SECURE INTERNET TECHNOLOGIES

xtradyne

Web Services Domain Boundary Controller v 3.1

Quick Installation Guide

PrismTech (a) makes no warranty of any kind with regard to this product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose; and (b) and the suppliers disclaim all liability in connection with your use of the product, including liability for all direct indirect damages or loss of profit, business interruption, loss, damage or destruction of data or for special, incidental or consequential damages or for any other indirect damages such as, but not limited to exemplary or punitive damages.

USA Corporate Headquarters
PrismTech Corporation
400 TradeCenter, Suite 5900
Woburn, MA, 01801
USA
Tel: (1) 781-569-5819

European Headquarters
PrismTech Limited
5th Avenue Business Park, Team Valley
Gateshead, Tyne & Wear, NE11 0NG
United Kingdom
Tel: +44 (0) 191-4979900

For product or technical questions, please contact our Customer Response Center,
Email: **crc@prismtech.com**
For licensing an pricing questions, please email to **sales@prismtech.com**

1st edition
Issue Date: 1 April 2010
Security Policy Server v. 3.1

Contents

1 Introduction

This guide is intended to support rapid evaluation of the DBC's basic features. The first part explains the components that make up the Domain Boundary Controller and how to install them. The second part demonstrates the DBC's basic functionality following a step by step example.

2 DBC Components and Installation

2.1 DBC Components

The *Domain Boundary Controller* is an infrastructure building block. It is modularized into the following components that may be physically separated:

- the *DBC Proxy*, which transparently intercepts client requests and enforces security policies,
- the *Security Policy Server* (SPS), which manages these policies, and
- the *Administration Console*, an advanced GUI tool, which allows for conveniently configuring the whole DBC system and managing security policies.

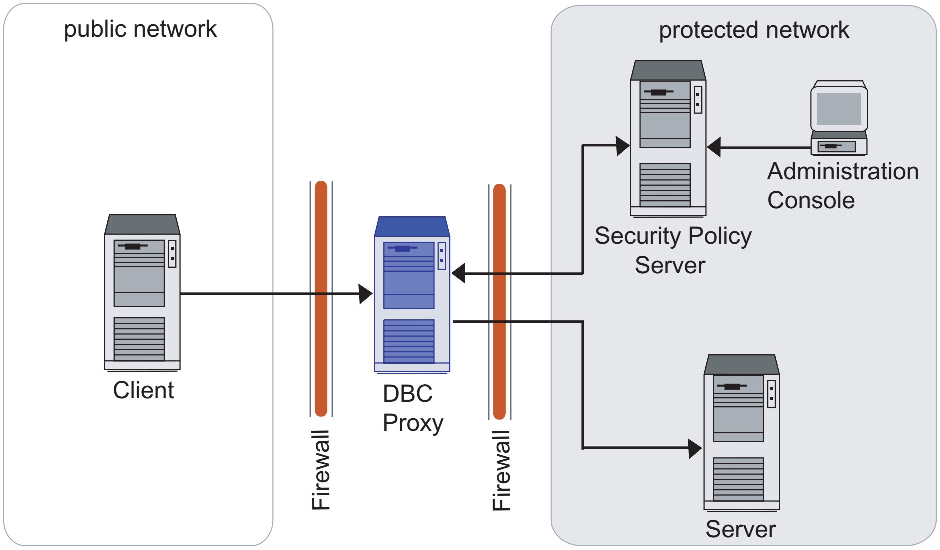


Fig. 1. Separation of DBC Components

As illustrated in figure 1 the main stronghold towards the public Internet is realized by the *DBC Proxy*. The Security Policy Server and the Admin Console are usually located inside the protected internal network, i.e., behind the DBC Proxy and any existing packet filters.

2.2 Component Specifications

DBC Proxy

The DBC Proxy is generally installed on a host that will become part of the firewall. All firewall machines are potential targets of attacks, so they require great care in the configuration of their operating system and their network components. Therefore, a hardened configuration of the operating system should be set up. A hardened configuration is a minimal system configuration which possibly relies on additional tools or libraries to increase the resilience of the system against attacks (cf. Chapter 6 “Hardened System” on page 55 of the Deployment Guide).

Security Policy Server

The Security Policy Server should be installed on a host located in a protected domain. However, for scenarios with less stringent security requirements, it can be installed on the same host as the DBC Proxy. If your security policy requires protection against attacks from the inside, you should also set up a hardened operating system configuration for the Security Policy Server host.

Administration Console

The Xtradyne Administration Console is written in Java and thus requires a platform where the Java2 Runtime Environment (JRE) is available. The JRE is included in the installation package of the Admin Console. Installation packages are available for the Linux and Solaris platforms as specified above. On the PC x86 platform the Admin Console is also supported on the following operating systems:

- Windows NT4 SP6a
- Windows 2000 SP2
- Windows 98 SR2
- Windows XP SR1a

To ensure acceptable performance, the host on which the Admin Console will be installed should have at least 256 MB RAM and 128 MB free disk space (recommended are 512 MB).

2.3 DBC Deployment

The Domain Boundary Controller can be deployed in different ways according to your requirements. For a detailed discussion, please refer to the Administrator’s Guide. For evaluation purposes all DBC components can be installed on one host.

2.4 Prerequisites

Table 1 gives an overview about the hardware and operating system requirements for the DBC Proxy, the Security Policy Server, and the Admin Console.

	Hardware Requirements	**Supported OS**
PC x86 Platform	• CPU: Intel Pentium III at 600 MHz • RAM: 256 MB minimum, 512 MB recommended • Free disk space: 512 MB • 1 Network Interface Card (NIC): up to four NICs supported on the Proxy host	• RHEL 3 and RHEL 4 WS or AS Update 4 (and higher) • SuSE Professional & Enterprise Linux 8.x and higher • Novell SuSE Linux Enterprise 10.x and open SUSE 10.x • Solaris 10 x86 (I-DBC only)
Sun sparc Platform	• CPU: 440-MHz UltraSPARC-II CPU • RAM: 256 MB minimum, 512 MB recommended • Free disk space: 512 MB • 1 Network Interface Card (NIC): up to four NICs supported on the Proxy host	• Solaris 9 (kernel patch 112233-02) • Solaris 8 (kernel patch 108528-06 or higher) • Solaris 10 (kernel patch 118822-29 or higher)

Table 1. DBC Hardware Requirements and Supported Platforms

Note that if you want to operate the DBC Proxy and Security Policy Server co-located on a single machine, this machine should have at least 512 MB RAM.

