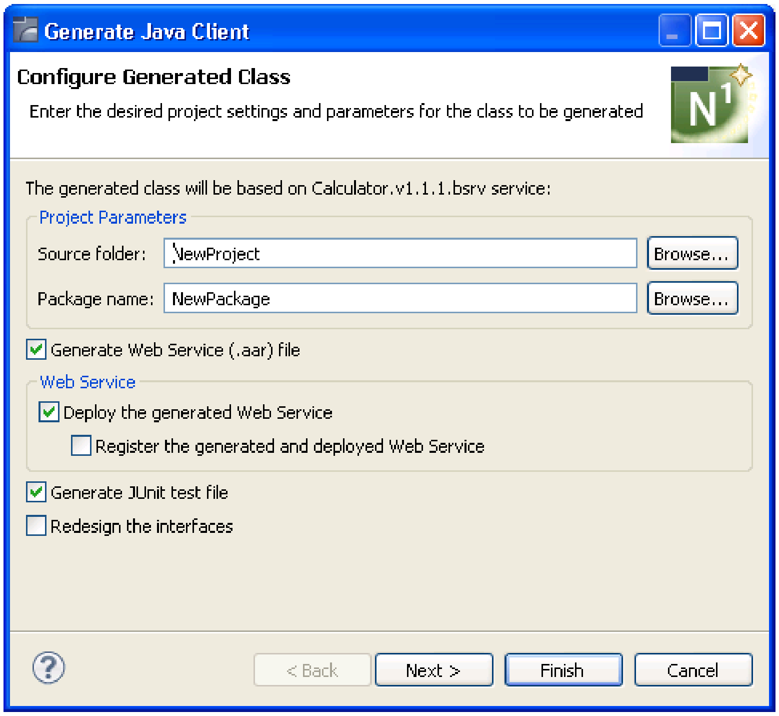
1. When WS-Security is turned on and the credentials are passed in the SOAP header, they will be used in the call to Broker.
2. In all other cases, the settings in the *nbs.properties* file will be used. For information, see *Set the nbs.properties File to Run a Dynamic RPC Java Class*.

To configure a Web service to use WS-Security:

1 Open the context menu for a business service in the **Navigator** view.

2 Select **Generate Web Service**.

The **Configure Generated Class** panel is displayed. For example:



Configure Generated Class

Enter the desired project settings and parameters for the class to be generated

The generated class will be based on Calculator.v1.1.1.bsrv service:

Project Parameters

Source folder: NewProject

Package name: NewPackage

☑ Generate Web Service (.aar) file

Web Service

☑ Deploy the generated Web Service

□ Register the generated and deployed Web Service

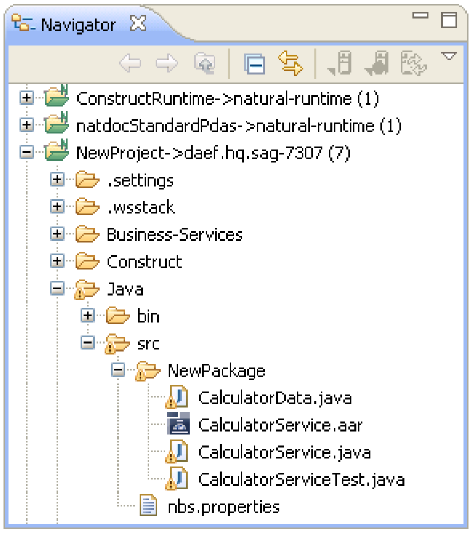
☑ Generate JUnit test file

□ Redesign the interfaces

3 Deselect **Deploy the generated Web Service**.

4 Select **Finish**.

The generated Web service files are displayed in the **Navigator** view. For example:



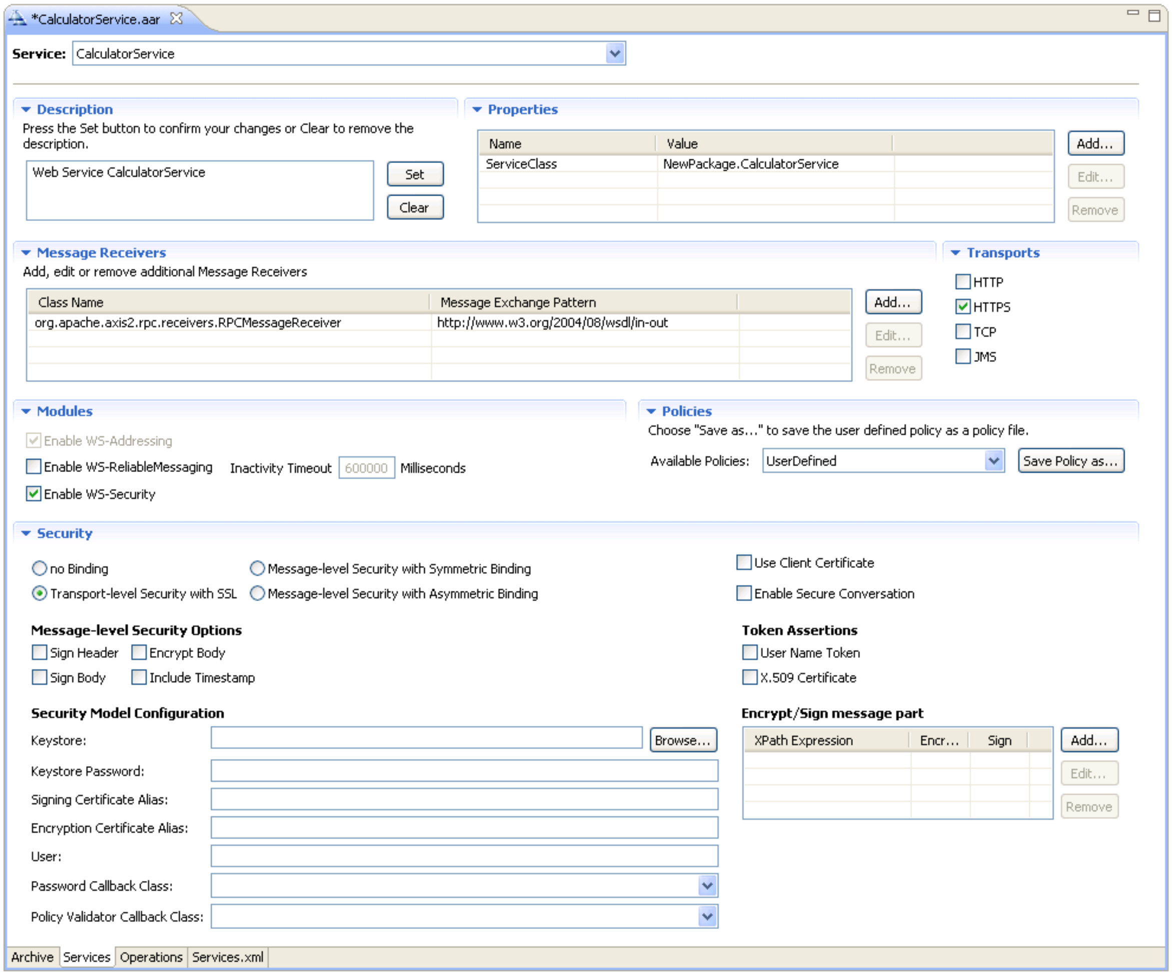
```
⊞ ConstructRuntime->natural-runtime (1)
⊞ natdocStandardPdas->natural-runtime (1)
⊟ NewProject->daef.hq.sag-7307 (7)
    ⊞ .settings
    ⊞ .wsstack
    ⊞ Business-Services
    ⊞ Construct
    ⊟ Java
        ⊞ bin
        ⊟ src
            ⊟ NewPackage
                CalculatorData.java
                CalculatorService.aar
                CalculatorService.java
                CalculatorServiceTest.java
            nbs.properties
```

5 Open the generated .aar file in the editor view.

6 Select the **Services** tab.

7 Select **Enable WS-Security** in the **Modules** section.

The settings in the **Security** section are displayed. For example:



*CalculatorService.aar

Service: CalculatorService

Description

Press the Set button to confirm your changes or Clear to remove the description.

Web Service CalculatorService

Properties

Name	Value
ServiceClass	NewPackage.CalculatorService

Message Receivers

Add, edit or remove additional Message Receivers

Class Name	Message Exchange Pattern
org.apache.axis2.rpc.receivers.RPCMessageReceiver	http://www.w3.org/2004/08/wsdl/in-out

Transports

□ HTTP
☑ HTTPS
□ TCP
□ JMS

Modules

☑ Enable WS-Addressing
□ Enable WS-ReliableMessaging Inactivity Timeout 600000 Milliseconds
☑ Enable WS-Security

Policies

Choose "Save as..." to save the user defined policy as a policy file.

Available Policies: UserDefined

Security

○ no Binding
● Transport-level Security with SSL
○ Message-level Security with Symmetric Binding
○ Message-level Security with Asymmetric Binding
□ Use Client Certificate
□ Enable Secure Conversation

Message-level Security Options

□ Sign Header □ Encrypt Body
□ Sign Body □ Include Timestamp

Token Assertions

□ User Name Token
□ X.509 Certificate

Security Model Configuration

Keystore:
Keystore Password:
Signing Certificate Alias:
Encryption Certificate Alias:
User:
Password Callback Class:
Policy Validator Callback Class:

Encrypt/Sign message part

XPath Expression	Encr...	Sign

8 Select the following settings.

- no Binding
- User Name Token (in the **Token Assertions** group).

