NaturalONE

Business Services

Version 9.3.4

October 2026

ADABAS & NATURAL

This document applies to NaturalONE Version 9.3.4 and all subsequent releases.

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

Copyright © 2009-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Document ID: ONE-SERVICES-DOC-934-20261004

Table of Contents

Preface v
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 Introduction 5
About Natural Business Services 6
About Service Development 6
Managing Business Services with NaturalONE 7
Prerequisites 8
3 Getting Started 9
Demo Application 10
Downloading a Business Service 10
Creating a Business Service Using the Wizard 12
Folder Structure in the Workspace 13
Editors 14
4 Creating New Business Services 15
Using the Business Service Wizard 16
Next Steps 19
File Name Format for Business Services 20
Uploading the Business Service Resources 20
5 Creating New Domains and Steplib Chain Files 21
General Information 22
Creating a New Domain 22
Creating a New Steplib Chain File 24
6 Using the Editors 27
General Information 28
Using the Business Service Editor 28
Using the Domain Editor 31
Using the Steplib Editor 33
Using the Dependencies View 36
7 Working with Business Service Resources on the Server 37
Downloading Business Service Resources from a Server 38
Managing Business Service Resources Directly on the Server 40
8 Generating Java Clients and Web Services 41
Using the Generate Java Client Wizard 42
Generated Folder Structure in the Workspace 46
Setting the nbs.properties File to Run a Dynamic RPC Java Class 48
Testing the Java Class 49
Writing an Application to Use the Generated Java Class 50
9 Defining Security for a Domain 53
General Information 54
Showing the NBS Security View 54

Showing the Security Groups for a Domain 54
Changing the Settings for a Security Group 57
Adding Security Groups 58
Removing a Security Group 59
Updating the Security Settings on the Server 60
10 Setting the Preferences for Business Services 61
Showing the Preferences for Business Services 62
Business Services 63
Installation 65

Preface

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

Introduction	About the Natural Business Services (NBS) product on the server and its client. Prerequisites on the NBS server.
Getting Started	About the demo application which can be downloaded from the NBS server. An overview on how to download and create business services. Folder structure in the workspace, and the available editors.
Creating New Business Services	How to generate a business services together with its supporting resources (domains and steplib chain files). Format of the file name for a business service. Information on the business service resources that need to be uploaded.
Creating New Domains and Steplib Chain Files	How to create domains and steplib chain files separately.
Using the Editors	How to use the business service editor, the domain editor, and the steplib editor.
Working With Business Service Resources on the Server	How to download business service resources from an NBS server. How to copy, paste, move and delete business service resources directly on a server.
Generating Java Clients and Web Services	How to generate Java clients and web services for the business services using a wizard.
Defining Security for a Domain	How to view and change the security settings for the domains that are available locally and on the server.
Setting the Preferences for Business Services	Information on the preferences that can be set for the business services.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction

- About Natural Business Services 6
- About Service Development 6
- Managing Business Services with NaturalONE 7
- Prerequisites 8

About Natural Business Services

NaturalONE - or more specifically, the Service Development component of NaturalONE - is the client for the Natural Business Services (NBS) product which runs on the server. This server product also includes Natural Construct.

When the Service Development component of NaturalONE has been installed, you can generate and maintain business services on the client, and you can upload the generated business service resources to your Natural Business Services environment on the server.

Before you read this documentation, it is recommended that you make yourself familiar with the Natural Business Services documentation, which is available separately. The most up-to-date Natural Business Services documentation is always available at ***https://documentation.software-ag.com/natural/nbs.htm***.

About Service Development

The Service Development component provides code generators that can be used to generate Natural subprograms for your business services. This is explained in the *Code Generation* documentation, in the section *Using the Code Generation Component*.

To test your business services and associated Natural subprograms, you have to use the Application Testing component of NaturalONE. This is explained in the *Application Testing* documentation.

The Service Development component is always installed together with the Application Testing component of NaturalONE. Full service Development functionality also requires the EntireX Designer plugin. See *Installation* > *Installing NaturalONE* > *Installing EntireX*

The Service Development component is always installed together with the Natural Construct component of NaturalONE, which provides access to Natural Construct on the server. You can use the Natural Construct component to generate code using Construct models. This is explained in the *Code Generation* documentation, in the section *Using Natural Construct*.

Managing Business Services with NaturalONE

Each business service combines a group of methods related to a common business entity, such as a customer or order. Processing for the methods is supplied by either existing or wizard-generated Natural subprograms.

The Service Development component of NaturalONE provides a wizard for generating business services. But you can also download your existing business services from a Natural Business Services environment on a mainframe or Linux server.

This wizard generates all default resources that are required by the business service. These include:

- **Domain**
  Domains are used to group related business services. Each business service is assigned to a domain. For each domain, you can define specific security permissions.
- **Steplib Chain**
  A steplib chain defines the libraries in which the Natural objects for a business service are stored. The libraries are accessed in the order in which they are listed in the steplib chain. By default, all business services in a domain are accessed using the same steplib chain.
- **Methods**
  The wizard allows you to generate a specific subprogram type (for example, a subprogram that provides browse functionality for a Natural object) or no subprogram at all. The type of method that is generated depends on the subprogram type that you select in the wizard.

After you have generated or download a business service, you can generate Java clients and web services from your business service.

Note: For information on how to test your business services, see the documentation for the *Application Testing* component of NaturalONE.

Prerequisites

Server

To use the Service Development component of NaturalONE, a Natural Business Services (NBS) environment is required on the server. For information on installing and configuring a Natural Business Services environment on a server, see the Natural Business Services documentation.

Although business services can be generated into a project connected to the local Natural runtime environment, they cannot be executed in the local Construct runtime environment. You have to upload the business services to the server so that they can be tested with Application Testing component of NaturalONE.

If you want to make use of the latest server functionality (for example, the Object-Maint-Enhanced-Subp model), Natural Business Services Version 8.2 is required on the server.

Note: If you want to test business services and subprograms, see NaturalONE's *Application Testing* documentation for information on the prerequisites.

Client

If the Service Development component of NaturalONE has not yet been installed, use the Installer to install it.

The Service Development component is always installed together with the Application Testing component and the Natural Construct component of NaturalONE.

3 Getting Started

- Demo Application 10
- Downloading a Business Service 10
- Creating a Business Service Using the Wizard 12
- Folder Structure in the Workspace 13
- Editors 14

Demo Application

Natural Business Services on the server provides sample business services. If you are new to business services, it is recommended that you download the following items from your Natural Business Services (NBS) server, and then make yourself familiar with the programming logic of the downloaded files:

- `DEMO` steplib
- `DEMO` domain
- `SYSBIZDE` system library
- Sample DDMs where the names start with `NSCT` (for example, `NCST-CUSTOMER`) in the `SYSTEM` library

See also the information on the supplied demo applications in *Getting Started with Natural Business Services*, which is part of the Natural Business Services documentation.

The topics below provide a quick overview of how to download or create a business service. See the remainder of this documentation for more detailed information.

If you are new to NaturalONE, you can find more detailed information on the standard NaturalONE functionality (such as creating projects and libraries, or mapping server environments) in *Using NaturalONE*. See also the references below.

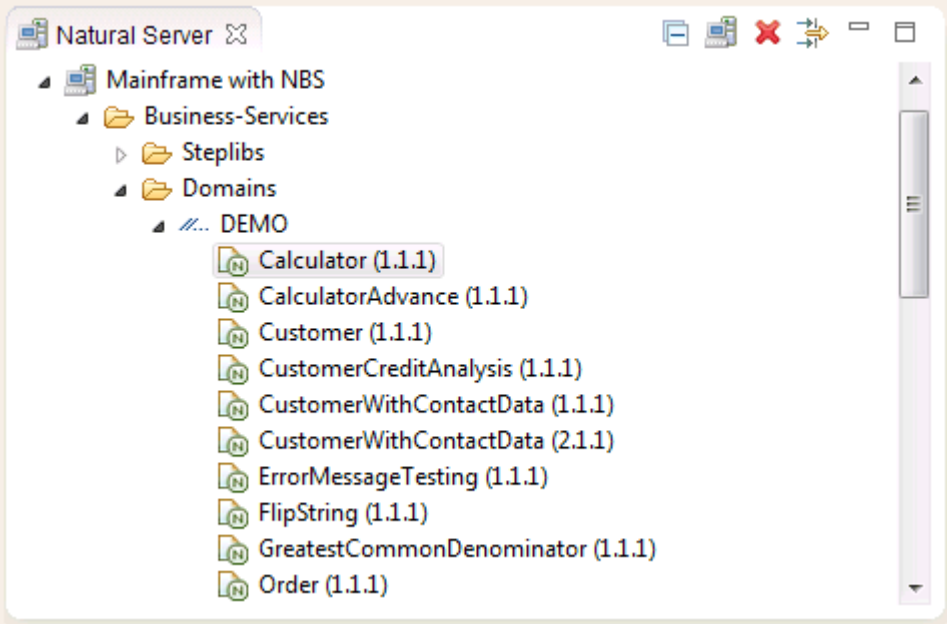
Downloading a Business Service

Business services have to be stored in Natural projects. Other than with the regular Natural objects, it is not possible to download business services into a new project. If you do not want to download the business services into one of your existing projects, you have to create a new project first.

To download a business services

1. In the **Natural Server** view, map an environment in which Natural Business Services has been installed. See also *Mapping a Natural Environment* in *Using NaturalONE*.
2. In the **Project Explorer** view, create a Natural project which is associated with a Natural Business Services (NBS) server. Since you have mapped this environment in the previous step, you can simply select this server from the second page of the project wizard. At this point, you need not specify any more options in the project wizard. See also *Creating a New Project Using a Wizard* in *Using NaturalONE*.
3. Go back to the **Natural Server** view and expand the node for the mapped environment.

A top-level node with the name **Business-Services** is shown if this is an environment in which Natural Business Services has been installed. To view the currently defined domains, expand the **Domains** node. To view the business services in a domain, expand the node for a domain. Example:



By default, the **Business-Services** node is only shown when an installation of Natural Business Services can be detected on the server. This behavior can be changed in the Business Services preferences. See also ***Installation*** in *Setting the Preferences for Business Services*.

4 Select a business service, invoke the context menu, choose the **Add to Existing Project** command. Select the Natural project in the resulting dialog box.

The selected business service is downloaded into the specified project. By default, all required resources which belong to a business service are automatically downloaded. This also includes the related subprograms. This behavior can be changed in the Business Service preferences. See also ***Business Services*** in *Setting the Preferences for Business Services*.

If the Construct runtime project does not yet exist in your workspace, you are asked whether you want to add it to your workspace. You can then choose the **Yes** button to create a new project with the name "ConstructRuntime". This project contains, for example, the error message texts stored in the `CSTMSG` library or the screen prompt texts stored in the `CSTLDA` library. The modules in this project will eliminate compile and parsing errors caused by missing Natural Construct resources. See the *Code Generation* documentation for more detailed information on the Construct runtime project.

5 To find out which resources are used by a downloaded business service, go to the **Project Explorer** view, double-click the downloaded business service to open it in the business service editor, and then check the resources listed in the **Dependencies** view. If resources are listed as "<Unknown>" in this view, double-click each unknown resource and confirm to download

it from the server. Or manually download the resources from the mapped Natural Business Services environment. Afterwards, make sure that the library containing these resources is defined in your steplib chain file. See also ***Using the Steplib Editor***.

6 In **Project Explorer** view, invoke the context menu for the business service and choose **Testing > Test Business Service**. If any items are still missing, you can view the corresponding messages in the **Error Log** view. If the required items can be found, the test editor appears. See the *Application Testing* documentation for detailed information on how to use the test editor.

Creating a Business Service Using the Wizard

Business services have to be stored in Natural projects. If you do not want to create the business service in one of your existing projects, you have to create a new project first.

To create a business service

1 In the **Project Explorer** view, create a Natural project which is associated with a Natural Business Services (NBS) server. If you have already mapped this environment previously, you can simply select this server from the second page of the project wizard. At this point, you need not specify any more options in the project wizard. See also *Creating a New Project Using a Wizard* in *Using NaturalONE*.

2 All Natural objects that are required by your business service need to be stored in a Natural library. Therefore, create a Natural library in the new project. See also *Creating Libraries* in *Using NaturalONE*.

By default, the business service wizard does not generate subprograms. In this case, you have to program the entire logic of your application manually. However, you can specify that a specific type of subprogram is to be generated (for example, an Object Browse or Object Maint subprogram type). In such a case, you have to specify the Natural library into which the subprogram and other related resources are to be generated.

If you intend to generate an Object Browse or Object Maint subprogram type, you also have to specify a DDM. Make sure that this DDM exists in your project before you invoke the wizard.