⚠

2.5 Installation

The distribution (archive or CD-ROM) contains the directories `linux_glibc-2.2`, `linux_glibc-2.3`, `solaris`, and `windows`. The table below indicates which supported Linux distribution use the `glibc-2.2` or `glibc-2.3` respectively. Please choose the installer for your distribution from the appropriate directory.

glibc Version	Linux Distribution
glibc-2.2	• SuSE Linux 8.x • SuSE Linux 9.0
glibc-2.3	• RedHat Enterprise Linux 3.0 (Update 4 and higher) and 4.0 • SuSE Linux 9.1 and higher

Table 2. glibc version of supported Linux Distributions

The following can be found in the respective directories:

Linux & Solaris

Linux & Solaris

- The installer for the **I-DBC Proxy**: `IDBCInstall-3.1.<x>.bin`
- The installer for the **WS-DBC Proxy**: `WSDBCInstall-3.1.<x>.bin`
- The **Admin Console** installer: `AdminConsoleInstall-3.1.<x>.bin`
- The installer for the **I-DBC example** application Frankfurter Bank: `FrankfurterBankIIOP_Install-1.1.<x>.bin`
- The installer for the **WS-DBC example** application Frankfurter Bank: `FrankfurterBankSOAP_Install-1.1.<x>.bin`

Windows

Windows

- The **Admin Console** installer: `AdminConsoleInstall-3.1.<x>.exe`
- The installer for the **I-DBC example** application Frankfurter Bank: `FrankfurterBankIIOP_Install-1.1.<x>.exe`
- The installer for the **WS-DBC example** application Frankfurter Bank: `FrankfurterBankSOAP_Install-1.1.<x>.exe`

Additionally, in the top level directory, you will find the following documentation files:

- a copy of this Administrator Guide in PDF format (`AdminGuide.pdf`),
- a Quick Installation Guide (`QuickInstallation.pdf`) to help with quick evaluation of the product,
- a Deployment Guide (`DeploymentGuide.pdf`) which explains different deployment scenarios and specialties e.g. high availability, and
- the release notes for this version.

Mounting the CD ROM

This section may be skipped if you downloaded the distribution as a tar archive.

On Linux, a CD can usually be mounted to make it available to the system by typing: Linux

`mount /cdrom on SuSE Linux`

`mount /mnt/cdrom` on Red Hat Linux

If this does not work, consult your operating system manuals. After mounting, the files contained on the CD will be available at the cdrom mount point.

On Solaris, the CD is usually mounted automatically under `/cdrom`. If not, use the command `volcheck`. If the Volume Manager is not running, determine the device name of the CD drive and enter the following commands to mount the CD: Solaris

```
mkdir /cdrom/dbc_cd
```

```
/usr/sbin/mount -f hsfs -r /dev/dsk/cddevice /cdrom/dbc_cd
```

Installer Prerequisites

To run the DBC installer root privileges are required.

Apart from installing the selected DBC component the installer will also perform some basic configuration and administrative operations. This includes the creation of an administrative UNIX user account (cf. “Choose a DBC User Name” on page 11).

Before starting the installer, make sure that the DISPLAY environment variable is set correctly, otherwise the installer will fail to run. When using `bash` or `sh` as shell, type:

```
DISPLAY=<host>:0; export DISPLAY
```

If no X windowing system is available, the installer can be run in console mode (start the installer with the option `-i console`).

2.6 Installation Steps

The installer comes up with a welcome screen and then guides you through the installation process. The following sections describe each panel in detail.

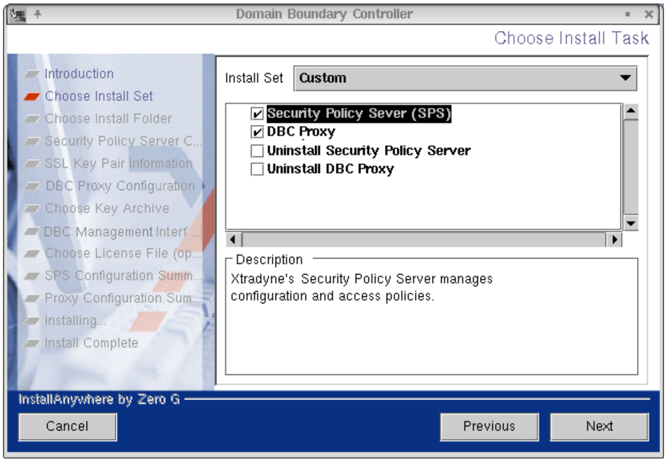
Choosing the Install Task

You can choose between the following installation tasks:

- Security Policy Server (SPS): This will install the Security Policy Server.
- DBC Proxy: This will install the DBC Proxy.
- Uninstall Security Policy Server: This will uninstall the Security Policy Server.
- Uninstall DBC Proxy: This will uninstall the DBC Proxy.

For evaluation purposes we recommend to install all components on a single host: Security Policy Server, and DBC Proxy. Check the appropriate boxes (as shown in figure 2, “Choosing Installation Tasks”) and proceed.

Note that when the Security Policy Server and the DBC Proxy are installed on different hosts, the Security Policy Server should be installed first because during the installation of the SPS, keys and certificates are created that are indispensable for the DBC Proxy.



Domain Boundary Controller

Choose Install Task

Install Set: Custom

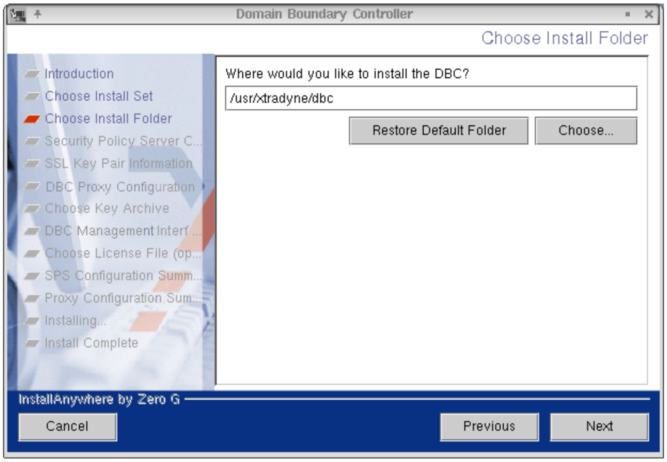
☑ Security Policy Sever (SPS)
☑ DBC Proxy
□ Uninstall Security Policy Server
□ Uninstall DBC Proxy

Description
Xtradyne's Security Policy Server manages configuration and access policies.

Fig. 2. Choosing Installation Tasks

Choose the Install Folder

Please enter an installation target directory, as shown in figure 3, “Choosing an installation directory”. If the default is acceptable (the directory `/usr/xtradyne/` on Linux and `/opt/xtradyne/` on Solaris), simply click on the “Next” button. In the following we write `<INSTALLDIR>` to refer to the installation directory of the DBC.



Where would you like to install the DBC?

/usr/xtradyne/dbc

Fig. 3. Choosing an installation directory

Depending on the chosen component the following subdirectories will be created:

- `idbc` (when installing an **I-DBC** Proxy),
- `wsdbc` (when installing a **WS-DBC** Proxy),
- `sps` (when installing a **Security Policy Server**).

Choose a DBC User Name

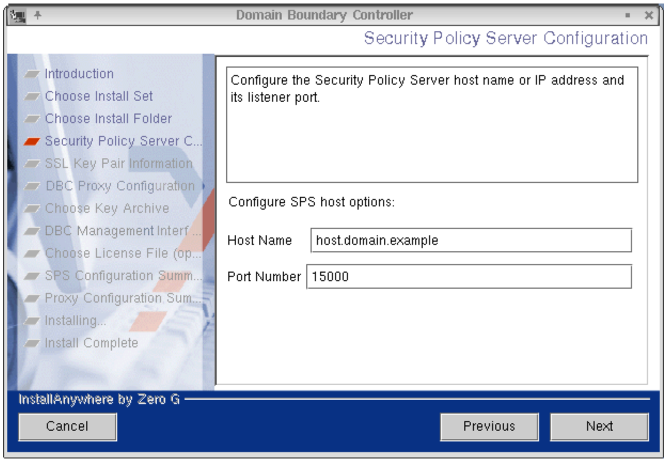
Please choose a user name for the DBC account. This user account will be created during the installation (`xtradyne` by default). DBC processes will run under this user ID.