9 Save the changes to the .aar file.

10 Open the context menu for the .aar file.

11 Select **Web Service Stack > Deploy Web Services Package**.

The **Deploy Your Web Services** panel is displayed. For a description of this panel, see *Deploy a Generated Web Service*.

On the client, the following SOAP header must be included with the correct credentials specified:

```
<soapenv12:Envelope xmlns:soapenv12="http://www.w3.org/2003/05/soap-envelope">
      <soapenv12:Header>
          <wsse:Security xmlns:soapenc12="http://www.w3.org/2003/05/soap-encoding"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema" ↩
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  ↩
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
            >
                  <wsse:UsernameToken>
                        <wsse:Username>MyUser</wsse:Username>
                        <wsse:Password>MyPassword</wsse:Password>
                  </wsse:UsernameToken>
            </wsse:Security>
      </soapenv12:Header>
      <soapenv12:Body>
```

Note: Since user credentials will be included in the SOAP header, it is highly recommended that you use HTTPS. For instructions on how to configure AXIS2 to use HTTPS, refer to the AXIS2 documentation.

23 Setting Preferences for Business Services

Set Business Service Preferences

You can set preferences for the Business Services component in the **Preferences** window for Business Services.

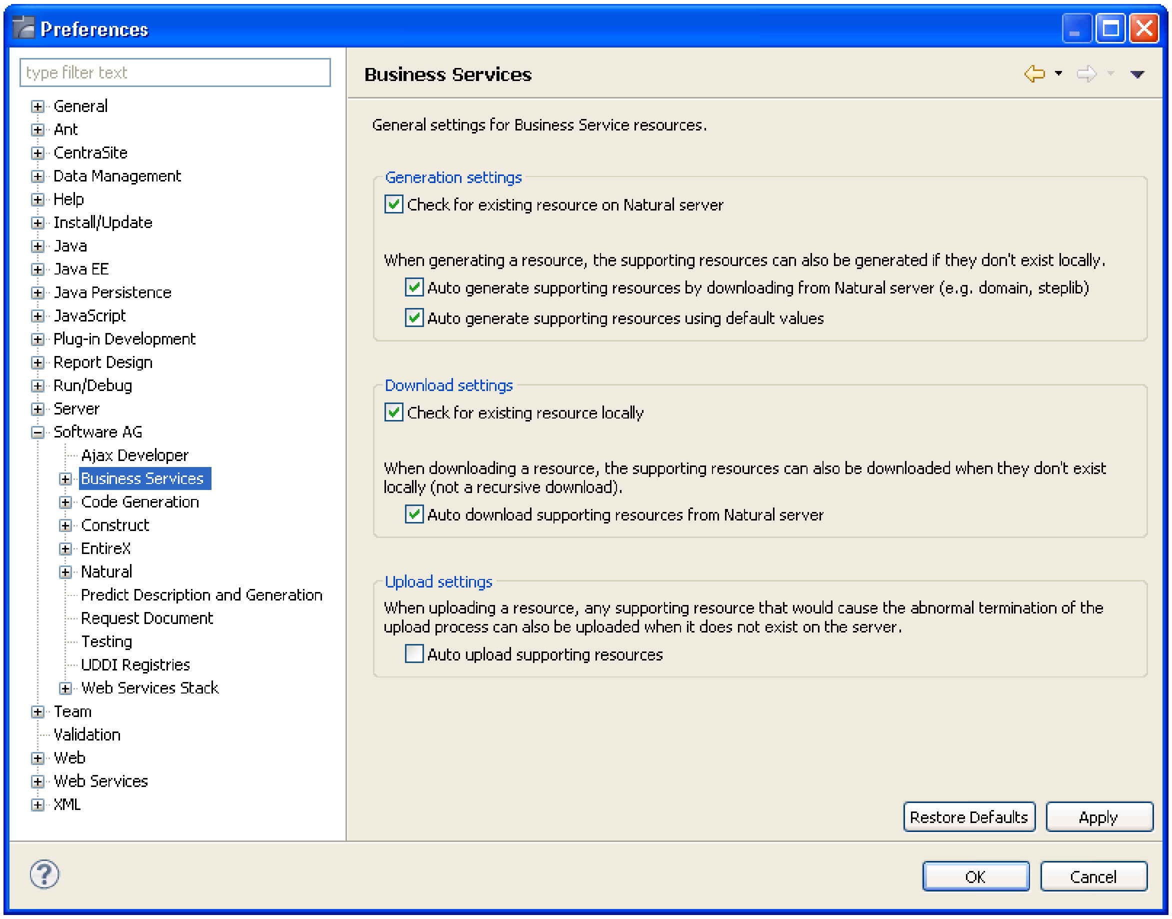
To set preferences for the Business Services component:

1 Select **Preferences** on the **Window** menu.

The **Preferences** window is displayed.

2 Select **Software AG > Business Services**.

The **Business Services** settings are displayed. For example:



Business Services

General settings for Business Service resources.