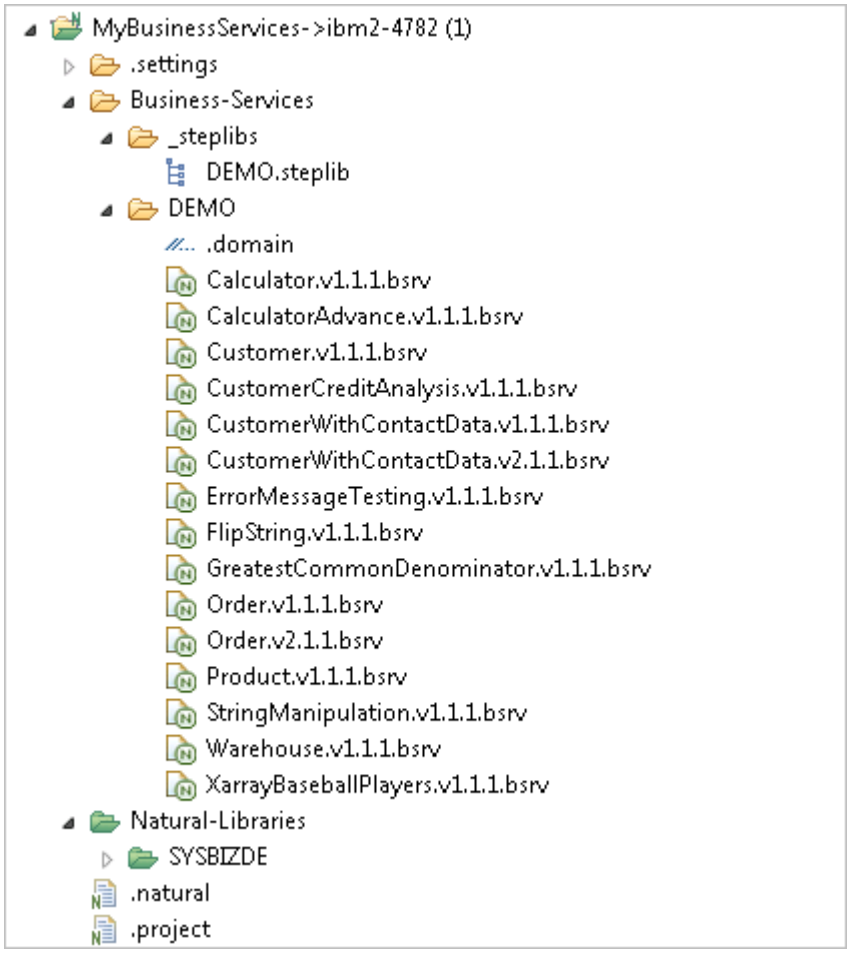
Note: If you select to generate a subprogram, additional pages are shown in the wizard that are not explained in this documentation. See the *Code Generation* documentation for detailed information on these subprogram types.

3 Select the project in which you want to create the business service, invoke the context menu and choose **Business Services > New Business Service**. Proceed as described in ***Creating New Business Services*** later in this documentation.

Note: The **Business Services** context menu is available when the Service Development component of NaturalONE has been installed.

Folder Structure in the Workspace

In a Natural project, all business services are stored in a folder named **Business-Services**. This folder cannot be renamed. For example, when you have downloaded the `DEMO` domain and the `SYSBIZDE` library from the server, your project may look as follows:



The **Business-Services** folder contains a subfolder named **_steplibs** and one or more domain folders, where each folder has the name of the corresponding domain (the above example shows the domain folder with the name **DEMO**).

- The **_steplibs** folder cannot be renamed. It contains one or more steplib chains. These are the files with the extension *.steplib*. Each steplib chain file defines the steplibs that can be used by the business services. The file *DEMO.steplib* in the above example defines `SYSBIZDE` as the steplib.

- Each domain folder contains the following:
  - A file with the name *.domain*, which defines the steplib chain to be used. This file cannot be renamed.

    As a rule, each domain has a steplib chain file with the same name. For example, the DEMO domain has a steplib chain file with the name *DEMO.steplib*.
  - One or more business services. These are the files with the extension *.bsrv*. Each *.bsrv* file defines the Natural subprograms which implement the service methods. These subprograms must be located in one of the defined steplibs.

Important: Even though NaturalONE allows you to rename some of the above-mentioned folders and files, it is recommended that you do not rename them after you have defined the business logic.

Editors

You can view and change the business service resources in the **Business-Services** folder with the following editors which are provided with NaturalONE:

- **Steplib Editor**
  You invoke this editor by double-clicking a file with the extension *.steplib*. This corresponds to the context menu command **Open With > Steplib Editor**.
- **Business Service Editor**
  You invoke this editor by double-clicking a file with the extension *.bsrv*. This corresponds to the context menu command **Open With > Business Service Editor**.
- **Domain Editor**
  You invoke this editor by double-clicking a file with the name *.domain* file. This corresponds to the context menu command **Open With > Domain Editor**.

Each editor provides two tabs. On the first tab, which is always shown by default, you can change the information if required. The second tab is the **XML** tab, which is read-only and can only be used to view the underlying XML.

Caution: Always use the editors provided with NaturalONE to edit the above-mentioned business service resources. Manually editing these files with different editors can potentially cause problems.

For more detailed information, see ***Using the Editors*** later in this documentation.

4 Creating New Business Services

- Using the Business Service Wizard 16
- Next Steps 19
- File Name Format for Business Services 20
- Uploading the Business Service Resources 20

Using the Business Service Wizard

With NaturalONE, it is only possible to create business services locally, in a Natural project. It is not possible to create them directly on the NBS server.

You create business services using a wizard. The business service wizard generates many resources required by the service. These include the business service file, the domain folder and *.domain* file, and the steplib chain file.

By default, the wizard checks for existing business services on the Natural server and displays a warning message if they already exist. It generates the supporting resources (domains and steplib chain files) if they do not yet exist locally. This default behavior can be modified in the Business Service preferences. See ***Business Services*** in *Setting the Preferences for Business Services*.

To use the business service wizard

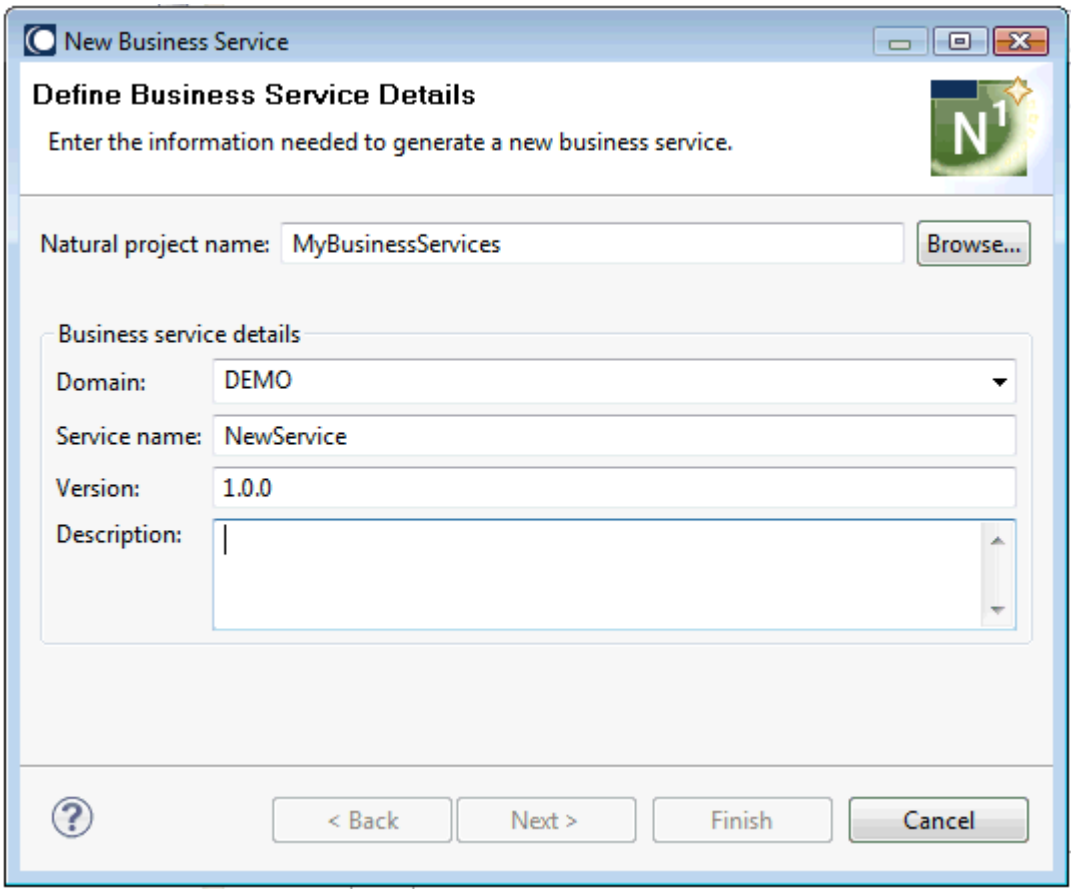
1 In the **Project Explorer** view, select the Natural project in which you want to create the new business service.

2 Invoke the context menu and choose **Business Services > New Business Service**.

   Or:

   From the **File** menu, choose **New > Other**. In the resulting dialog box, expand **Software AG > Business Services**, select **Business Service** and choose the **Next** button.

   The **New Business Service** dialog box appears.



3 Specify the following information:

Option	Description
Natural project name	The project in which the business service is to be created. If you have selected a project before invoking this dialog box, the project name is automatically shown in this text box.
Domain	If your project already contains one or more domains, a domain name is provided. If you do not want to use the provided domain, you can either type the name of a new domain that is to be created, or - if more domains are defined in your project - you can select the name of a different domain from the drop-down list box.
Service Name	The default name "NewService" is provided. Type an appropriate name for the new service.
Version	The default version number "1.0.0" is provided. Type an appropriate version number for the new service. The version number consists of three parts that are separated by dots. It must match the following format:

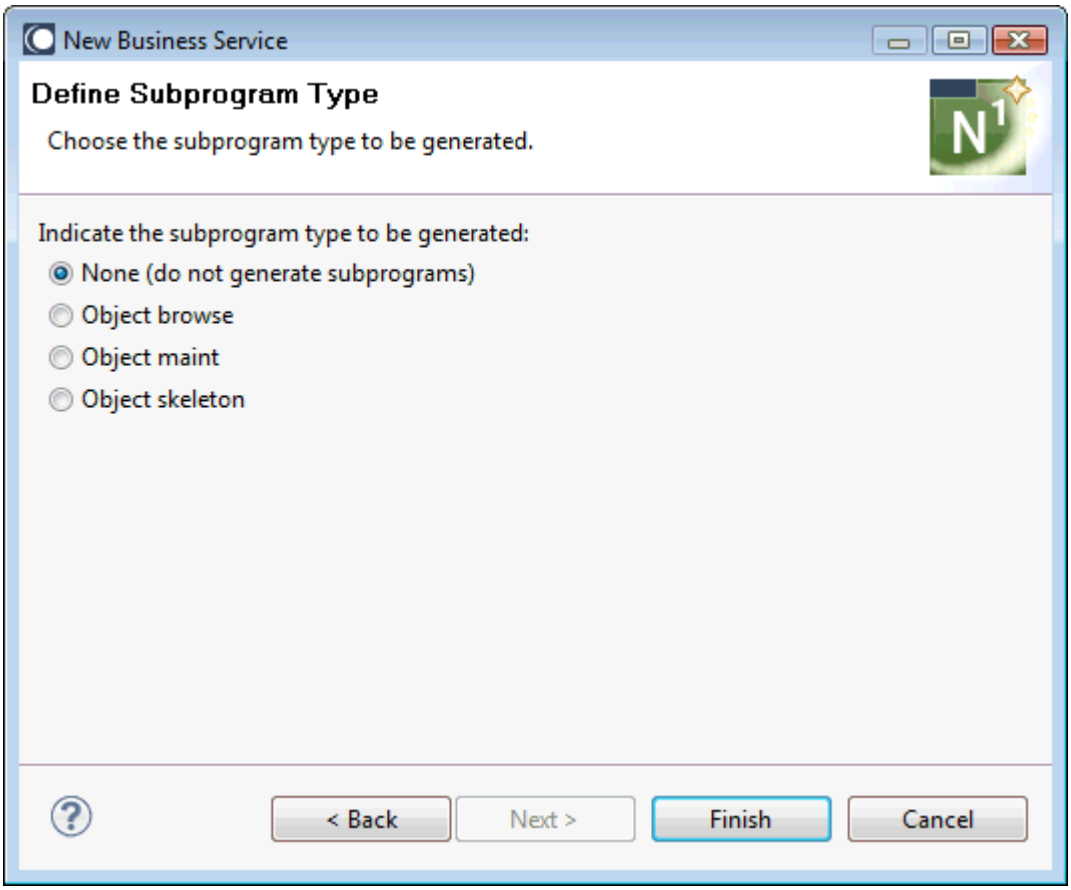
Option	Description
	nn.nn.nn
Description	Type a brief description of the service.

4 Choose the **Finish** button if you do *not* want to generate a subprogram for your business service.

Or:

Choose the **Next** button if you want to generate a subprogram for your business service.

When you choose the **Next** button, the following dialog box appears.



5 If you want to generate a subprogram, select one of the following options:

Option	Description
Object browse	Generates a subprogram that provides browse functionality for a Natural object. Parameter data areas are also generated. You have to specify the name of an existing DDM for this purpose.
Object maint	Generates a subprogram that maintains complex data objects and updates all entities within an object. Local data areas and parameter data areas are also generated. You have to specify the name of an existing DDM for this purpose.
Object skeleton	Generates a skeleton subprogram which uses the input/output parameters and default methods that you specify in the wizard.

If you select one of the above options, additional pages are shown in the wizard that are not explained in this documentation. For detailed information on the above modules and the options that you have to specify in the wizard, see the *Code Generation* documentation.

Note: The above modules can also be generated when you invoke the **Code Generation** context menu for a project.

6 Choose the **Finish** button to generate the business service file and, if specified, the subprograms. If they do not already exist, the associated domain and steplib chain files are also generated.

The newly created business service is automatically shown in the business service editor. For more information (for example, on how to add methods to the business service), see ***Using the Business Service Editor***.