Security Policy Server Configuration

This panel only appears if the “**Security Policy Server**” task is selected.

The next step in the installation process is to enter basic configuration information for the Security Policy Server, as shown in figure 4, “Entering configuration data for the

Security Policy Server”. You will be asked to provide information about the host name and the port number by which the Security Policy Server can be contacted.


Domain Boundary Controller
Security Policy Server Configuration
Introduction
Choose Install Set
Choose Install Folder
Security Policy Server C...
SSL Key Pair Information
DBC Proxy Configuration
Choose Key Archive
DBC Management Interf...
Choose License File (op...
SPS Configuration Summ...
Proxy Configuration Sum...
Installing...
Install Complete
InstallAnywhere by Zero G
Cancel
Previous
Next

Configure the Security Policy Server host name or IP address and its listener port.

Configure SPS host options:

Host Name host.domain.example

Port Number 15000

Fig. 4. Entering configuration data for the Security Policy Server

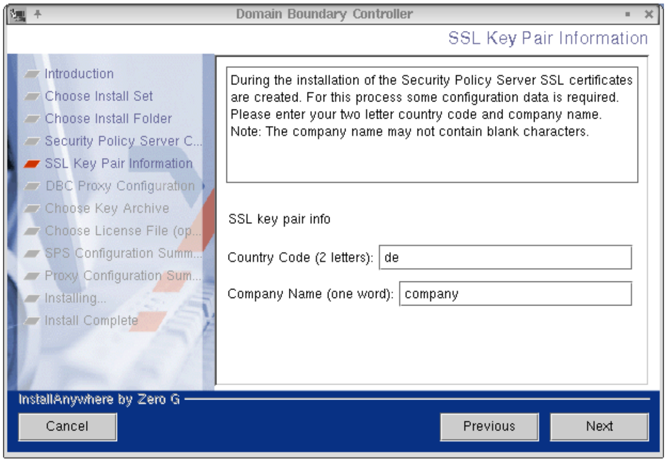
⚠ Note that the listener port for the Security Policy Server may need to be opened on a firewall if the DBC is deployed across several machines (cf. section “Typical Deployment Scenarios” on page 63 of the Administrator’s Guide).

Choose SSL Key Pair Information

This panel only appears if the “**Security Policy Server**” task is selected.

During installation of the Security Policy Server SSL certificates and trust stores are created to establish trust relations between DBC components. Please enter your company name and two letter country code (e.g., `us` for United States of America, `uk` for the

United Kingdom, de for Germany, etc.) into the text boxes. This data will be used as the Issuer Name in the generated certificates.



During the installation of the Security Policy Server SSL certificates are created. For this process some configuration data is required. Please enter your two letter country code and company name. Note: The company name may not contain blank characters.

SSL key pair info

Country Code (2 letters): de

Company Name (one word): company

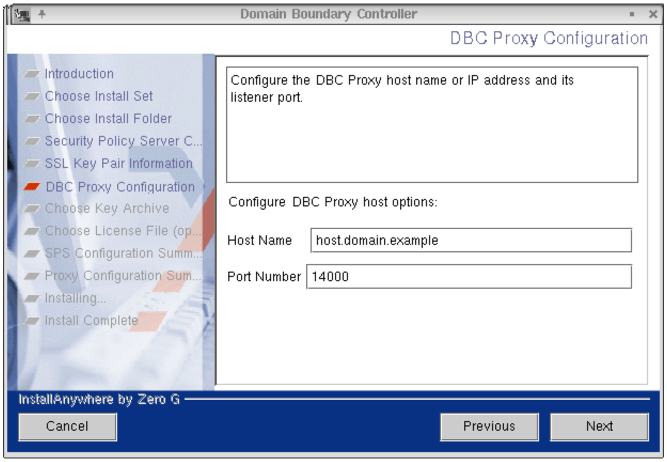
Fig. 5. SSL Key Pair information

A detailed description of the DBC’s key management, i.e., which communication paths are secured by which keys and a list of all the key and certificate files generated by the installer, is given in chapter 10, “Installing Keys and Certificates” on page 201 of the Administrator’s Guide.

DBC Proxy Configuration

The next step in the installation process is to enter basic configuration information for the DBC Proxy, as shown in figure 6, “Entering configuration data for the Security Policy Server”. You will be asked to provide information about the host name and the port number by which the DBC Proxy can be contacted.

Note that if you install only the SPS this information is optional and can be configured later on with the Admin Console.



Configure the DBC Proxy host name or IP address and its listener port.

Configure DBC Proxy host options:

Host Name host.domain.example

Port Number 14000

Fig. 6. Entering configuration data for the Security Policy Server

Note that the listener port for the DBC Proxy may need to be opened on a firewall if the DBC is deployed across several machines (cf. section “Typical Deployment Scenarios” on page 63 of the Administrator’s Guide). The default listener port for the **I-DBC** is 14000. The default listener port for the **WS-DBC** is 14001.

Choose Key Archive

This panel only appears if **only** the “**DBC Proxy**” task is selected.

When installing the Security Policy Server certificates and trust stores are created. The certificates needed on the DBC Proxy host can be found in the file `<INSTALLDIR>/sps/adm/ProxyKeys.tar`. Please provide this file name here. If this file is not available, you may skip this step and install the keys later on. Please refer to section “Generating Keys” on page 90 of the Administrator’s Guide on how to install the keys manually.

Choose License File

Depending on the selected component you can provide a license file for the SPS or the Proxy. The license is available from your reseller. If you did not obtain a license yet you may skip this step and install the license later on (cf. gray box “Installing the License” on page 88 of the Administrator's Guide). Note that DBC and SPS will not start if no license file is present.

Configuration Summary

This panel shows the configuration summary of the each selected component. Please review the data carefully before continuing.

Installing...

After clicking "Next", the installation will be carried out. This will take a few moments.

In case of any problems during the installation of the DBC Proxy, see chapter 11, "Troubleshooting" on page 217 of the Administrator's Guide.

2.7 Startup, Shutdown, and Restart

Security Policy Server

The script for starting the Security Policy Server is called `xdn_sps`. It is located in the directory `/etc/init.d`. A symbolic link (`rcxdn_sps`) is created in `/usr/sbin/` on Solaris and `/sbin` on Linux. The start script is automatically executed when entering into runlevel 3 or 5 and runlevel 2 on Solaris.

The script can also be run from the command line. You have to be root to run the script. The following parameters can be used:

- to start the Security Policy Server: `rcxdn_sps start`
- to restart the Security Policy Server: `rcxdn_sps restart`
- to stop the Security Policy Server: `rcxdn_sps stop`
- to get status information: `rcxdn_sps status`

Note that you can also restart the Security Policy Server with the Admin Console. Section "When to Restart the DBC Proxy / Security Policy Server" on page 120 of the Administrator's Guide lists which changes in the configuration require a restart of the Security Policy Server.

DBC Proxy

The script for starting the I-DBC Proxy is called `xdn_idbc`. The script for starting the WS-DBC is called `xdn_wsdbc`. They are located in the directory `/etc/init.d`.