Generation settings

☑ Check for existing resource on Natural server

When generating a resource, the supporting resources can also be generated if they don't exist locally.

☑ Auto generate supporting resources by downloading from Natural server (e.g. domain, steplib)

☑ Auto generate supporting resources using default values

Download settings

☑ Check for existing resource locally

When downloading a resource, the supporting resources can also be downloaded when they don't exist locally (not a recursive download).

☑ Auto download supporting resources from Natural server

Upload settings

When uploading a resource, any supporting resource that would cause the abnormal termination of the upload process can also be uploaded when it does not exist on the server.

□ Auto upload supporting resources

Using this window, you can:

Option	Description
Check for existing resource on Natural server	When this option is selected, the code generator will check the **Natural server** for the existence of the business service resource being generated and display a warning message if it exists.
Auto generate supporting resources by downloading from Natural server (e.g. domain, steplib)	When this option is selected, the code generator will automatically download supporting resources, such as the domain and/or steplib files, from the **Natural server** if they do not exist locally.
Auto generate supporting resources using default values	When this option is selected, the code generator will automatically generate the supporting resources, such as the domain and/or steplib file, using default values if the resources do not exist locally. **Note:** This option only applies when **Auto generate supporting resources by downloading from Natural server** is not selected, or is selected but the resource does not exist on the **Natural server**.
Check for existing resource locally	When this option is selected, the code generator will determine whether the selected business service resource(s) exists locally. If it does, a warning message will be displayed.
Auto download supporting resources from Natural server	When this option is selected, the code generator will automatically download the supporting resources from the **Natural server** if they do not exist locally.
Auto upload supporting resources	When this option is selected, the code generator will automatically upload dependent resources, such as the domain file, if they do not exist on the **Natural server**. **Note:** This option eliminates server exceptions that can prevent the selected resources from being uploaded to the server.

3 Select **OK** to save the preferences.

Set CentraSite Preferences

You can set up default values for the CentraSite connection in the **Preferences** window for CentraSite. These values will be filled in automatically when the **Add Metadata to CentraSite** wizard is displayed.

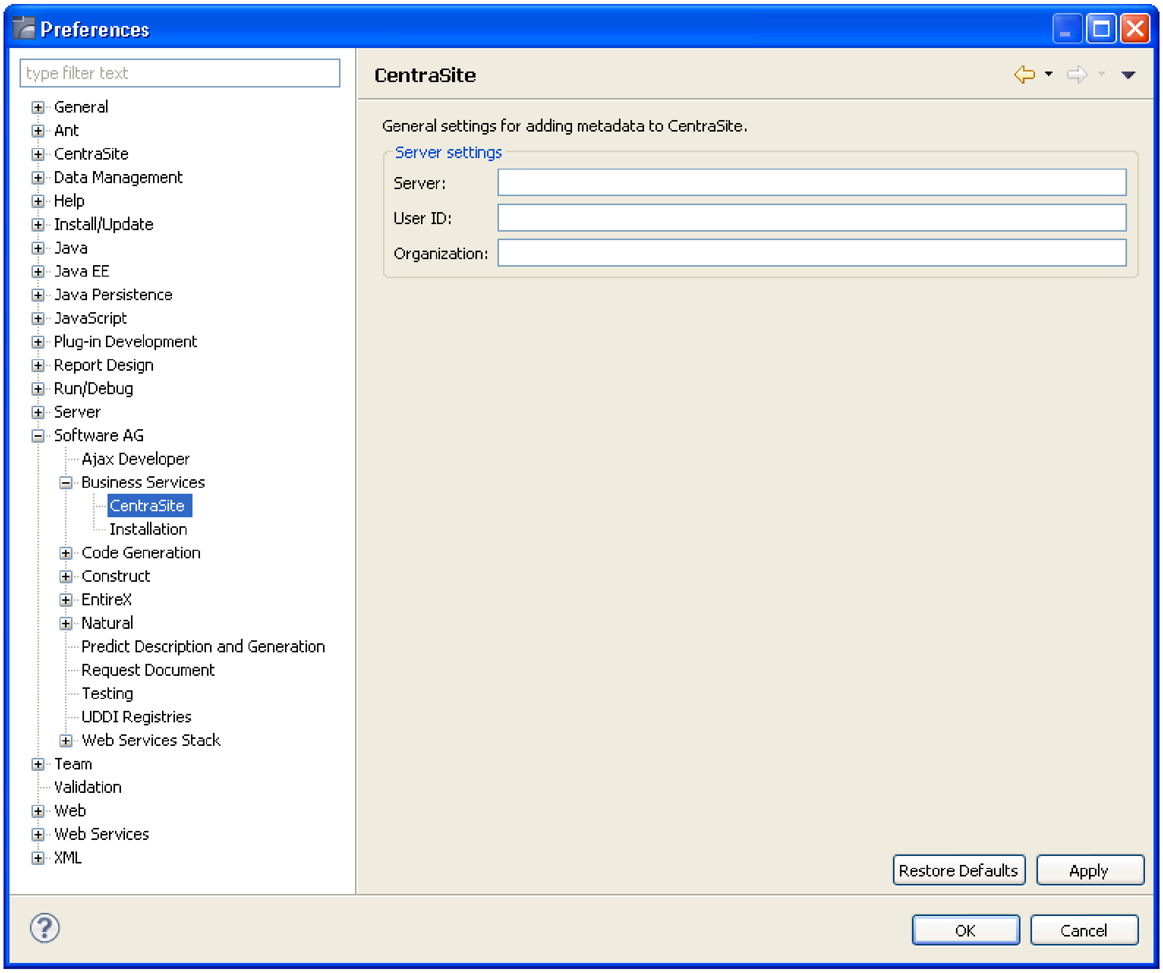
To set CentraSite preferences:

1 Select **Preferences** on the **Window** menu.

The **Preferences** window is displayed.

2 Select **Software AG > Business Services > CentraSite**.

The **Preferences** window for CentraSite is displayed. For example:



General settings for adding metadata to CentraSite.

3 Type the CentraSite connection path in **Server**.

4 Type the user identification for CentraSite in **User ID**.

5 Type the name of the organization in **Organization**.

6 Select **OK** to save the preferences.

The **Server** and **User ID** values will now be provided on the **Add Business Service Metadata to CentraSite** panel when it is displayed.

Set Installation Preferences

To function properly, certain UI functions require a Business Services installation on the Natural server. For example, the Business-Services root node in the **Natural Server** view can be used to download Business Services resources from a Natural server to a local Natural project, but only when there is a Business Services installation on the server. By default, these UI functions will be made visible based on the installation of Business Services on the Natural server. To accomplish this, a server call determines which products are installed on the server and the results are cached until Designer shuts down, which allows for only one server call per host|port|session parameter.

You can set installation preferences in the **Preferences** window for Business Services **Installation**.

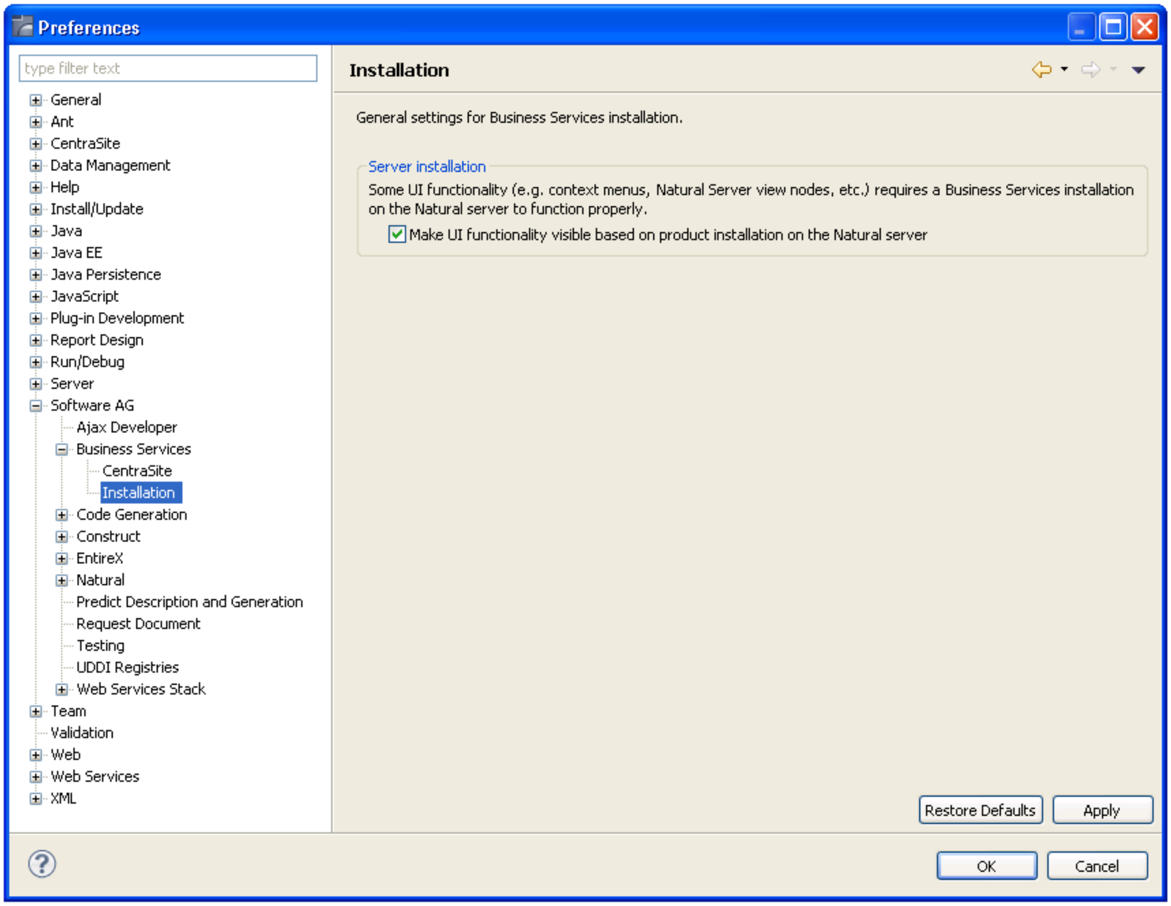
To set installation preferences:

1 Select **Preferences** on the **Window** menu.

The **Preferences** window is displayed.

2 Select **Software AG > Business Services > Installation**.

The **Preferences** window for Installation options is displayed. For example:



Installation

General settings for Business Services installation.

Server installation

Some UI functionality (e.g. context menus, Natural Server view nodes, etc.) requires a Business Services installation on the Natural server to function properly.

☑ Make UI functionality visible based on product installation on the Natural server

Using this window, you can:

Task	Procedure
Make all UI functions visible, even when Business Services is not installed on the Natural server.	Deselect **Make UI functionality visible based on Natural server installation**. No server calls will be made to determine which products are installed on the server.

3 Select **OK** to save the preferences.

24 Setting Security Privileges

Security for business services is applied at the domain level by associating a group with a domain. Access can then be granted or denied at a domain, business service, and/or method level. When you select the **Business Services** root node or a domain node in either the **Navigator** or **Natural Server** view, details about the current security settings are displayed in the **NBS Security** view.

This section covers the following topics:

Note: For more information on defining security, see *Server Security Overview* in *Natural Business Services Administration*.

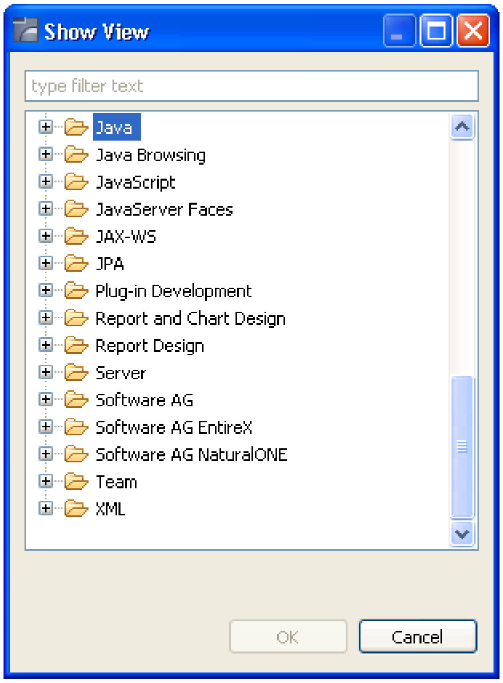
Activate the NBS Security View

By default, the **NBS Security** view is not available.

To activate the NBS Security view:

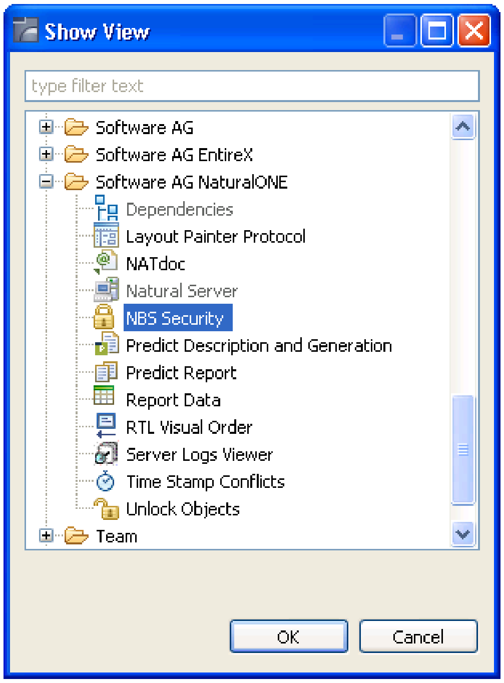
1 Select **Show View > Other** on the **Window** menu.

The **Show View** window is displayed. For example:



2 Select **Software AG NaturalONE > NBS Security**.

For example:



3 Select **OK**.

The **NBS Security** view is displayed.

Note: If you reset the perspective, you must re-activate the **NBS Security** view.