Next Steps

After you have generated a business service using the wizard, you can proceed as described below:

1. Make sure that your domain information is correct. See ***Using the Domain Editor***.
2. Create all libraries that are to be used as steplibs.
3. Make sure that your steplib chain file defines all steplibs that are to be used. See ***Using the Steplib Editor***.
4. If you have not selected to create a subprogram in the business service wizard: create the subprogram that implements the methods that are to be used by your business service.
5. Add all required methods to the business service. See ***Using the Business Service Editor***.
6. Upload all business service resources. See ***Uploading the Business Service Resources***.
7. Test the business service. See NaturalONE's *Application Testing* documentation for detailed information.

File Name Format for Business Services

The name of a business service file is composed of the service name and a version number, separated by a dot. The version number consists of three parts that are also separated by dots. When you rename a business service file, you have to keep this structure. The name must match the following format:

```
name.vnn.nn.nn.bsrv
```

For example, *MyCustomer.v3.2.1.bsrv* or *MyOrder.v1.10.20.bsrv*.

It is recommended that you do not rename a business service file after you have defined the business logic.

Uploading the Business Service Resources

All objects that are addressed by a business service must be stored in an `FNAT` or `FUSER` system file on the server. You use the standard NaturalONE functionality to upload the business service file and all other generated resources (such as the domain file and steplib chain file) to the server.

When a business service is uploaded, it is placed into the NBS repository on the server (`LFILE 135`). During the upload, it is not checked whether the objects that are used by the business service already exist. When the business service is executed, however, it is checked whether all defined subprograms are available.

When the option **Build Natural projects automatically** is enabled in the Natural preferences, the appropriate server is automatically updated each time you save a business service resource in the Eclipse workspace or when you add a new business service resource. When this option is disabled in the Natural preferences, you have to update the server manually. The business service files, domain files and steplib chain files need not be stowed or cataloged on the server. Therefore, you can simply use the **Upload** command.

If you change your steplib definitions after the upload (for example, when you define a new library), you have to make sure that the updated steplib chain file is uploaded to the server.

If you save your data in the repository of a version control system, you have to make sure that this data is available in Natural when the service is executed.

5 Creating New Domains and Steplib Chain Files

- General Information 22
- Creating a New Domain 22
- Creating a New Steplib Chain File 24

General Information

Domains and steplib chain files are automatically generated in your project when you create a business service using the wizard. If you need more domains and steplib chain files, you can generate them as described below.

Creating a New Domain

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 In the **Project Explorer** view, select the Natural project in which you want to create the new domain.

   Or:

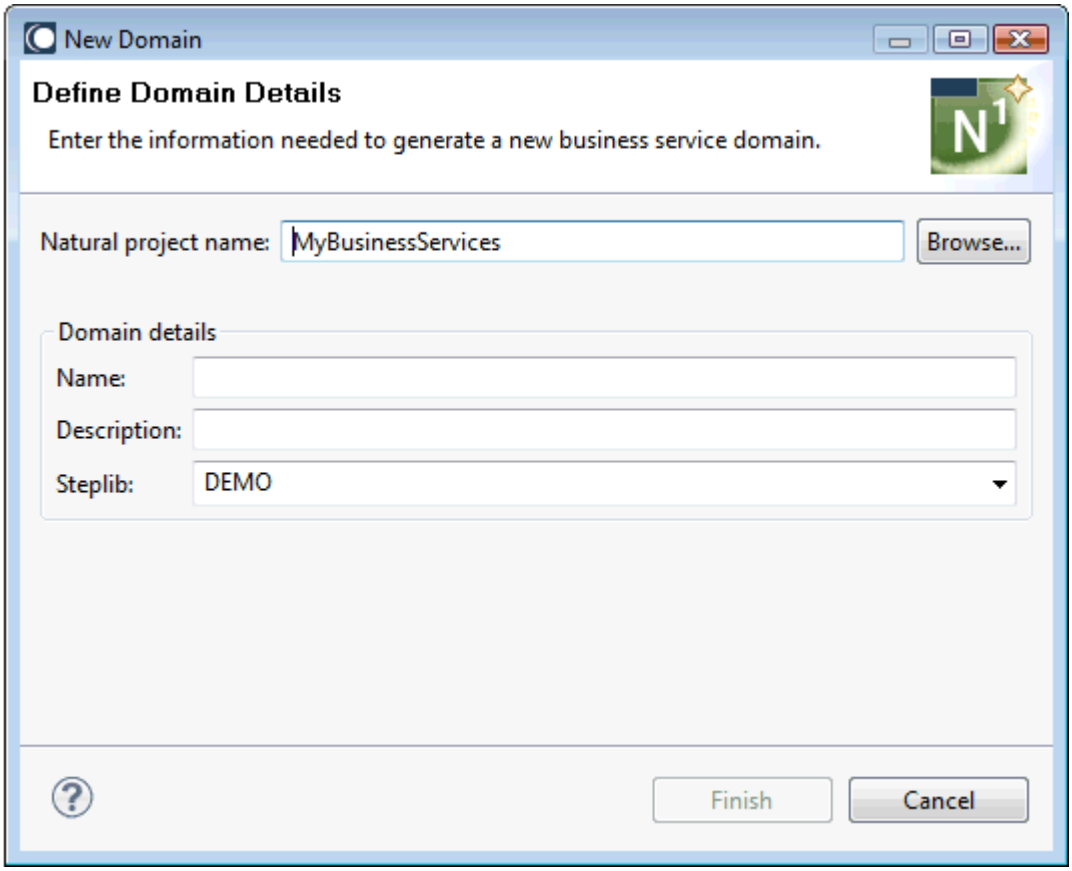
   Select the **Business-Services** node in that project.

2 Invoke the context menu and choose **Business Services > New Domain**.

   Or:

   From the **File** menu, choose **New > Other**. In the resulting dialog box, expand **Software AG > Business Services**, select **Domain** and choose the **Next** button.

   The **New Domain** dialog box appears.



3 Specify the following information:

Option	Description
Natural project name	The project in which the domain is to be created. If you have selected a project before invoking this dialog box, the project name is automatically shown in this text box.
Name	Type the name of the domain. By default, the wizard checks the server connection to verify whether the domain currently exists. If it does, a warning message is shown. You can now decide whether you want to overwrite the content of the existing domain on the server, or whether to enter a different name in order to create a new domain. This default behavior can be modified in the Business Service preferences. See ***Business Services*** in *Setting the Preferences for Business Services*.
Description	Type a brief description of the domain.
Steplib	It is recommended that the steplib chain file has the same name as the domain. If you want to generate a new steplib chain file that has the same name as the domain, type that name here. The drop-down list box provides for selection all steplibs that are currently defined in the project. You can also select an existing steplib chain file.

4 Choose the **Finish** button.

A new domain folder with the specified name is shown within the **Business-Services** node of the project. A file with the name *.domain* has been created in that domain folder. The *.domain* file specifies the steplib chain file to be used. If you have specified the name for a new steplib chain file, this file has been created in the **_steplibs** node.

Creating a New Steplib Chain File

A steplib chain file defines a set of Natural libraries that a Natural RPC server must access to call a subprogram or proxy. The server accesses the libraries in the order they are listed in the steplib chain file.

When you create the steplib chain file as described below and if your environment is protected by Natural Security, you have the advantage that you can specify an existing library in your project from which to copy the steplib definitions.

Keep the following in mind:

- If your project belongs to a secure environment, steplibs can only be defined in the library properties.
- If your project does not belong to a secure environment, steplibs can only be defined in the project properties.

To create a new steplib chain file

1 In the **Project Explorer** view, select the Natural project in which you want to create the new steplib chain file.

Or:

Select the **Business-Services** node in that project.

Or:

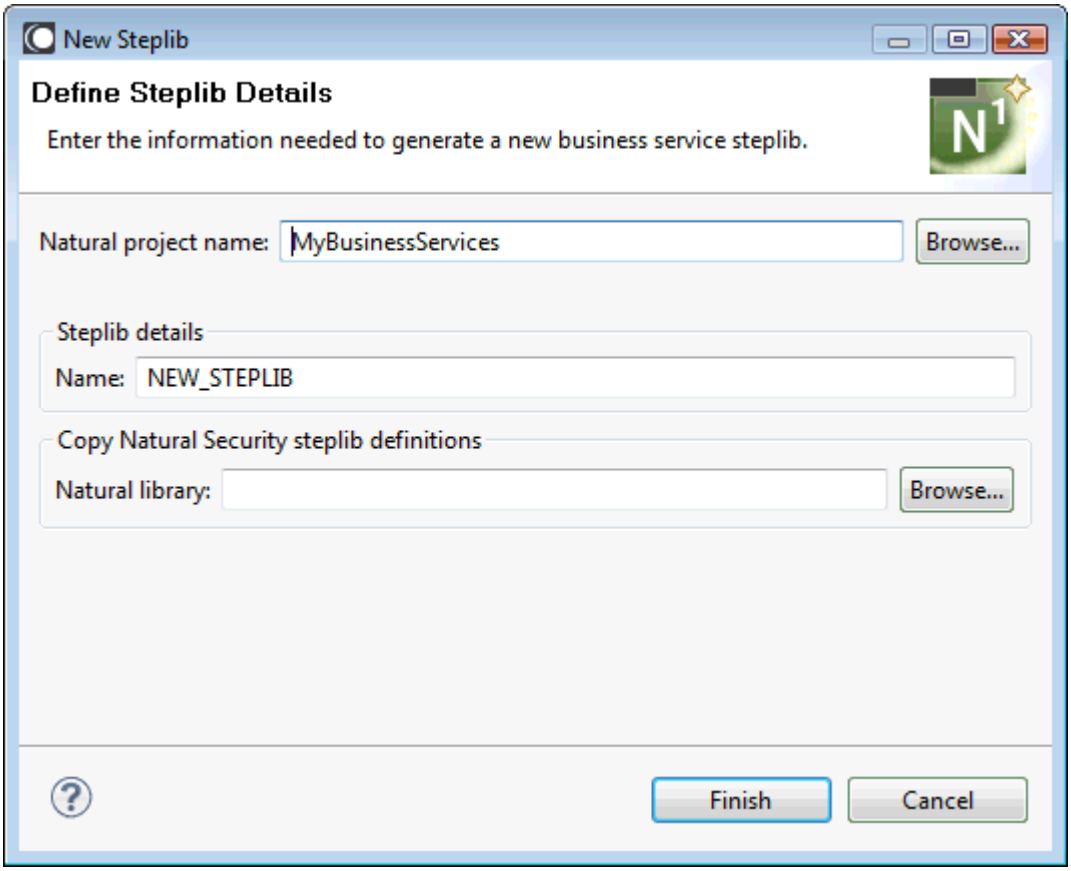
Select the **_steplibs** node in that project.

2 Invoke the context menu and choose **Business Services > New Steplib**.

Or:

From the **File** menu, choose **New > Other**. In the resulting dialog box, expand **Software AG > Business Services**, select **Steplib** and choose the **Next** button.

The **New Steplib** dialog box appears.



3 Specify the following information:

Option	Description
Natural project name	The project in which the steplib chain file is to be created. If you have selected a project before invoking this dialog box, the project name is automatically shown in this text box.
Name	Type the name for a new steplib chain file over the default name.
Natural Library	This option is only available when Natural Security is active in the associated server environment. In this case, steplibs can only be defined in the library properties. If you want to copy the steplib definitions from a specific library of the current project, type the name of that library. Or choose the **Browse** button, to select that library from a dialog box. If you do not specify a library name, the generated steplib chain file will list a steplib library with the same name as the steplib chain file (only the first 8 characters of that name are used). This steplib may not yet exist in your project.

4 Choose the **Finish** button.

The new steplib chain file is created in the **_steplibs** node, and the steplib editor is invoked, showing the definitions for the new steplib chain file. See also ***Using the Steplib Editor***.

6 Using the Editors

- General Information 28
- Using the Business Service Editor 28
- Using the Domain Editor 31
- Using the Steplib Editor 33
- Using the Dependencies View 36

General Information

Specific editors are available for the following business service resources:

- business service files
- domain files
- steplib chain files

Each editor provides two tabs. On the first tab, which is always shown by default, you can change the information if required. The second tab is the **XML** tab, which is read-only and can only be used to view the underlying XML.

Caution: Always use the editors provided with NaturalONE to edit the above-mentioned business service resources. Manually editing these files with different editors can potentially cause problems.

If you want to edit the above-mentioned business service resources, they must be available locally (that is, in the **Project Explorer** view). You cannot edit or view them directly on the server (that is, in the **Natural Server** view). If you want to edit or view the business service resources that are stored on the server, you have to download them first. See *Downloading Business Service Resources from a Server*. After editing the resources, you can return them to the server using NaturalONE's standard functionality (for example, with the **Upload** command).

Using the Business Service Editor

The business service editor is used to view and change the following aspects of the business service definition: subprogram proxy and subprogram used by the business service, description of the business service, and the methods used by the business service.