Symbolic links (`rcxdn_idbc` for the I-DBC Proxy and `rcxdn_wsdbc` for the WS-DBC) are created in `/usr/sbin/` on Solaris and in `/sbin` on Linux. The start script is automatically executed when entering runlevel 3 or 5 on Linux and in runlevel 2 on Solaris.

The script can also be run from the command line. To run the script, you have to be `root`. The parameters `start`, `restart`, `stop`, and `status` can be used.

Note that you can also restart the DBC with the Admin Console. Section “When to Restart the DBC Proxy / Security Policy Server” on page 120 of the Administrator’s Guide lists which changes in the configuration require a restart of the DBC.

2.8 Installation of the Admin Console

The Admin Console can be installed on Windows, Linux, or Solaris. In the distribution (archive or CD-ROM), you will find the directories `linux_glibc2-2`, `linux_glibc2-3`, `solaris`, and `windows`. These directories contain the executables for the respective operating systems that will install the Admin Console (`AdminConsole.bin` for Linux and Solaris and `AdminConsole.exe` for Windows).

Note that the Admin Console is a Java application, therefore the version of the glibc (for Linux) is not relevant. If you would like to install on a Linux system, you can choose an `AdminConsoleInstall-3.1.<x>.bin` installer from any of the linux directories.

Windows

To start the installation execute the file `AdminConsoleInstall-3.1.<x>.exe` in the `windows` directory.

Linux and Solaris

When installing from CD-ROM, please mount the CD-ROM as described in section “Mounting the CD ROM” on page 9. To start the installation execute the file `AdminConsole.bin` in the `linux` or `solaris` directory you just mounted.

Installation Steps

The installer will lead you through the set up process.

1. Introduction
2. Choose an Install Folder. The default destination is:
   - on Windows: `<Program Files>\xtradyne\`
   - on Linux: `/usr/xtradyne/`
   - on Solaris: `/opt/xtradyne/`
3. Please choose the `license.txt` file. It will be installed in the directory `<INSTALLDIR>/adminconsole/bin`. You can skip this step and copy or update the license file later manually. But if the Admin Console is started without a valid license not all features are available.
4. Installation Summary

Installing SSL Keys

We recommend configuring the Admin Console to use SSL with client authentication to communicate securely with the Security Policy Server. However, for evaluation purposes keys don't need to be installed and client authentication can be disabled (this is actually the default).

If you would like to install SSL keys for the Admin Console, please refer to section "Installing SSL Keys" on page 101 of the Administrator's Guide.

2.9 Summary

Up to now you have successfully installed the Domain Boundary Controller. The next section shows along with an example how to use the DBC to protect a service.

3 Protecting Web Services – Example

This chapter presents the example Web Service “Frankfurter Bank”. This is a simple fictitious bank application to illustrate how to run a Web Service across the WS-DBC and how to define an access control policy using the Administration Console.

3.1 Problems and Solution

While Web Services are a convenient, low-overhead architecture for integrating heterogeneous enterprise resources, they do open up access paths to critical assets that did not previously exist. Because Web Services use HTTP as the underlying transport protocol to enable firewall traversal, ordinary packet-filter firewalls are no longer an effective protection mechanism. Since these firewalls are typically configured to pass on HTTP requests but do not understand the SOAP messages contained in them, basically any host on the Internet can now send SOAP messages to a resource. If no additional precautions are taken then Web Services become an easy target for attackers.

The *Web Services DBC* is a flexible security appliance that can be transparently integrated into existing network infrastructures to protect Web Services. It acts as a security proxy for Web Services by providing virtual (proxified) service endpoint URLs for existing Web Services. The WS-DBC is a complete security solution that provides full AAAA (*Authentication, Authorization, Auditing, Administration*) functionality and content inspection to protect enterprise resources that are exposed as Web Services.

In the following sections the example Web Service Frankfurter Bank is presented and protected by employing the WS-DBC. Note that not all the features of the WS-DBC and the Frankfurter Bank Web Service are explained here, as this is beyond the scope of this Quick Installation Guide.

3.2 Getting familiar with the Frankfurter Bank Web Service

You are the security administrator of a fictitious bank Web Service “Frankfurter Bank”. The web service offers some operations for account managers and also has a small database (cf. figure 7, “Frankfurter Bank”):

- there are two account managers: Mrs. Monica Meyer and Mr. Stefan Bauer,
- they can withdraw and deposit money from/to accounts,

- there are two accounts in the database: The account with the number “4711” is owned by Max Smith and initially holds 750 EUR. The account with the number “1234” is owned by Ann Carter and initially holds 333 EUR.

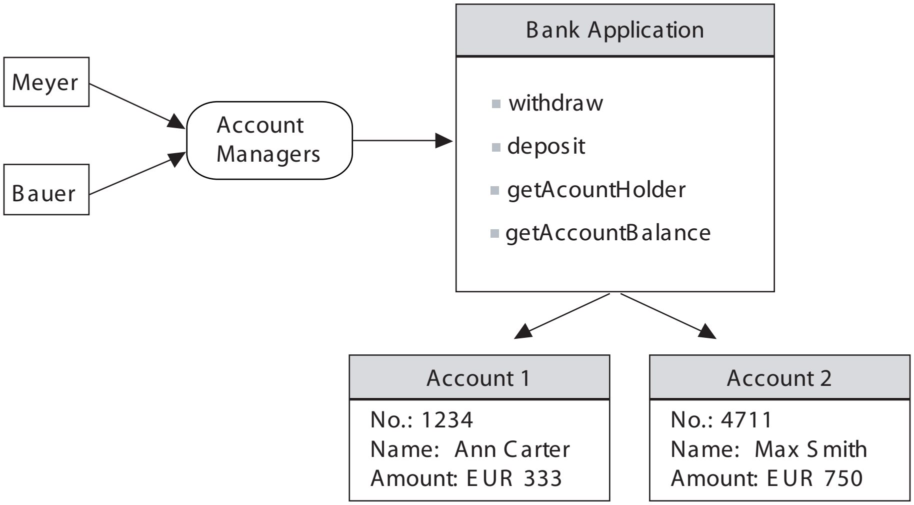


Fig. 7. Frankfurter Bank

Prerequisites for the Frankfurter Bank Example

The Frankfurter Bank example is a web service based on Apache Axis and written in Java. The Frankfurter Bank Server is a self-contained simple Axis Server. The example can be run on Linux, Solaris, or Windows.

The installer for this example is included in the DBC distribution (`FrankfurterBankSOAP_Installer.bin` for Linux/Solaris, or for Windows `FrankfurterBankSOAP_Installer.exe` respectively). Running the installer will lead you through the setup of the example. By default, Frankfurter Bank is installed in `<user_home>/FrankfurterBank/soap` on Linux/Solaris and in `C:\Program Files\FrankfurterBank\soap` on Windows.