Define Security for a Domain

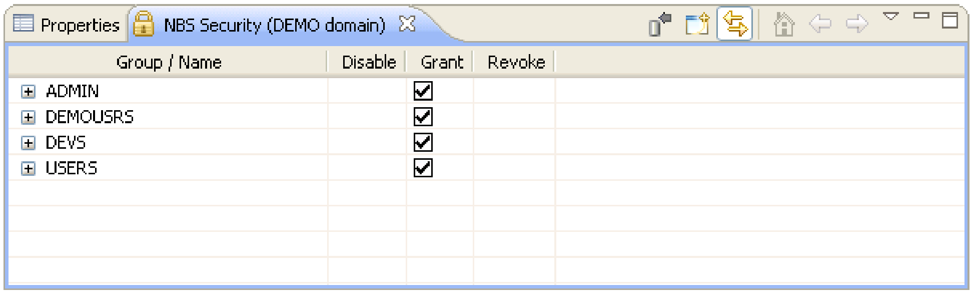
▶ To define security for a domain:

1 Expand the **Business Services** root node in either the **Navigator** or **Natural Server** view.

The list of available domains is displayed.

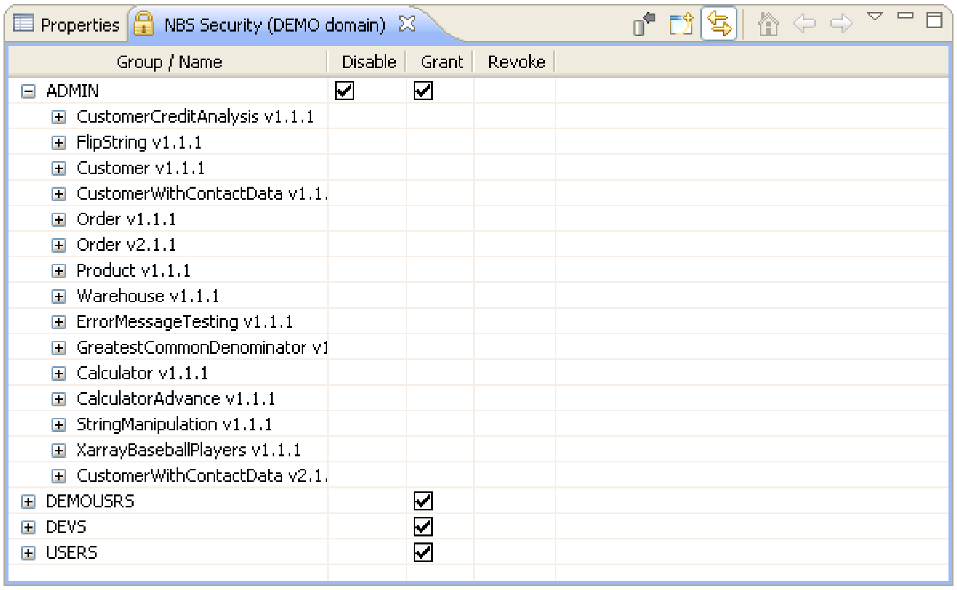
2 Select the domain you want to secure.

The **NBS Security** view displays security details about the groups within the domain. For example:



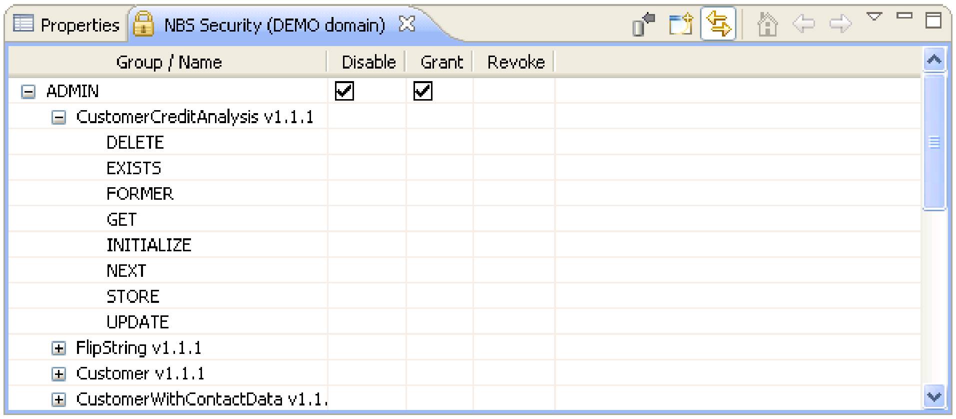
Group / Name	Disable	Grant	Revoke
⊞ ADMIN		☑	
⊞ DEMOUSRS		☑	
⊞ DEVS		☑	
⊞ USERS		☑	

To display the business services within a group, expand the group name. For example:



Group / Name	Disable	Grant	Revoke
⊟ ADMIN	☑	☑	
⊞ CustomerCreditAnalysis v1.1.1			
⊞ FlipString v1.1.1			
⊞ Customer v1.1.1			
⊞ CustomerWithContactData v1.1.			
⊞ Order v1.1.1			
⊞ Order v2.1.1			
⊞ Product v1.1.1			
⊞ Warehouse v1.1.1			
⊞ ErrorMessageTesting v1.1.1			
⊞ GreatestCommonDenominator v1			
⊞ Calculator v1.1.1			
⊞ CalculatorAdvance v1.1.1			
⊞ StringManipulation v1.1.1			
⊞ XarrayBaseballPlayers v1.1.1			
⊞ CustomerWithContactData v2.1.			
⊞ DEMOUSRS		☑	
⊞ DEVS		☑	
⊞ USERS		☑	