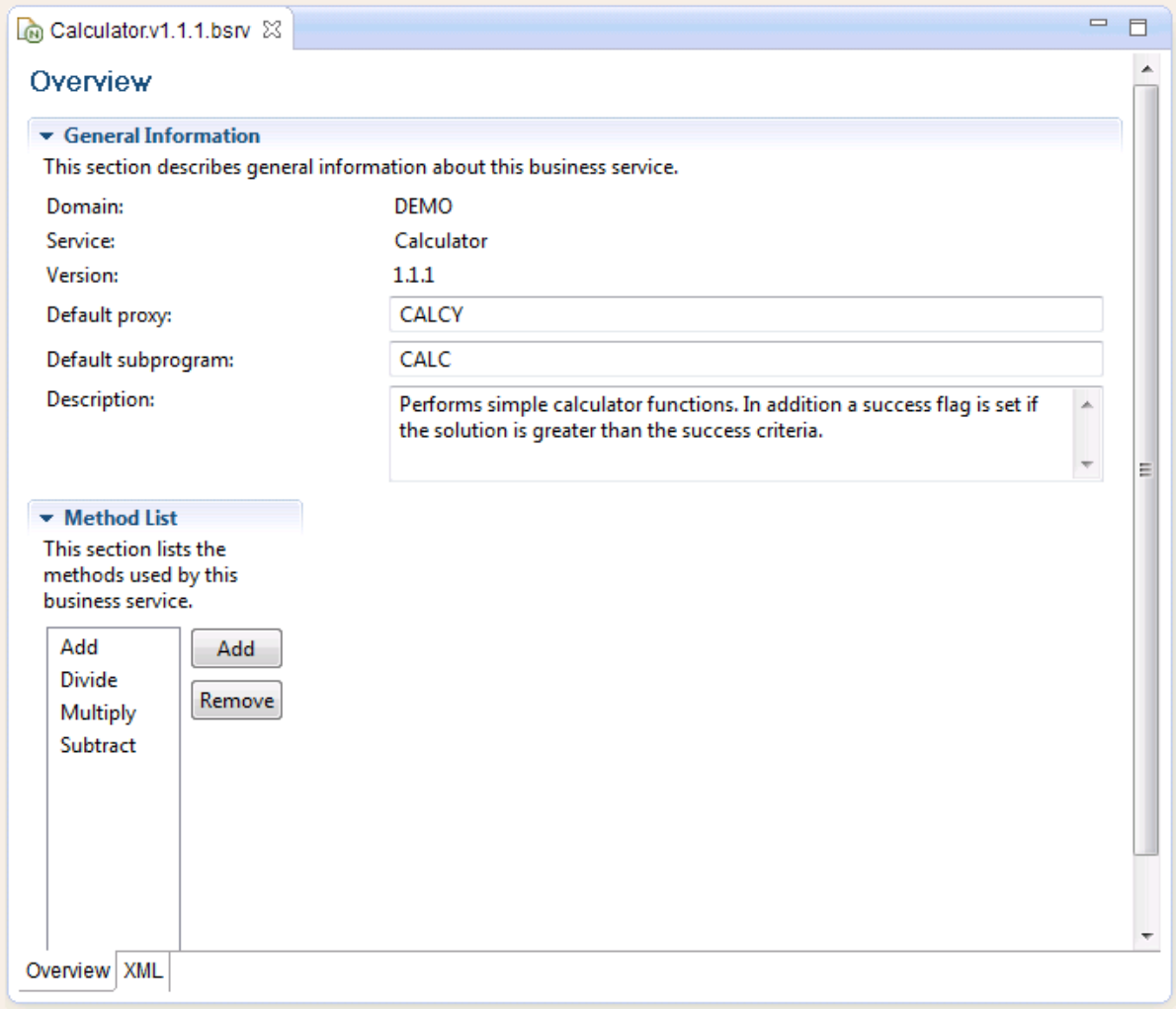
To edit the business service definition

1. In the **Project Explorer** view, expand the **Business-Services** node of your project.
2. Expand a folder which has the name of a domain.
3. Double-click a file with the extension *.bsrv*.

   Or:

   Select a file with the extension *.bsrv*, invoke the context menu and choose **Open With > Business Service Editor**.

   The business service editor appears. Example:



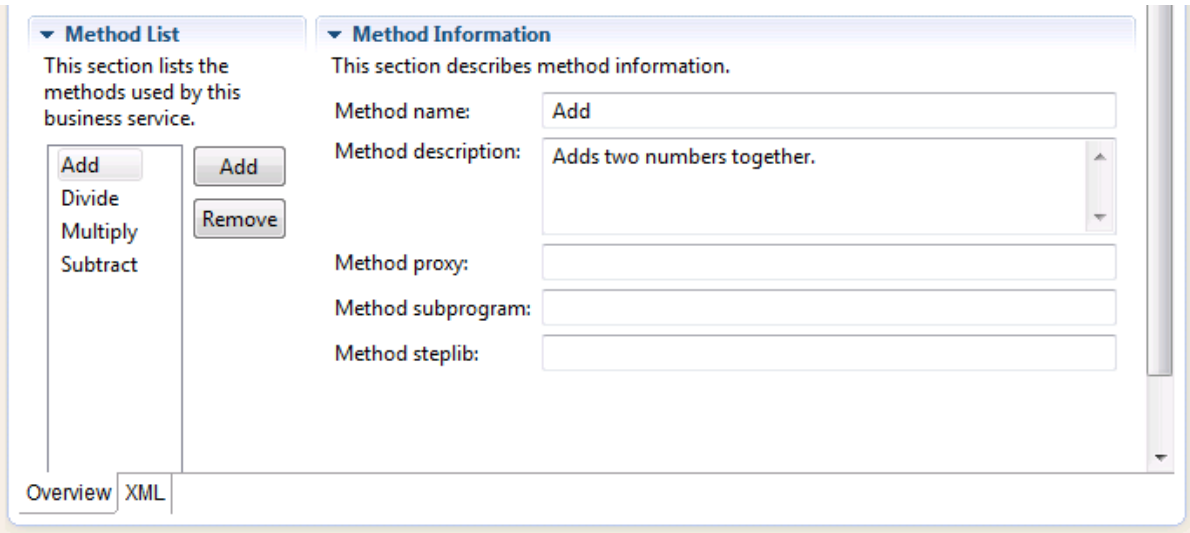
4 You can change the following general information:

Option	Description
Default proxy	The name of the subprogram proxy used for this service. This proxy is used by default if a method that is listed in the editor does not use its own method proxy (see below). Proxies are used for the communication between the client and the server. They invoke Natural business services from a client. Proxies can only be generated using Natural Business Services on the server, for example, by using the Subprogram-Proxy-Client model. See the Natural Business Services documentation for further information.

Option	Description
Default subprogram	The name of the subprogram that implements the service methods. This subprogram is used by default if a method that is listed in the editor does not use its own subprogram (see below). **Important:** You must specify a default subprogram for the service if any of the listed methods do not specify a subprogram.
Description	A brief description of the business service.

5 To view or change the information that is defined for a method, select the method in the method list.

The lower part of the editor now shows additional information. Example:



6 You can change the following method information:

Option	Description
Method name	The name of the method as defined in the subprogram. When you have created a new business service that does not use a subprogram, a method with the name "Default" is initially shown in the method list. After you have created the subprogram which implements your methods, you have to change "Default" to the name of an actual method used in your subprogram.
Method description	A brief description of the method.
Method proxy	The name of the subprogram proxy used for this method. If defined, the method proxy is used instead of the default proxy.
Method subprogram	The name of the subprogram that implements this method. If defined, the method subprogram is used instead of the default subprogram.

Option	Description
Method steplib	The name of a steplib chain file used by this method. You need only specify a method steplib if the default steplib chain file that is defined in the *.domain* file is not sufficient.

7 You can use the following command buttons:

Command Button	Description
Add	Adds a method. You have to specify the required method information (see above).
Remove	Removes the selected method.

8 Save your changes using the standard Eclipse functionality (for example, press CTRL+S)

Using the Domain Editor

The domain editor is used to view and change the description and to define a different steplib chain.

To edit the domain definition

1 In the **Project Explorer** view, expand the **Business-Services** node of your project.

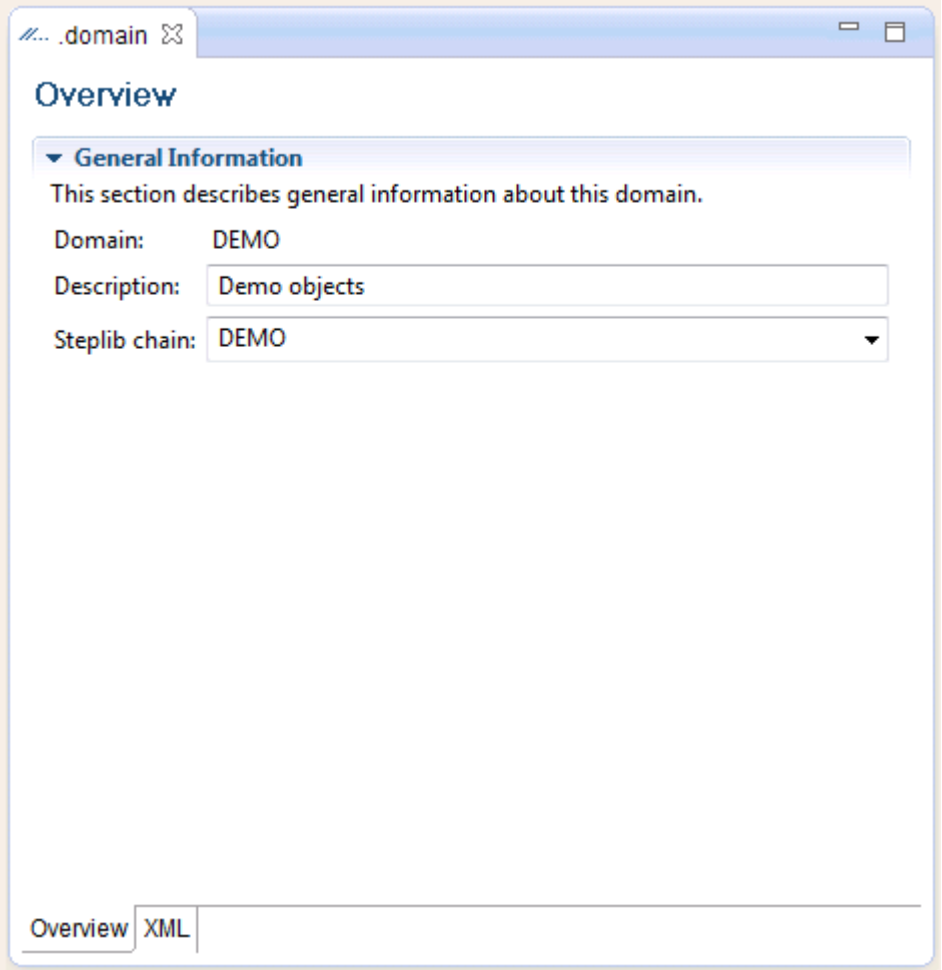
2 Expand a folder which has the name of a domain.

3 Double-click the *.domain* file.

Or:

Select the *.domain* file, invoke the context menu and choose **Open With > Domain Editor**.

The domain editor appears. Example:



4 You can change the following information:

Option	Description
Description	The description of the domain.
Steplib chain	The name of the steplib chain file to be used. This corresponds to a file with the extension *.steplib* in the **_steplibs** node of your project. It is possible to enter the name of steplib chain file that does not yet exist. Keep in mind that there is no plausibility check when editing the domain definition. Such a check is only performed during runtime when the business service is executed.

5 Save your changes using the standard Eclipse functionality (for example, press CTRL+S)

Using the Steplib Editor

The steplib editor is used to view, change and reorder the steplibs in a steplib chain. A steplib contains the subprograms that implement the service methods. You can use any library in your project as a steplib. You can also use libraries from other projects as steplibs.

When a business service is executed, it always accesses information from the server. Therefore, it is important that the business service can find the libraries in the appropriate `FUSER` or `FNAT` system file.

To edit the steplib definition

1 In the **Project Explorer** view, expand the **Business-Services** node of your project.

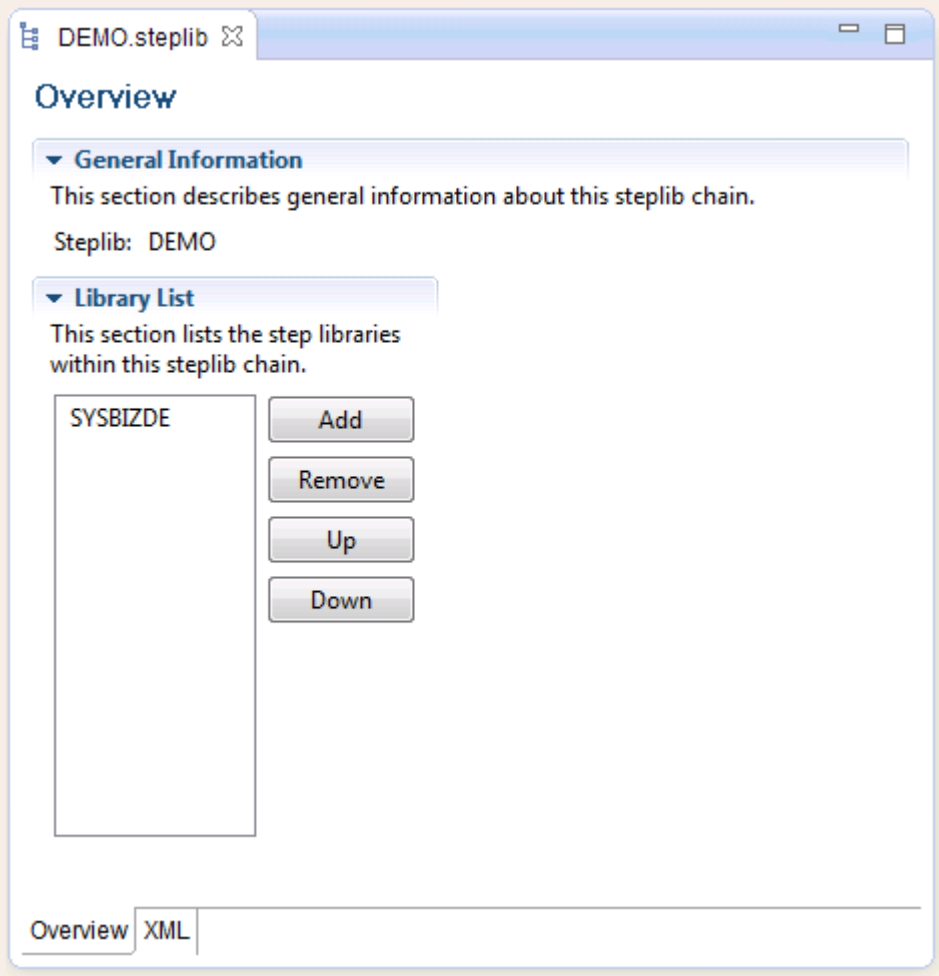
2 Expand the **_steplibs** node.

