

Natural Business Services

Natural Business Services Subprogram-Proxy-Client Model

Version 5.3.1

February 2010

This document applies to Natural Business Services Version 5.3.1.

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

Copyright © 2006-2010 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Table of Contents




1 Natural Business Services Subprogram-Proxy-Client Model	1
2 Introduction	3
3 Using the Subprogram-Proxy-Client Model	5
In the Generation Subsystem	6
In the Eclipse Plug-in for Natural Business Services	7
In the Natural Plug-in for Natural Business Services	7
4 Calling Natural Objects	9

1 Natural Business Services Subprogram-Proxy-Client Model

Natural Business Services Subprogram-Proxy-Client Model describes how to generate a subprogram proxy using the Subprogram-Proxy-Client model.

This documentation is intended for developers and others who want to create subprogram proxy clients using the supplied model.

Natural Business Services Subprogram-Proxy-Client Model covers the following topics:

 Introduction	Provides an overview of the Subprogram-Proxy model.
 Using the Subprogram-Proxy-Client Model	Describes the Subprogram-Proxy-Client model.
 Calling Natural Objects	Describes the parameter interface and how to invoke services locally and via EntireX.

2 Introduction

The Subprogram-Proxy-Client model generates a subprogram proxy to invoke Natural business services from a client. Clients can call these Natural subprograms from the same Natural environment or use EntireX to call them from a non-Natural Business Services Natural environment (by supplying the NBS server values in the call to the Natural client).

When generating a subprogram proxy to invoke business services from a client, remember the following:

- The proxy must be generated on the server.
- After generation, the proxy and all proxy-specific PDAs must be moved to the client and stowed.



Note: To invoke business services from a client that does not have Natural Business Services installed, you must first install the Natural client service runtime component. For information, see the applicable installation guide.

3 Using the Subprogram-Proxy-Client Model

- In the Generation Subsystem 6
- In the Eclipse Plug-in for Natural Business Services 7
- In the Natural Plug-in for Natural Business Services 7

You can generate the Subprogram-Proxy-Client model in the Generation subsystem or use the Business Service wizard in the Natural or Eclipse plug-in to do this during the generation of a business service. This section covers the following topics:

In the Generation Subsystem

▶ **To generate the Subprogram-Proxy-Client model in the Generation subsystem:**

- 1 Enter "ncstg" on the Natural command line.

The Natural Construct Generation main menu is displayed.



Note: For more information on using the Generation main menu, see *Natural Construct Generation*.

- 2 Type "M" in *Function*.
- 3 Type the name of your subprogram proxy client in *Module*.
- 4 Type "Subprogram-Proxy-Client" in *Model*.
- 5 Press Enter.

The Standard Parameters panel for the Subprogram-Proxy-Client model is displayed. For example:

```
CUSQMA          SUBPROGRAM-PROXY-CLIENT Subprogram          CUSQMA0
Jun 01          Standard Parameters                          1 of 1

Module ..... SPCLIENT
System ..... CSTDEMO_____

Title ..... Subp proxy client for..._
Description ..... Description of subprogram proxy client that..._____
_____
_____

Server proxy subp .. _____ *
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
main  help  retrn quit                                userX main
```

- 6 Select the name of the server proxy from `Server proxy subp`.

Use one of the following methods to determine the name of the server proxy:

- Scan the library for subprograms generated by the Subprogram-Proxy model
- View the service module names listed in the repository in the Natural or Eclipse plug-in

- 7 Press Enter.

The Generation main menu is displayed.

- 8 Enter "G" in `Function` to generate the subprogram proxy client.

In the Eclipse Plug-in for Natural Business Services

To generate the subprogram proxy client in the Eclipse plug-in:

- For a new business service, select **Auto-generate Natural clients** on the **Enter Service Name and Select Domain** panel before generating the service. For information, see *Create a Business Service*.
- For an existing business service, select **Generate Natural client** on the context menu for the service in the Service Repository Explorer. For information, see *Generate a Natural Client*.

In the Natural Plug-in for Natural Business Services

To generate the subprogram proxy client in the Natural plug-in:

- For a new business service, select **Auto-generate Natural clients** on the **Enter Service Name and Select Domain** panel before generating the service. For information, see *Create a Business Service*.
- For an existing business service, select **Generate Natural client** on the context menu for the service in the Business Service Repository. For information, see *Generate a Natural Client*.

4 Calling Natural Objects

Each generated Natural client will have the following parameter interface:

```
PARAMETER
01 CALL-PARMS
  02 BROKER-ID (A32)
  02 SERVER-NAME (A80)
  02 SERVER-CLASS (A80)
  02 SERVER-SERVICE (A80)
01 #METHOD (A32)
PARAMETER
...[your service parameters]...
```

▶ **To invoke the service locally:**

- Do not supply values for the fields in the CALL-PARMS structure.

▶ **To invoke the service via EntireX:**

- Supply the values for your Dispatch server in the CALL-PARMS structure (in SERVER-NAME).

The default values in the CALL-PARMS structure are:

- SERVER-NAME=DISPATCH
- SERVICE-CLASS=BUSINESS
- SERVER-SERVICE=MAIN



Tip: You can also use the CSXCSQI supplied sample routine in the SYSCSTX library to automatically configure the CALL-PARMS interface when your client is invoked. For more information, refer to the source for CSXCSQI.