Running Frankfurter Bank

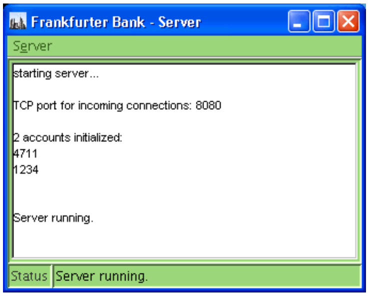
To get a first impression of how the web service works, go to the directory where the web service has been installed and start the Bank Server and Bank Client:

- Linux/Solaris:
  - Start the Frankfurter Bank Server: `bin/FBServer`
  - Start the Frankfurter Bank Client: `bin/FBClient`
- Windows:
  - Frankfurter Bank Server: **Start ➜ Programs ➜ Frankfurter Bank IIOP ➜ FBServer**
  - Frankfurter Bank Client: **Start ➜ Programs ➜ Frankfurter Bank IIOP ➜ FBClient**

FBServer

The Server comes up with a window that displays some status information:

- the listener port (by default TCP port 8080),
- the predefined accounts,
- and the status (e.g. Server running).



```
starting server...

TCP port for incoming connections: 8080

2 accounts initialized:
4711
1234


Server running.
```

Status Server running.

Fig. 8. Frankfurter Bank Example: Server

FBClient

When starting the Frankfurter Bank Client it will use the default URL `http://127.0.0.1/:8080/axis/services/FrankfurterBankPort` to contact the Frankfurter Bank Web Service (depicted in figure 9, “Frankfurter Bank Client and Server: Direct connection”).

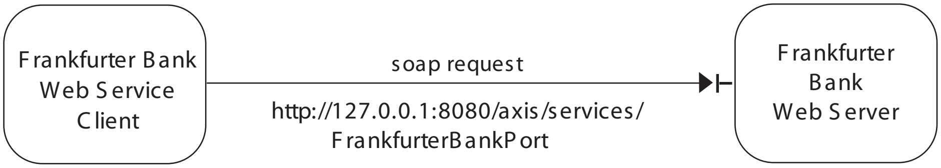
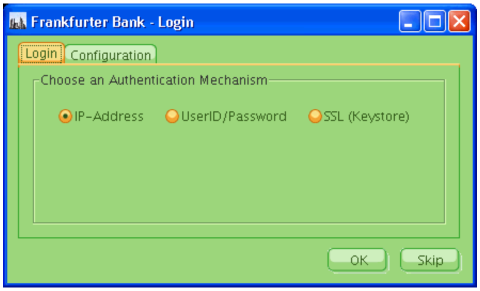

Frankfurter Bank Web Service Client
soap request
http://127.0.0.1:8080/axis/services/FrankfurterBankPort
Frankfurter Bank Web Server

Fig. 9. Frankfurter Bank Client and Server: Direct connection

The client program comes up with a panel which offers different authentication methods on the “Login” tab (for a first start, you can choose authentication by “IP-Address”).

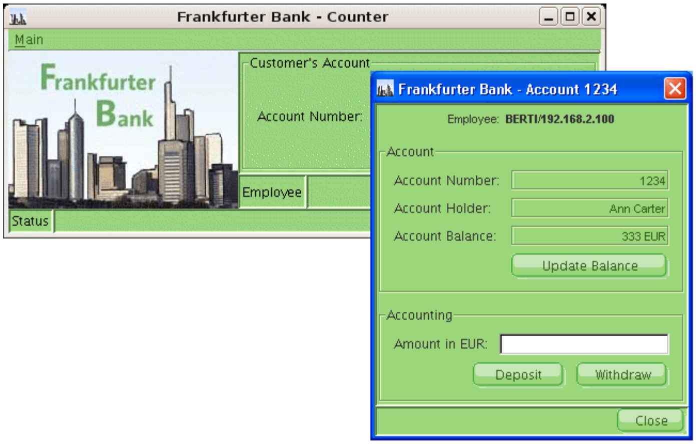


● IP-Address ○ UserID/Password ○ SSL (Keystore)

Fig. 10. Frankfurter Bank Example: Client Login Panel

On the “Configuration” tab you can configure the service URL (`http://127.0.0.1/:8080/axis/services/FrankfurterBankPort` by default) and whether the transfer panel shall be shown. This is explained in detail in the Administrator’s Guide, chapter “Protecting Web Services – Example”.

After successful authentication an interactive window will appear, where the account managers can place their queries.



Employee: BERTI/192.168.2.100

Account Number: 1234

Account Holder: Ann Carter

Account Balance: 333 EUR

Fig. 11. Select an account

Deposit and withdraw money from the two accounts as you like and update the balance by pressing the “Update Balance” button.

Now that the application has been developed and passed all tests, it shall be installed in "real world" environment. To meet the strict security requirements you have decided to protect the web service with the WS-DBC. These security requirements include that only the account managers may deposit and withdraw money from the accounts and to do so they have to authenticate using SSL or UserID/Password authentication. The Frankfurter Bank application itself has no access control implemented. So right now everybody can do everything!

The following sections present how to run the Frankfurter Bank web service with the DBC (without any access control). In section "Protecting an Application with the WS-DBC" on page 28, the Admin Console will be used to define an access control policy which will be enforced by the WS-DBC.

3.3 *Running Frankfurter Bank With the WS-DBC*

To make a service available through the WS-DBC, you have to

1. **Add the resource (Service) to the Security Policy:**
   Make a new entry for the protected target service (optional WSDL import). Policy details are defined by editing the resource properties.
2. **Define a Resource Mapping:**
   The WS-DBC Proxy provides a *virtual service endpoint* for clients that want to access the target resource. This virtual service address is a URL that consists of the WS-DBC Proxy's address and the logical service name (see figure 12, "Frankfurter Bank Client and Server: Connecting through the WS-DBC"). The resource mapping defines how the virtual service address is mapped to the actual server address, so that a request to the virtual address can be passed on to the actual service address by the WS-DBC Proxy.

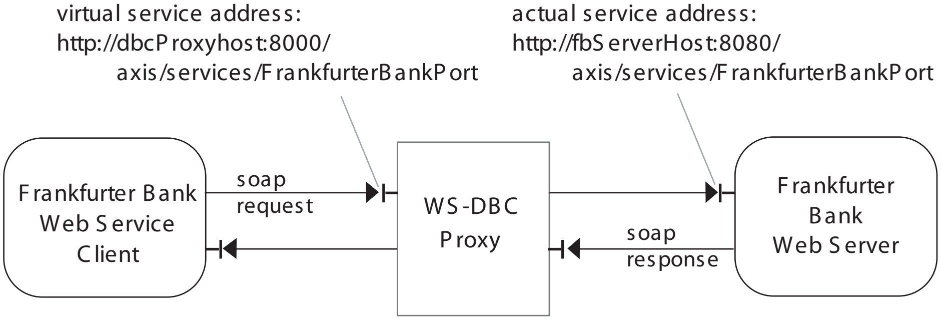

virtual service address:
http://dbcProxyhost:8000/
axis/services/FrankfurterBankPort
actual service address:
http://fbServerHost:8080/
axis/services/FrankfurterBankPort
Frankfurter Bank Web Service Client
soap request
WS-DBC Proxy
soap response
Frankfurter Bank Web Server

Fig. 12. Frankfurter Bank Client and Server: Connecting through the WS-DBC

Let’s Get Started!

Start the SPS, WS-DBC, and Admin Console:

- to start the Security Policy Server (as root): `rcxdn_sps start`
- to start the Security Policy Server (as root): `rcxdn_wsdbc start`
- to start the Admin Console: `<INSTALLDIR>/AdminConsole/bin/AdminConsole` (on Linux) or `AdminConsole.exe` (on Windows).
- login with the Admin Console to the SPS (enter the hostname or IP address and port number of the SPS (e.g., `DBCHost:15000`) and use the default user `admin` with the password `admin`).