3 Double-click a file with the extension *.steplib*.

Or:

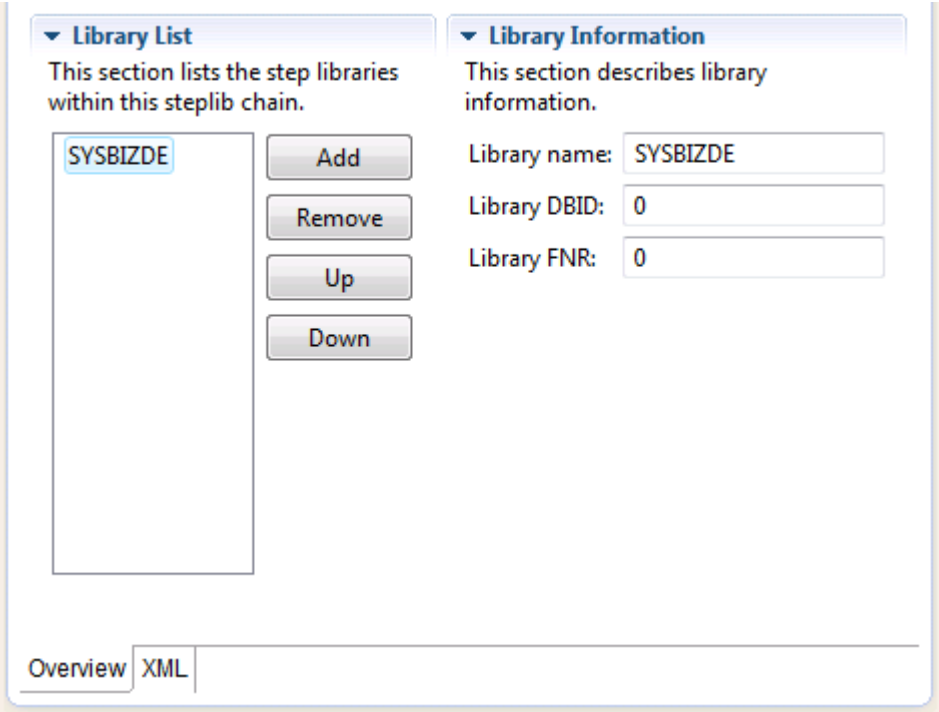
Select a file with the extension *.steplib*, invoke the context menu and choose **Open With > Steplib Editor**.

The steplib editor appears. Example:



4 To view or change the information that is defined for a steplib, select the library in the library list.

The lower part of the editor now shows additional information. Example:



5 You can change the following library information:

Option	Description
Library name	The name of the library.
Library DBID	The ID of the database which contains this library. Normally, you need not define a DBID and FNR because all required objects are stored on the server which is defined in the project properties. This is only required in advanced configurations when uploading to the server (for example, when you have defined a business service with several methods where the corresponding subprograms are stored in different libraries, and the libraries are stored in different databases and files).
Library FNR	The number of the database file which contains this library. Only required in advanced configurations when uploading to the server.

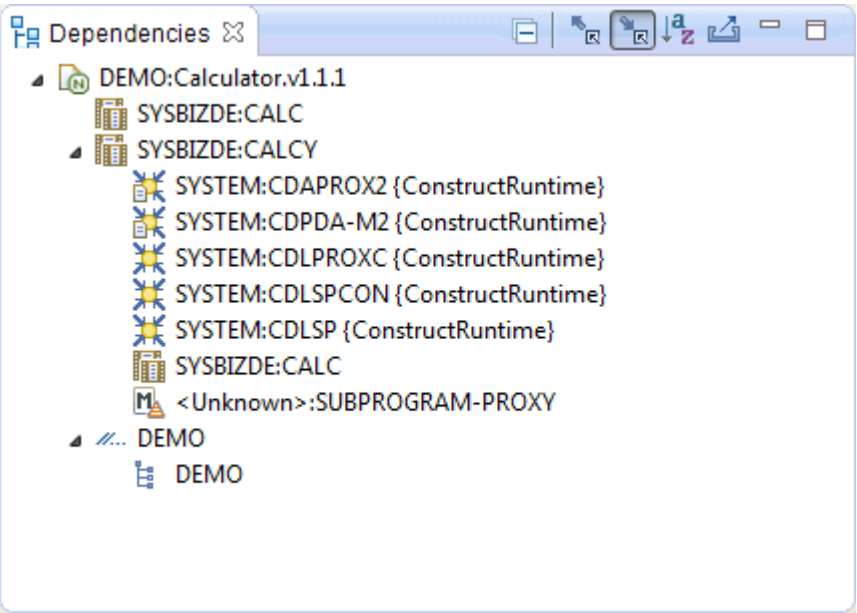
6 You can use the following command buttons:

Command Button	Description
Add	Adds a library to the steplib chain. You have to specify the required library information (see above).
Remove	Removes the selected library from the steplib chain.
Up	Moves the selected library up one position in the library list.
Down	Moves the selected library down one position in the library list.

7 Save your changes using the standard Eclipse functionality (for example, press CTRL+S)

Using the Dependencies View

When a business services resource (for example, a business service file) is shown in the active editor, the **Dependencies** view shows the dependencies between that resource and other business services resources and/or Natural resources.



When a supporting resource (domain or steplib) cannot be found locally using the project steplib chain and project references, "<Unknown>" is displayed with the name of the resource. You can download the unknown object using the **Open** command from the context menu or by double-clicking the object.

If an unknown module is not shipped with the Construct runtime project, either manually download it from the server or create it locally. If the module is shipped with the Construct runtime project, add the project. See the *Code Generation* documentation for more detailed information on the Construct runtime project.

Object models such as `SUBPROGRAM-PROXY` are marked with the following icon: . These models cannot be downloaded using the **Open** command in the context menu (or by double-clicking). Normally, you would not download such a model unless you want to maintain the model yourself.

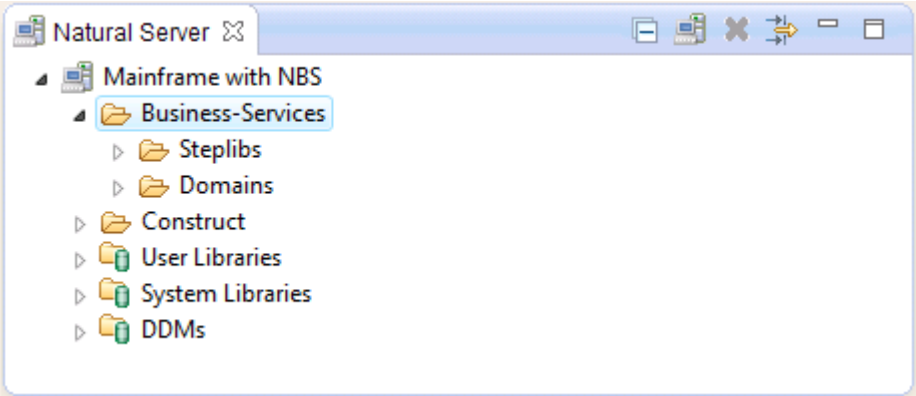
For more information on the **Dependencies** view (for example, on the icons in the local toolbar), see the description of the source editor in *Using NaturalONE*.

7 Working with Business Service Resources on the Server

- Downloading Business Service Resources from a Server 38
- Managing Business Service Resources Directly on the Server 40

Downloading Business Service Resources from a Server

Using the **Natural Server** view, you can download existing business service resources from a Natural Business Services (NBS) server into a Natural project. When Natural Business Services has been installed in the server environment, a **Business-Services** node is shown for the mapped environment, containing further subnodes for steplibs and domains.



By default, the **Business-Services** node is only shown when an installation of Natural Business Services can be detected on the server. This behavior can be changed in the Business Services preferences. See ***Installation*** in *Setting the Preferences for Business Services*.

To download business service resources

It is only possible to download business service resources into an existing Natural project. They cannot be downloaded into a new Natural project. If you want to download business service resources into a new project, create that project first.

1 In the **Natural Server** view, map the required NBS server. See also *Mapping a Natural Environment* in *Using NaturalONE*.

   When mapping to an NBS server, you have to enter a user ID and password. These credentials must be defined to Natural Business Services security on the server, and they must have the appropriate permissions.

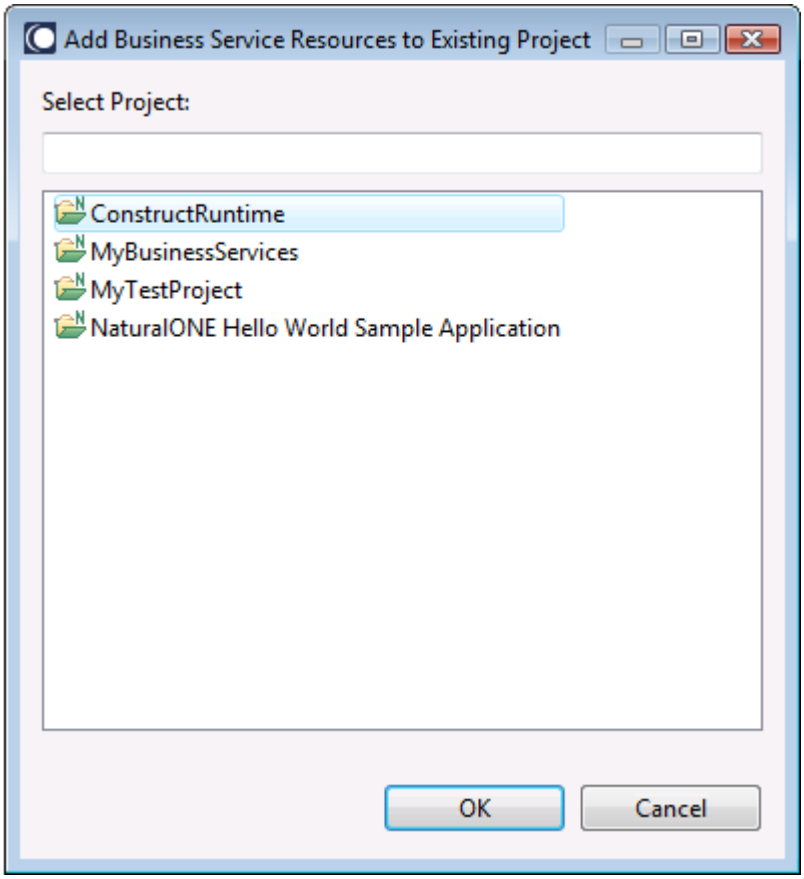
2 In the **Natural Server** view, select the **Business-Services** node to download the entire contents of that node.

   Or:

   If you only want to download selected items, select the **Steplibs** folder, the **Domains** folder, or any single items in these folders.

3 Invoke the context menu and choose **Add to Existing Project**.

The **Add Business Service Resources to Existing Project** dialog box appears.



4 Select the project into which you want to download the business service resources and choose the **OK** button.

The definitions for all selected services, domains, and steplibs are downloaded into the selected project. In the **Project Explorer** view, you can view them in the **Business-Services** node of the project. If this node does not yet exist, it is automatically created during the download.

Notes:

1. By default, any supporting resources (domains and steplib chain files) and related subprograms that were not selected are automatically downloaded (for example, the steplib chain file for a selected domain is automatically downloaded). You can change this behavior in the Business Services preferences. See also ***Business Services*** in *Setting the Preferences for Business Services*.
2. If the Construct runtime project does not yet exist in your workspace, you are asked whether you want to add it to your workspace. You can then choose the **Yes** button to create a new project with the name "ConstructRuntime". This project contains, for example, the error message texts stored in the `CSTMSG` library or the screen prompt texts stored in

the `CSTLDA` library. The modules in this project will eliminate compile and parsing errors caused by missing Natural Construct resources. See the *Code Generation* documentation for more detailed information on the Construct runtime project.

3. If you want to check whether any steplibs are missing in your project after the download, you can open the file which defines the steplib chain (that is, the file with the extension *.steplib*). When the library list contains any steplibs which are not available in your project, you should download them from the server. See ***Using the Steplib Editor*** for further information.

Managing Business Service Resources Directly on the Server

In the **Natural Server** view, you use the following context menu commands with your business service resources:

Command	Description
Copy	Copies the selected business service files to the clipboard so that they can be used with a subsequent **Paste** command.
Paste	Pastes the copied business service files into a different domain of the same mapped environment. It is not possible to paste the business service files into any other node which is not a domain node of the *same* mapped environment.
Move	Moves the selected business service files from the current domain into a different domain of the same mapped environment. The **Move Objects** dialog box appears in which you have to select a destination. Even though the **Move Objects** dialog box may also list other mapped environments, it is only possible to move the selected business service files into a different domain of the *same* mapped environment.
Delete	Deletes the selected business service files, domain folders or steplib chain files. You have to confirm this action. **Caution:** When you delete a domain folder, you also delete all business services contained in this domain.

Note: Domain folders and steplib chain files cannot be copied, pasted or moved.