To display the methods defined for a service, expand the business service node. For example:



Group / Name	Disable	Grant	Revoke
⊟ ADMIN	☑	☑	
⊟ CustomerCreditAnalysis v1.1.1			
DELETE			
EXISTS			
FORMER			
GET			
INITIALIZE			
NEXT			
STORE			
UPDATE			
⊞ FlipString v1.1.1			
⊞ Customer v1.1.1			
⊞ CustomerWithContactData v1.1.			

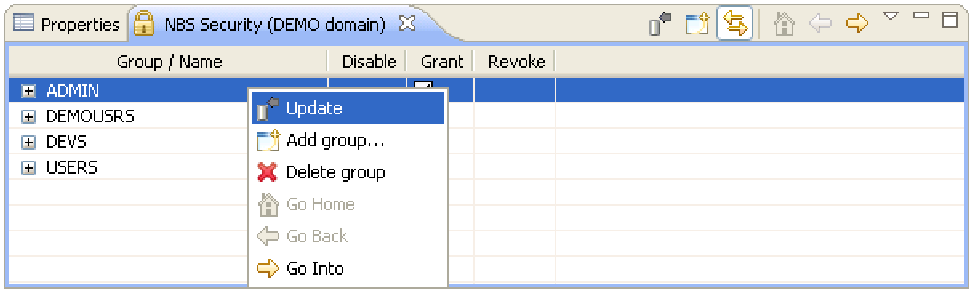
3 Select a security option at the group, service, or method level.

The security options are:

Security Option	Description
Disable	Temporarily revokes access to a group, service, or method. When the group, service, or method is enabled, the previous settings are restored.
Grant	Allows access to the corresponding group.
Revoke	Disallows access to the corresponding group.

4 Open the context menu for the group, service, or method and select **Update**.

For example:



Or:

Select the group, service, or method and select the toolbar option.

A progress window is displayed as the security settings are updated on the server.

Toolbar Options

The following toolbar options are available in the **NBS Security** view:

Toolbar Icon	Description
	Updates the security settings on the server.
	Displays a window to add a group to the current domain. See *Add a Security Group*.
⇆	Synchronizes the domain details in the **NBS Security** view based on which domain is currently selected in the **Navigator** view. For example, if this option is on (as in the example) and you select a different domain in the **Navigator** view, the details for the new domain are displayed in the **NBS Security** view. If this option is off (⇆) and you select a different domain, the details in the view do not change.

Toolbar Icon	Description
	Tip: As it may take some time to retrieve security details from the server, you can turn this option off when you do not require these details.

Add a Security Group

You can also use the **NBS Security** view to add a group to a domain.

To add a group to a domain:

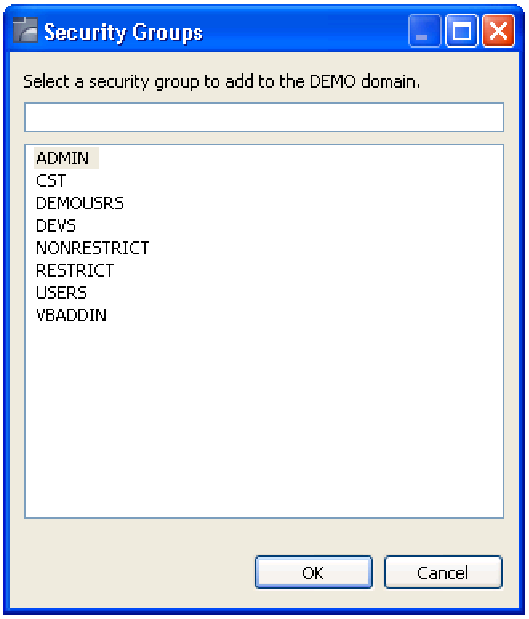
1 Open the context menu for one of the groups listed in the **NBS Security** view.

2 Select **Add group**.

Or:

Select one of the groups and select the toolbar option.

The **Security Groups** window is displayed. For example:



Select a security group to add to the DEMO domain.

ADMIN
CST
DEMOUSRS
DEVS
NONRESTRICT
RESTRICT
USERS
VBADDIN

3 Select the group you want to add.

4 Select **OK**.

The group is added to the domain.

Delete a Security Group

You can also use the **NBS Security** view to remove a group from a domain.

▶ To delete (remove) a group from a domain:

1. Open the context menu for the group in the **NBS Security** view.
2. Select **Delete group**.

   The group is removed from the domain.