Make the Frankfurter Bank Web Service available through the WS-DBC

As a first step we want to grant complete access to the whole service with all the operations. In a second step we will define a more fine-grained access control policy (section “Define a more fine-grained policy” on page 33).

A resource (representing the target service) can be added to the security policy manually or by importing the service’s WSDL. As the Frankfurter Bank web service’s WSDL is included in the installation, we will import it.

The WSDL file can be found in `wsdl/FrankfurterBank.wsdl`. The WSDL is depicted in figure 13.

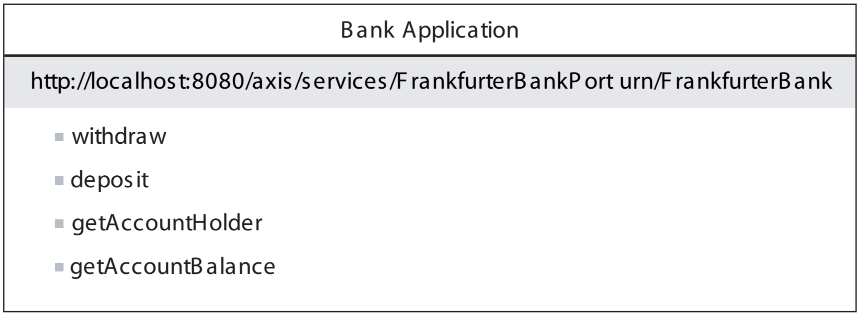
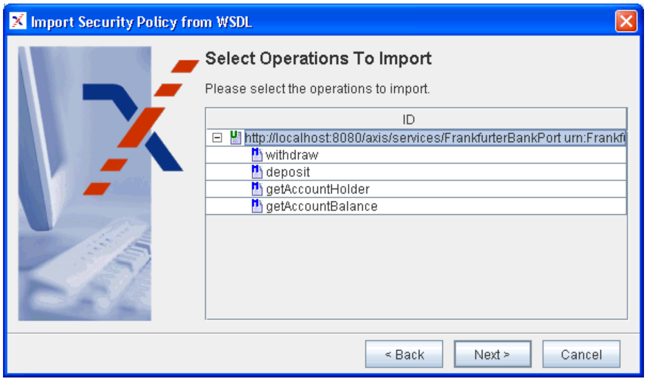

Bank Application
http://localhost:8080/axis/services/FrankfurterBankPort urn/FrankfurterBank
withdraw
deposit
getAccountHolder
getAccountBalance

Fig. 13. Frankfurter Bank WSDL: Types and operations

To import the application’s WSDL file, bring up the Admin Console, choose **File ➜ Import WSDL....** and select `wsdl/FrankfurterBank.wsdl` in the file chooser. Press the “Next” button and then follow the WSDL Import Wizard. As we want to grant access to all operations of the WSDL it is sufficient to import only the type, not the operations (as depicted in figure 14).



Select Operations To Import

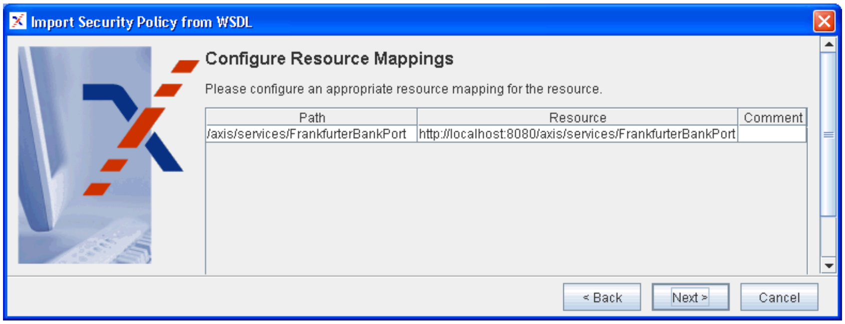
Please select the operations to import.

ID
http://localhost:8080/axis/services/FrankfurterBankPort urn:Frankfu
withdraw
deposit
getAccountHolder
getAccountBalance

Fig. 14. Select Resources to Import

Configure a Resource Mapping

The next step is to configure a resource mapping to provide a virtual service endpoint to service clients. Accept the default suggested by the Import Wizard. You can later view and edit the resource mapping on the “Resource Mappings” panel (below the “WS-DBC Proxy” panel).



Configure Resource Mappings

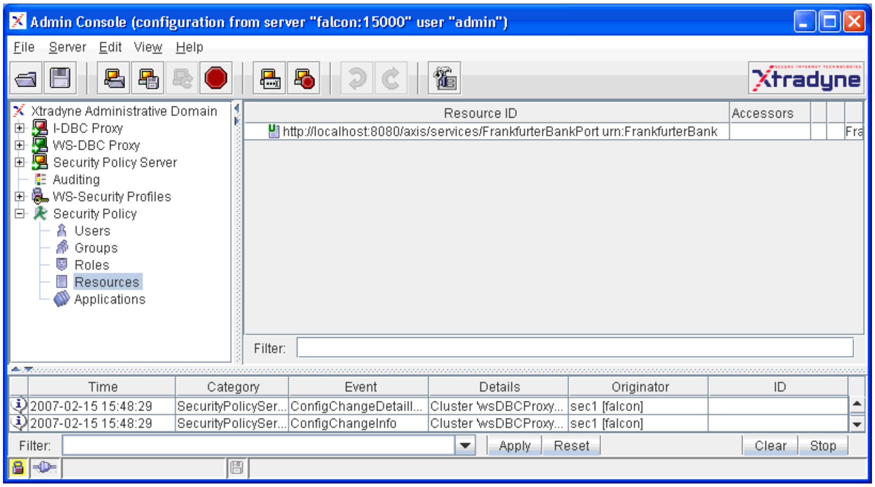
Please configure an appropriate resource mapping for the resource.

Path	Resource	Comment
/axis/services/FrankfurterBankPort	http://localhost:8080/axis/services/FrankfurterBankPort	

Fig. 15. Configure a Resource Mapping

Note that you may skip the “Create Application” panel of the Import Wizard.

After completing the WSDL import you can view the resource by selecting “Resources” in the navigation tree on the left side of the Admin Console Window (figure 20).



Admin Console (configuration from server "falcon:15000" user "admin")

- Xtradyne Administrative Domain
  - I-DBC Proxy
  - WS-DBC Proxy
  - Security Policy Server
  - Auditing
  - WS-Security Profiles
  - Security Policy
    - Users
    - Groups
    - Roles
    - Resources
    - Applications

Resource ID	Accessors			
http://localhost:8080/axis/services/FrankfurterBankPort urn:FrankfurterBank				Fra

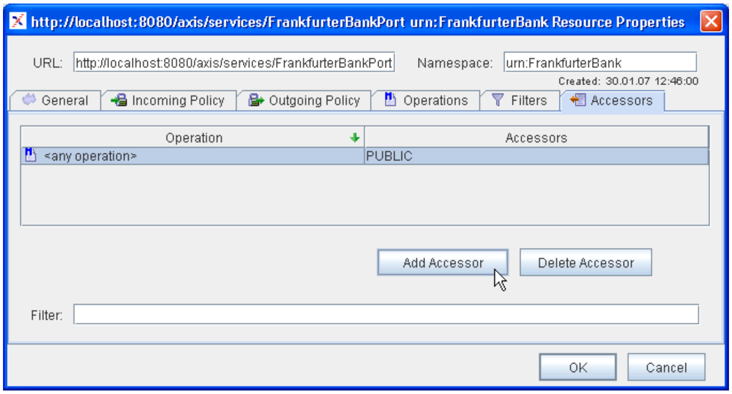
Filter:

Time	Category	Event	Details	Originator	ID
2007-02-15 15:48:29	SecurityPolicySer...	ConfigChangeDetaill...	Cluster wsDBCProxy...	sec1 [falcon]	
2007-02-15 15:48:29	SecurityPolicySer...	ConfigChangeInfo	Cluster wsDBCProxy...	sec1 [falcon]	

Fig. 16. Import the Frankfurter Bank WSDL

Grant PUBLIC access

For testing purposes we will grant public access to the Frankfurter Bank resource, then run the service across the WS-DBC and then, in a second step, define access rules. To grant public access, double-click on the Frankfurter Bank resource and select the “Accessors” tab. Select the `<any operation>` entry in the operation table and then press the “Add Accessor” button (cf. figure 18). Choose the pre-defined user “PUBLIC” from the list and press “OK”.



http://localhost:8080/axis/services/FrankfurterBankPort urn:FrankfurterBank Resource Properties

URL: http://localhost:8080/axis/services/FrankfurterBankPort Namespace: urn:FrankfurterBank

Created: 30.01.07 12:46:00

Operation	Accessors
<any operation>	PUBLIC

Fig. 17. Grant public access

Enable Audit Events

For better monitoring of the WS-DBC you can enable audit events with the Admin Console: Select the "Auditing" panel in the navigation tree on the left side of the Admin Console and press the "Enable All" button (by default only the failure events are enabled).

Save the configuration to the SPS (**CTRL-W** or **Server - Write to Security Policy Server**).

Running the Client

Start the Frankfurter Bank client and configure the target URL for our newly specified service on the "Configure" tab. The target URL is the URL where the WS-DBC can be reached, in our example: `http://<HOST>:8000/axis/services/FrankfurterBankPort`, where `<HOST>` has to be replaced by the host on which the WS-DBC is installed. This starts the client pointing it to the WS-DBC (as depicted in "Frankfurter Bank Client and Server: Connecting through the WS-DBC" on page 23). You can use authentication by IP-Address (as we defined PUBLIC access, it does not matter…).

To verify that the application really runs across the DBC you can watch the events in the event browser at the bottom of the Admin Console (double-click on the event to bring up the details). If you enabled all events before, you should see the events

- `Proxy.HTTPConnectionAcceptedSuccess` and
- `Proxy.HTTPConnectionEstablishedSuccess`.

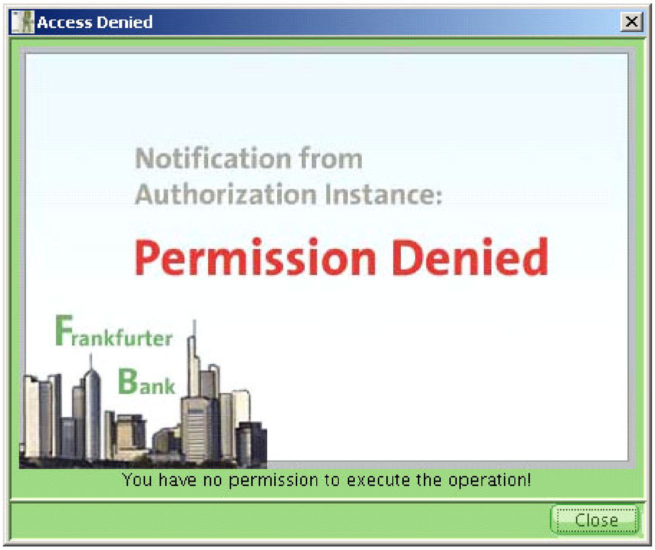
They show that the DBC accepted a HTTP connection from the Frankfurter Bank client on port 8000 (default plain TCP listener port) and that a HTTP connection was established from the DBC to the Frankfurter Bank Server (to port 8080). You should also see the events

- `Proxy.HTTPMessageSentInfo` and
- `Proxy.HTTPMessageReceivedInfo`

which show that HTTP Messages have been exchanged between client and server across the DBC.

Now, take away PUBLIC access for the Frankfurter Resource (on the "Resources - Accessors" panel) and restart the Frankfurter Bank Client. Login with authentication by

IP address. Try to view one of the two accounts. This will fail and a “No Permission” window will show up (see figure 18).



Notification from Authorization Instance:

Permission Denied

You have no permission to execute the operation!

Fig. 18. All access denied

This is the expected behavior as all calls are subject to access control by the WS-DBC but we have not defined any access rights yet. As a next step we will make the application accessible for the two account managers.

3.4 Protecting an Application with the WS-DBC

To define security policies for the protected resource you have to define access rules and authentication requirements that clients must respect:

1. **Define Users, Groups, and Roles**:
   As a prerequisite for defining access rules, you need to set up the user entities to which your policy rules refer. For ease of management, users can be grouped (e.g., teams, departments, etc.) and assigned to roles (for concepts see “Access Control” on page 215 of the Admin Guide).
2. **Define how the Resource shall be protected**:
   Define the required authentication and the required protection for clients accessing the resource.

3. **Define the Resource's Accessors**:
   Define which users, groups, or roles may access the resource or certain operations of the resource.

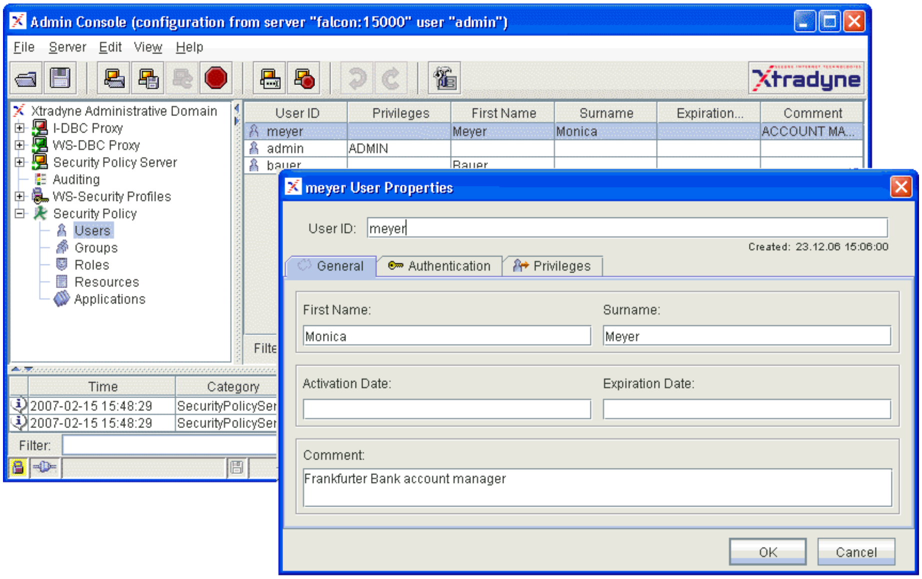
3.5 Define Security Policies for the Protected Resource

As a first step we want to give the account managers complete access to the whole service with all the operations. In a second step we will define a more fine-grained access control policy (section "Define a more fine-grained policy" on page 33).