8 Generating Java Clients and Web Services

- Using the Generate Java Client Wizard 42
- Generated Folder Structure in the Workspace 46
- Setting the nbs.properties File to Run a Dynamic RPC Java Class 48
- Testing the Java Class 49
- Writing an Application to Use the Generated Java Class 50

Using the Generate Java Client Wizard

After creating business services locally or downloading them from an existing server installation, you can generate Java clients and web services for the business services using the **Generate Java Client** wizard.

Using this wizard, you can

- **Generate Java clients**
  You can develop Java client applications that access Natural Business Services server components via dynamic RPC. The wizard creates Java clients from the business service files in a NaturalONE project and then generates the Java client interface objects used by the Java client to access the server components.
- **Generate web services**
  You can generate web services from the business service files in a NaturalONE project. The generated objects are web service archives (files with the extension *.aar*). They can be deployed in a Web Services Stack runtime environment. Web service client applications can then access these web services and expose business logic implemented by the Natural server components.

To use the Generate Java Client

1. In the **Project Explorer** view, select the business service file (that is, a file with the extension *.bsrv*) from which you want to generate a Java client or web service.
2. Invoke the context menu and choose one of the following commands: **Generate Java Client** or **Generate Web Service**.

   The **Generate Java Client** wizard is shown for both commands. The only difference is that specific options are already preselected depending on the chosen command.
3. Specify all required information as described in the topics below. Some options require additional information. In such a case, you have to use the **Next** button repeatedly to proceed from the first page of the wizard to the last page.
4. When all required information has been provided, choose the **Finish** button.

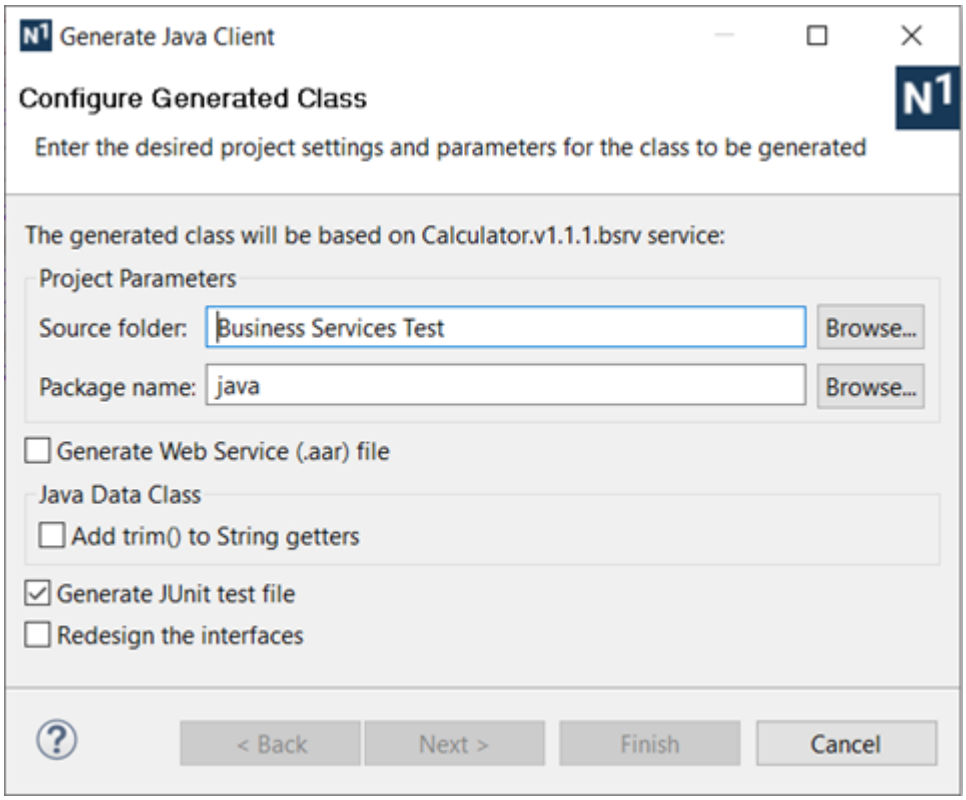
The different pages of the deployment wizard are described below.

- General Generation Settings

- Design Interface for Natural Subprogram

General Generation Settings

On the first page of the wizard, you define the general settings for the generation.



Source folder

The source folder in which to generate the Java class. This folder must already exist. This can be the folder for any type of project (for example, a Java project), or a folder within a project. You can choose the **Browse** button to select the folder from a dialog box.

If you select a project as the source folder, the wizard will automatically create and use `<project-name>`/*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*.

Package name

The name of the Java package. The package does not have to currently exist. You can either type the name for a new Java package, or you can choose the **Browse** button to select an existing package from a dialog box.

Generate Web Service (.aar) file

This option is preselected if you have invoked the wizard using the **Generate Web Service** command.

If selected, the web service file (that is, a file with the extension *.aar*) will be created when the class is generated.

Add trim() to String getters

If selected, a Java trim statement (for example, `string_variable.trim()`) will be added to each String property getter method in the generated Java Data class (`<name>`*Data.java*). The trim statement will return a copy of the String property with leading and trailing whitespace omitted.

Generate JUnit test file

If selected, a JUnit test file is generated which contains test functions that correspond to each method in the business service. For more information on the generated JUnit test file, see ***Testing the Java Class***.

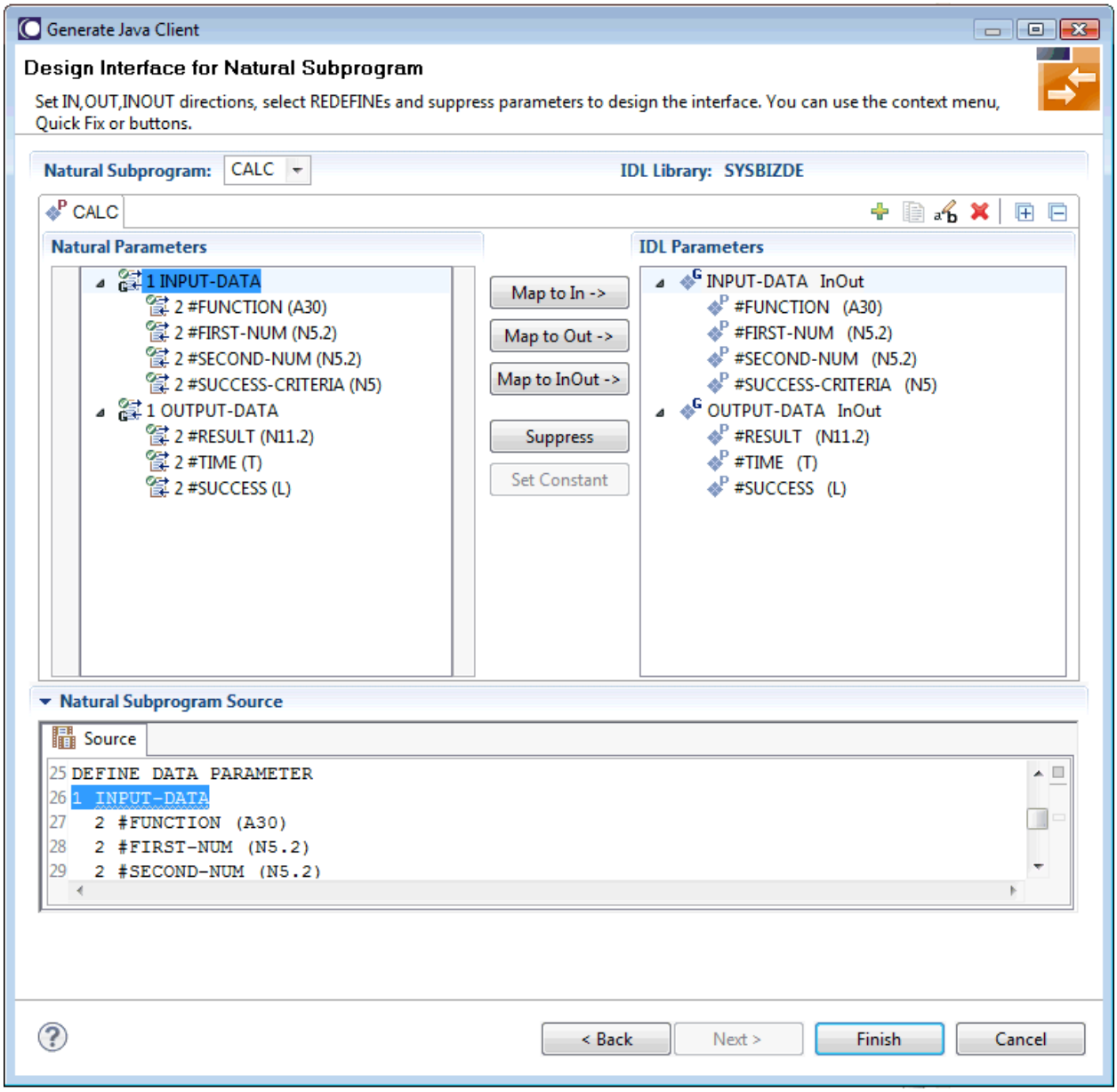
Redesign the interfaces

If selected, the wizard will display an additional page on which you can redesign the interface for Natural subprograms. For example, you can change the default IDL directions or you can select which portions of a redefined field are to be used. See also ***Design Interface for Natural Subprogram***.

Note: When an exception occurs as the result of one of your actions in the wizard, this may happen because required resources cannot be found. Check whether all required resources are located in your workspace and that your steplib chain file lists the required library.

Design Interface for Natural Subprogram

This page is shown when the option **Redesign the interfaces** has been selected in the wizard. You can then redesign the interface generated for a Java client.



The upper left pane on this page shows the parameters of the Natural subprogram to extract from. For each Natural subprogram parameter, you can choose an operation as described in the following table.

Task	Procedure
Change the default IDL directions for the generated Java client	In the upper left pane, select the level 1 parameter that you want to change and choose **Map to In**, **Map to Out** or **Map to InOut**. The operation is reflected in the right pane. Continue to define level 1 parameters as required. Natural parameters that are mapped in the interface are displayed with a green tick.

Task	Procedure
Define which parameters redefined in the Natural PDA are part of the extracted interface	In the upper left pane, select the `REDEFINE` base parameter or any `REDEFINE` path and choose **Map to InOut**.
Suppress unnecessary parameters, which creates a less cluttered IDL client interface and minimizes the amount of data transferred at runtime	In the upper left pane, select the parameter and choose **Suppress**. Natural parameters suppressed in the interface are displayed in italic type.

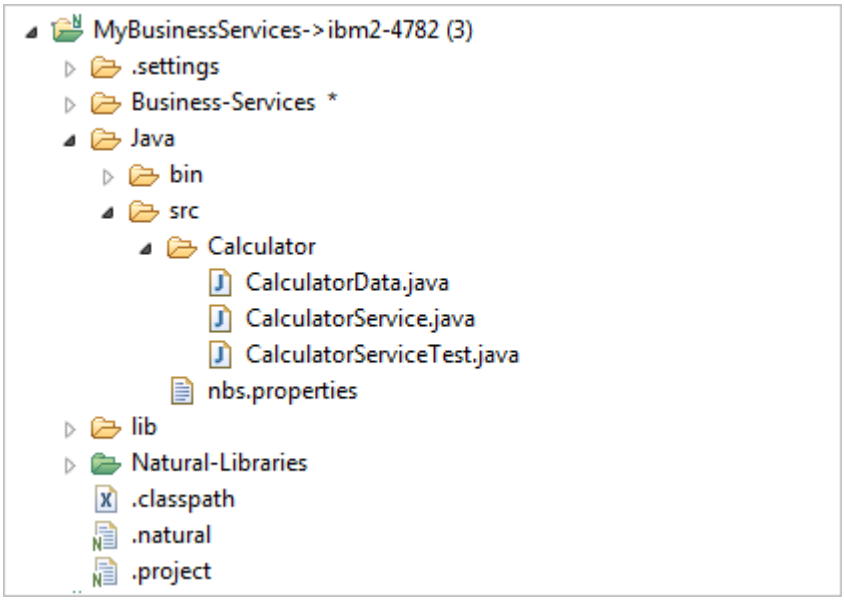
The upper right pane displays the extracted interface (IDL). The IDL directions will be inserted as top-level parameters (level 1) in the interface definition for the Natural subprogram.

The bottom pane is used for reference purposes. It displays the Natural subprogram source and, if available, its PDA sources, each in a separate tab. To reserve more space for viewing the upper panes, you can collapse the bottom pane when it is not needed.

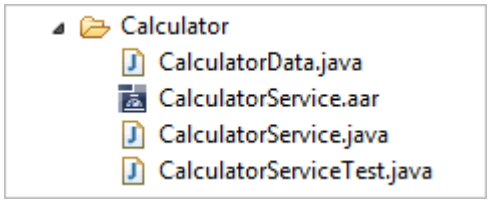
For more information on this page, refer to the EntireX documentation.

Generated Folder Structure in the Workspace

The folders and files that have been generated by the wizard are shown in the **Project Explorer** view. The following example shows a package with the name "Calculator" which is stored in the *Java/src* folder of the project - a web service has not been generated:



If you have also selected to generate a web service, the package contains an additional *.aar* file as shown in the following example:



The following files are generated into the *Java/src* folder:

- **Data Class**
  Java class containing the PDA for the business service, which is mapped to Java. The class has the same name as the business service followed by the string "Data". For example, *CalculatorData.java*.
- **Business Service Class**
  Java class containing the code that communicates with the business service, as well as the Add method. The class has the same name as the business service followed by the string "Service". For example, *CalculatorService.java*.
- **JUnit Class**
  This class is only generated when the option **Generate JUnit test file** has been selected in the wizard. The class has the same name as the business service followed by the string "ServiceTest". For example, *CalculatorServiceTest.java*. See also ***Testing the Java Class***.
- **Web Service Archive File**
  This file is only generated when the option **Generate Web Service (.aar) file** has been selected in the wizard. The file has the same name as the business service followed by the string "Service". For example, *CalculatorService.aar*.

  See the **IBM EntireX documentation** for information about working with the *.aar* file.
- **Properties File for the Business Service**
  The name of this file is always *nbs.properties*. This file is used by the `NBSSettings` class to load Broker RPC connection settings. See also ***Setting the nbs.properties File to Run a Dynamic RPC Java Class***.

Note: Some of the above files are automatically opened in the corresponding editors.

Library files with the following names are generated into the *Java/lib* folder: *wrappercore.jar* (library containing the helper classes and functionality for the generated Java class).

Setting 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. This file is used to load Broker RPC connection settings.

The `NBSSettings` class will attempt to find the *nbs.properties* 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. Class loader `NBSSettings`.
4. Context class loader.

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

Property	Description
`broker.id`	Broker connection ID.
`broker.service`	Broker service endpoint.
`broker.waitTime`	Broker wait time.
`user.id`	User ID to use with Broker.
`password`	Password to use with Broker. **Caution:** For security reasons, it is recommended that the password is not set in the *nbs.properties* file. It is much more secure to set the Broker credentials programmatically.

For example:

```
broker.id=localhost:1971
broker.service=RPC/SRV1/CALLNAT
broker.waitTime=60S
user.id=PWRUSR
password=
```

Notes:

1. If the generated code is to be used in AXIS2, you must copy the *nbs.properties* file to the `<tomcat>`/*webapps*/`<ws stack>`/*WEB-INF*/*classes* folder.
2. The default broker wait time is set to 60 seconds, if this property is not specified in the *nbs.properties* file. If wait timeouts on the client side occur, the value of the broker wait time (`broker.waitTime`) may be increased. To use another wait time, add the following property in the *nbs.properties* file: `broker.waitTime=<nnn>S` (e.g. `broker.waitTime=999S`).

Defining 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 web service `MyService` in the Java package `package.name` is invoked, the user ID `USERIDA` and the password `PWDA` will be used.

When any other service within the Java package `package.name` is invoked, the user ID `USERIDB` and the password `PWDB` will be used.

All other services will use the user ID `PWRUSR` and no password.

Testing the Java Class

When the **Generate JUnit test file** option is selected in the wizard, a JUnit class is generated. This class contains test functions that correspond to each method in the business service. For example:

```
@Test
 public void add() throws Exception {
  CalculatorService service = new CalculatorService();
  CalculatorData.CalcClass.InputDataClass inputdataclass = new ↩
CalculatorData.CalcClass.InputDataClass();
  CalculatorData.CalcClass.OutputDataClass outputdataclass = new ↩
CalculatorData.CalcClass.OutputDataClass();
  // TODO initialize data
  CalculatorData.CalcClass.CalcClassResult result = service.add(inputdataclass, ↩
outputdataclass);
  Assert.assertNotNull("Result is null", result);
  // TODO Add data verification here
 }
```

`//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 initialize data
inputdataclass.setFirstNum(BigDecimal.valueOf(2));
inputdataclass.setSecondNum(BigDecimal.valueOf(2));
```

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

```
// TODO Add data verification here
Assert.assertEquals(BigInteger.valueOf(4), ↩
result.getOutputData().getResult().toBigInteger());
```

Tests can be invoked using standard Eclipse JUnit functionality (that is, 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;
}
```

Writing 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 broker = new Broker("localhost:1971","GUEST");

// Create the service class.
CalculatorService calcservice = new CalculatorService(broker, "RPC/SRV1/CALLNAT");

// Create the service data classes
CalculatorData.CalcClass.InputDataClass input = new ↩
CalculatorData.CalcClass.InputDataClass();
CalculatorData.CalcClass.OutputDataClass output = new ↩
CalculatorData.CalcClass.OutputDataClass();

// Set input values for the call
input.setFirstNum(BigDecimal.valueOf(2));
input.setSecondNum(BigDecimal.valueOf(2));
input.setFunction("Add");

// Invoke a service method, passing in the data classes
// make sure to catch any errors that might happen.
```

```
CalculatorData.CalcClass.CalcClassResult calcresult = null;
try {
 calcresult = calcservice.add(input, output);
} catch (Exception e) {
 e.printStackTrace();
}
// Examine the result
System.out.println("Result is: " + calcresult.getOutputData().getResult());
```

9 Defining Security for a Domain

- General Information 54
- Showing the NBS Security View 54
- Showing the Security Groups for a Domain 54
- Changing the Settings for a Security Group 57
- Adding Security Groups 58
- Removing a Security Group 59
- Updating the Security Settings on the Server 60

General Information

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 the domain, business service, and/or method level.

To view and change the security settings for the domains, you use the **NBS Security** view.

Note: For more information, see *Setting Business Service Security Options* in the *Natural Business Services Administration* documentation. This is part of the Natural Business Services documentation, which is available separately.

Showing the NBS Security View

The **NBS Security** view is not shown by default when you open the NaturalONE perspective.

To show the NBS Security view

1 From the **Window** menu, choose **Show View > Other**.

2 In the resulting **Show View** dialog box, expand the **Software AG NaturalONE** node.

3 Select **NBS Security** and choose the **OK** button.

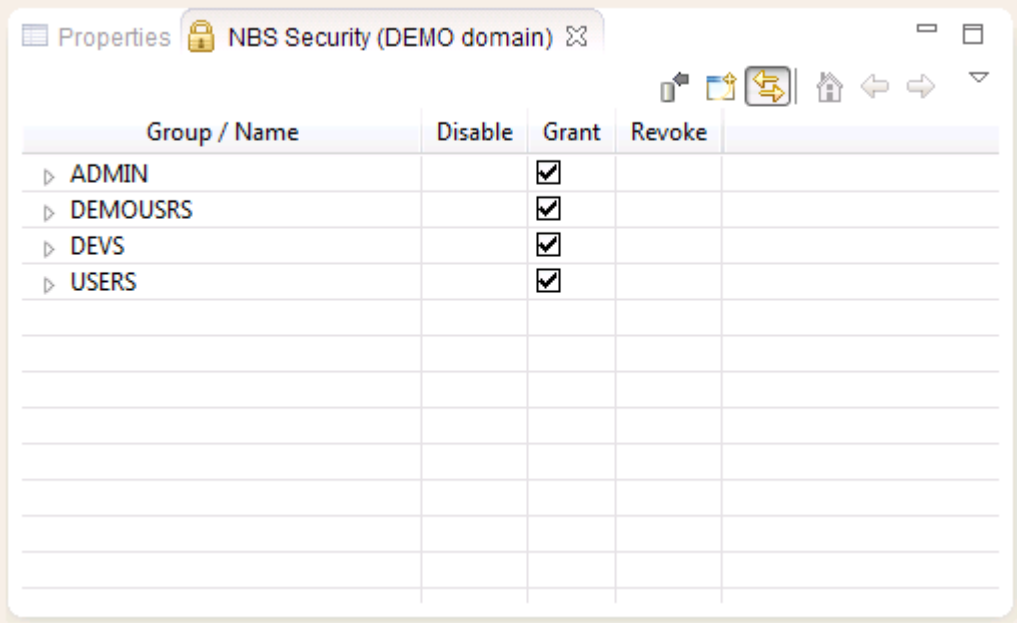
Showing the Security Groups for a Domain

You can display the security groups for domains that are available both locally and on the NBS server. For each group, you can display the business services within a group, and the methods that are defined for a business service.

To show the security details for the groups within a domain

1 In the **Project Explorer** view or **Natural Server** view, select a domain.

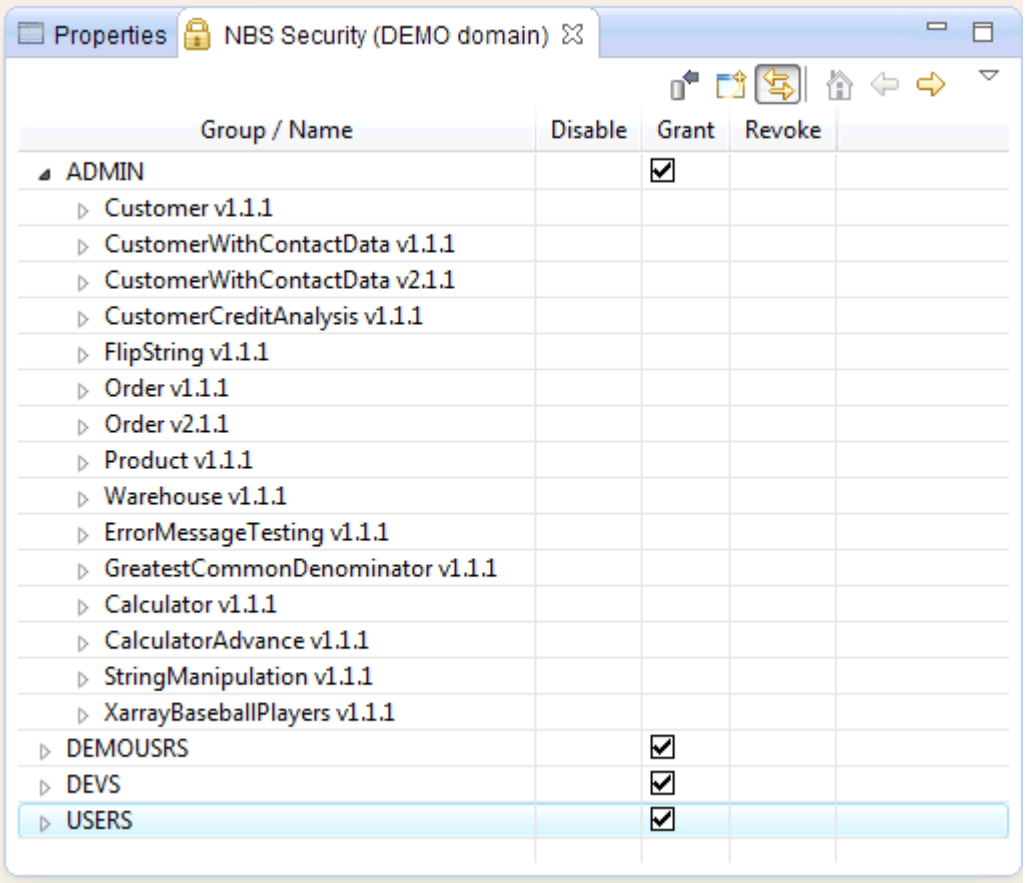
Details about the security settings of that domain are shown in the **NBS Security** view.



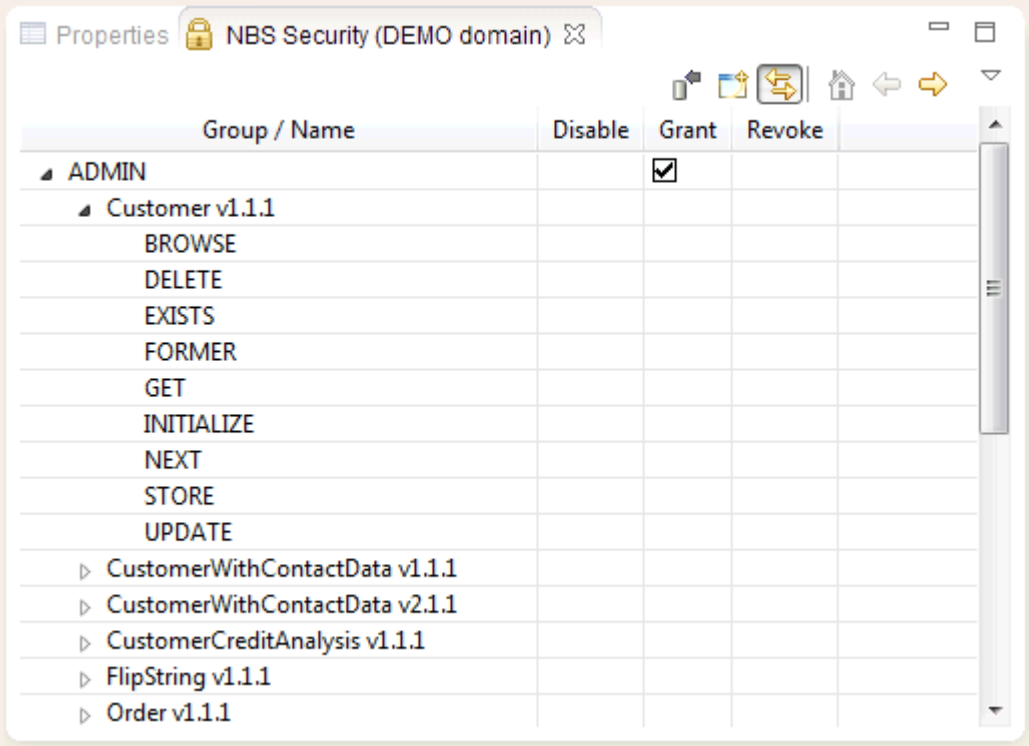
By default, the Synchronize icon ( ) is selected in the local toolbar of the **NBS Security**. When this icon is selected and you select a different domain in the **Project Explorer** view or **Natural Server** view, the details for the new domain are shown in the **NBS Security** view. If the Synchronize icon is not selected and you select a different domain, the details in the view do not change.

Tip: As it may take some time to retrieve security details from the server, you can deselect the Synchronize icon when you do not require these details.

2 To display the business services within a group, expand the group. Example:



3 To display the methods defined for a business service, expand the business service. Example:



Changing the Settings for a Security Group

You can set the security options described in the table below. You can set them on group level for all items in this group, on service level including all defined methods, or on method level only.

Security Option	Description
Grant	Allows access to an item.
Revoke	Disallows access to an item.
Disable	Temporarily revokes access to an item. When the item is reenabled, the previous setting (Grant or Revoke) is restored.

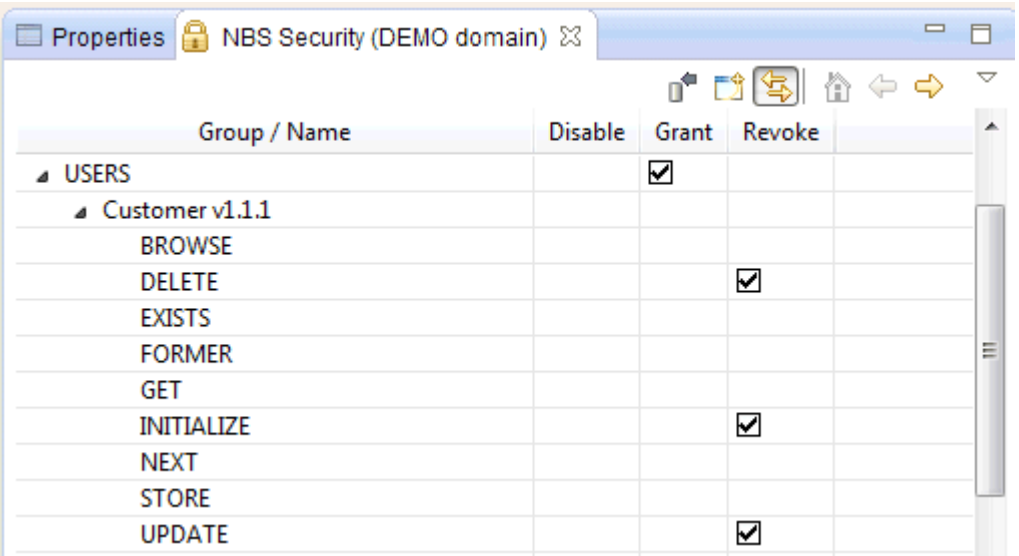
The **NBS Security** view shows columns for the above security options. When a security option is set for an item, a checkmark is shown in the corresponding column.

If you change the settings for a security group, you have to **update** the security settings on the server. Otherwise, the server does not know that the settings have been changed.

To change the security settings

- In the **NBS Security** view, click a cell in the desired column.

For example, if you have disallowed access to certain methods, this may now look as follows:



Note: To save the changed security settings, you have to update them on the server.

Adding Security Groups

You can add security groups, which have previously been defined on the server, to a domain. Keep in mind that new security groups cannot be defined on the client. This can only be done on the server.

To add a security group to a domain

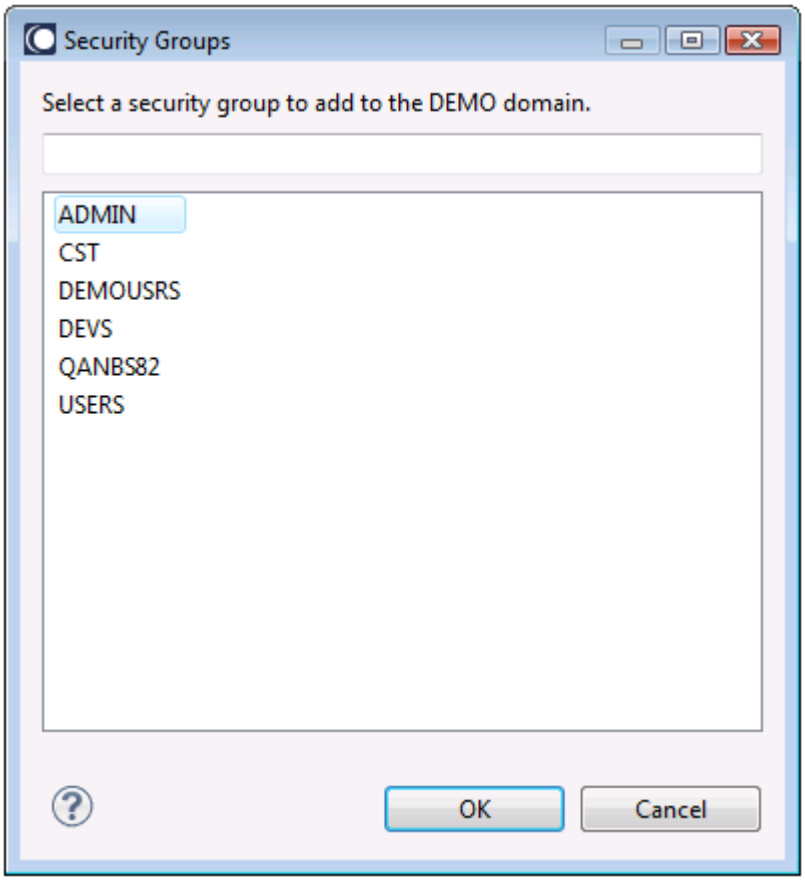
1 In the **Project Explorer** view or **Natural Server** view, select the domain to which you want to add a security group.

2 In the **NBS Security** view, invoke the context menu and choose **Add Group**.

Or:

Choose the following icon in the local toolbar:

The **Security Groups** dialog box appears. Example:



3 Select the group you want to add.

4 Choose the **OK** button.

Removing a Security Group

You can remove a security group from a domain.

If you remove a security group, you have to **update** the security settings on the server. Otherwise, the server does not know that the group has been removed.

To remove a security group from a domain

1 In the **NBS Server** view, select the group that you want to delete.

2 Invoke the context menu and choose **Delete Group**.

Note: You will not be asked whether you really want to remove the group. However, this change will only become effective when you update the security settings on the server.

Updating the Security Settings on the Server

Each time you change the settings for a security group, you have to update them on the server.

To update the security settings on the server

1 In the **NBS Server** view, select a group, business service or method.

2 Invoke the context menu and choose **Update**.

Or:

Choose the following icon in the local toolbar:

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

10 Setting the Preferences for Business Services

- Showing the Preferences for Business Services 62
- Business Services 63
- Installation 65

Showing the Preferences for Business Services

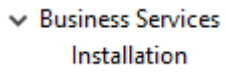
The preferences for business services are set in the **Preferences** dialog box of Eclipse.

To show the preferences for business services

1 From the **Window** menu, choose **Preferences**.

2 In the tree of the resulting dialog box, expand the **Software AG** node and then the **Business Services** node.

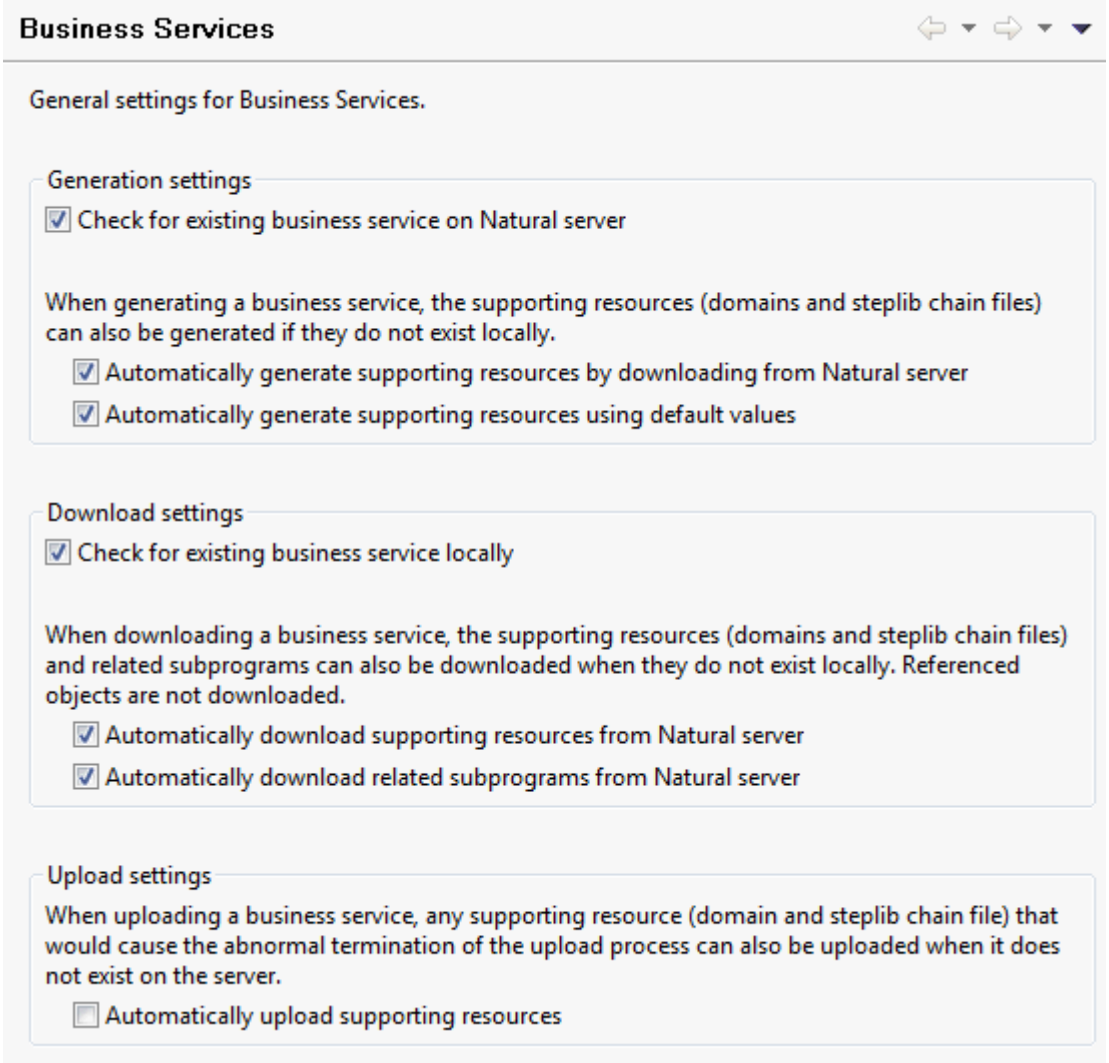
Different pages are provided for setting different types of preferences.



3 Select one of the pages in the tree and set the required options as described in the topics below.

4 Choose the **OK** button to save your changes and to close the dialog box.

Business Services



Generation Settings

The following options apply when you create new business services locally. See also *Creating New Business Services*.

Check for existing business service on Natural server

When selected, the Natural server is checked for the existence of the business service being generated. If it exists already, a warning message is shown.

Automatically generate supporting resources by downloading from Natural server

When selected, the domains and steplib chain files are automatically downloaded from the Natural server if they do not yet exist locally.

Automatically generate supporting resources using default values

When selected, the domains and steplib chain files are automatically generated using default values if they do not yet exist locally.

Note: This option only applies when **Automatically generate supporting resources by downloading from Natural server** is not selected, or is selected but the resource does not exist on the Natural server.

Download Settings

The following options apply when you download business services from a server. See also *Downloading Business Service Resources from a Server*.

Check for existing business service locally

When selected, it is checked whether the selected business service already exists locally. If it exists already, a warning message is shown.

Automatically download supporting resources from Natural server

When selected, the domains and steplib chain files are automatically downloaded from the Natural server if they do not yet exist locally.

Automatically download related subprograms from Natural server

When selected, the subprograms which belong to a business service are automatically downloaded from the Natural server if they do not yet exist locally.

The referenced Natural objects such as DDMs or data areas, however, are not downloaded automatically.

Upload Settings

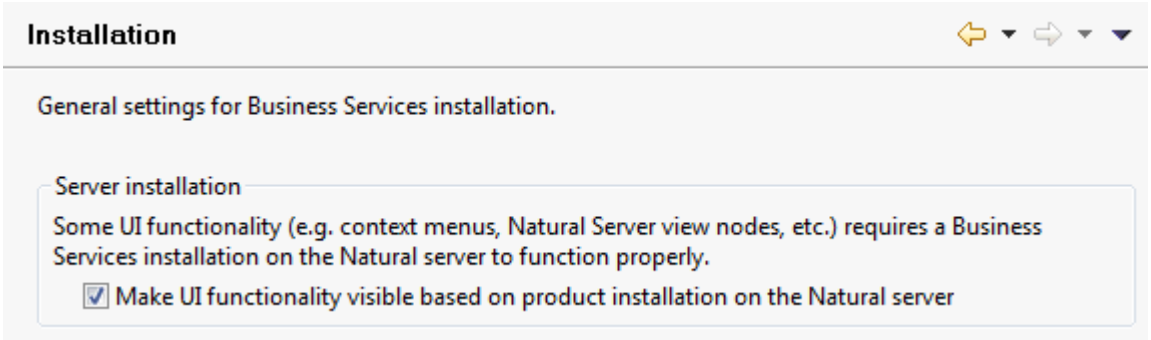
The following option applies when you upload business services to a server. See also *Uploading the Business Service Resources*.

Automatically upload supporting resources

When selected, the domains and steplib chain files are automatically uploaded if they do not yet exist on the Natural server. This eliminates server exceptions that can prevent the selected resources from being uploaded to the server.

Installation

To function properly, certain UI functions require a Natural 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 service resources from a Natural server to a local Natural project, but only when there is a Natural Business Services installation on the server.



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

When selected (default), the UI functions are made visible based on the installation of Natural Business Services on the server. To accomplish this, a server call determines which products are installed on the server and the results are cached until NaturalONE shuts down.

When deselected, all UI functions are visible, even when Natural Business Services is not installed on the server. No server calls will be made to determine which products are installed on the server.