For now we start by defining a user in the policy representing the account manager and define how this user shall authenticate. Then give this user permission to access the Frankfurter Bank resource.

Create Users

Go to the "User" panel and choose **New User** from the context menu. On the "General" tab enter the surname as the User ID (e.g., `meyer` for account manager Mrs. Meyer) and the activation and expiration dates as you like (e.g., an empty expiration date for an unlimited period).



User ID	Privileges	First Name	Surname	Expiration...	Comment
meyer		Meyer	Monica		ACCOUNT MA...
admin	ADMIN				
bauer		Bauer			

Time	Category
2007-02-15 15:48:29	SecurityPolicySer
2007-02-15 15:48:29	SecurityPolicySer

User ID: meyer

Created: 23.12.06 15:06:00

First Name: Monica

Surname: Meyer

Activation Date:

Expiration Date:

Comment: Frankfurter Bank account manager

Fig. 19. User entry for account manager Mrs. Meyer

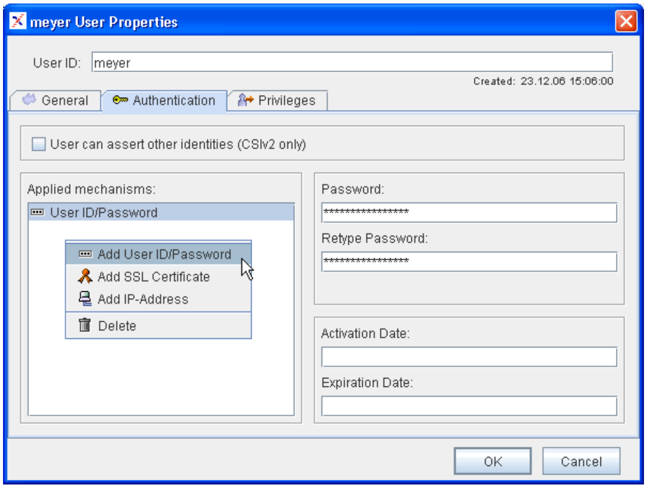
Authentication

Now go to the “Authentication” tab to define by which authentication method user meyer will have to authenticate.

The DBC offers several authentication mechanisms (listed from the strongest to the weakest):

- SSL authentication
- UserID/Password authentication (HTTP Basic Authentication)
- Authentication by IP-Address
- (no authentication)

Let’s define UserID/Password authentication for user `meyer`: Right-click into the “Applied mechanisms” region and choose “Add UserID/Password” from the context menu (as depicted in figure 20). Now enter a password and retype it.


meyer User Properties

User ID: meyer

Created: 23.12.06 15:06:00

□ User can assert other identities (CSIv2 only)

Applied mechanisms:
User ID/Password

Password: ****************

Retype Password: ****************

Activation Date:

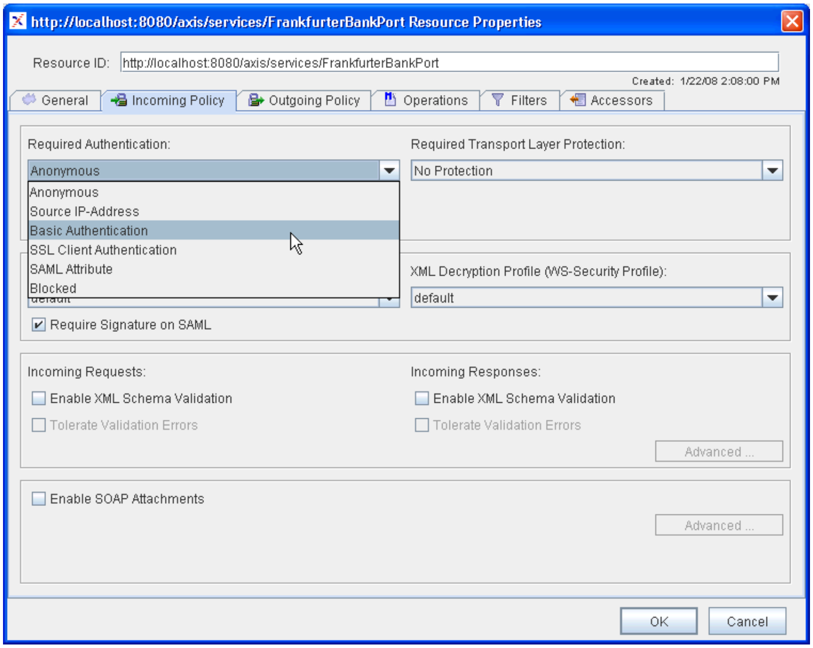
Expiration Date:

Fig. 20. Add authentication mechanisms

Required Authentication for the Protected Resource

The next step is to define a required authentication method for the resource, i.e., we want to make sure that only users that authenticate with User ID/Password authentication or a stronger authentication method (like SSL) may access the resource. To do so, double-click on the three resources in the resource table and select the “Incoming Policy” tab.

Choose “Basic Authentication” from the “Required Authentication” drop-down menu (as depicted in figure 21).



http://localhost:8080/axis/services/FrankfurterBankPort Resource Properties

Resource ID: http://localhost:8080/axis/services/FrankfurterBankPort

Created: 1/22/08 2:08:00 PM

Required Authentication: Anonymous

Required Transport Layer Protection: No Protection

XML Decryption Profile (WS-Security Profile): default

☑ Require Signature on SAML

Incoming Requests:

□ Enable XML Schema Validation

□ Tolerate Validation Errors

Incoming Responses:

□ Enable XML Schema Validation

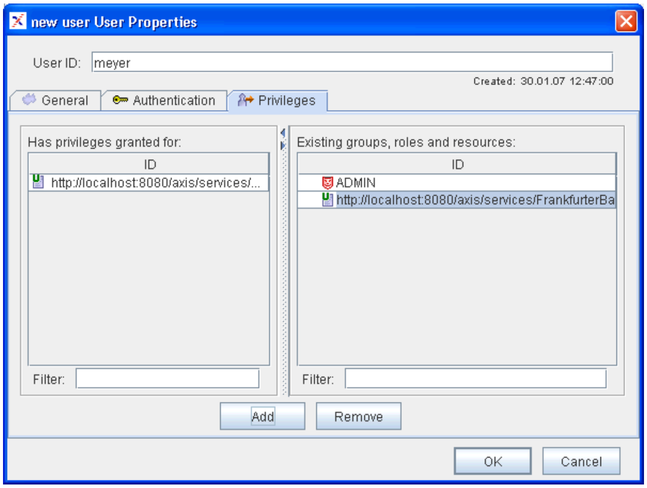
□ Tolerate Validation Errors

□ Enable SOAP Attachments

Fig. 21. Define User ID/Password Authentication as required authentication

Assign Permissions

As a last step we want to grant user meyer access to the resources. To do so, go to the “Users” table again and double-click on user `meyer`. Choose the “Privileges” tab, select the Frankfurter Bank resource and then press the add button (see figure 22). The resources will appear on the left side of the table.



new user User Properties

User ID: meyer

Created: 30.01.07 12:47:00

Has privileges granted for:

ID
http://localhost:8080/axis/services/...

Existing groups, roles and resources:

ID
ADMIN
http://localhost:8080/axis/services/FrankfurterBa

Fig. 22. Grant user `meyer` permission to access FB resources:

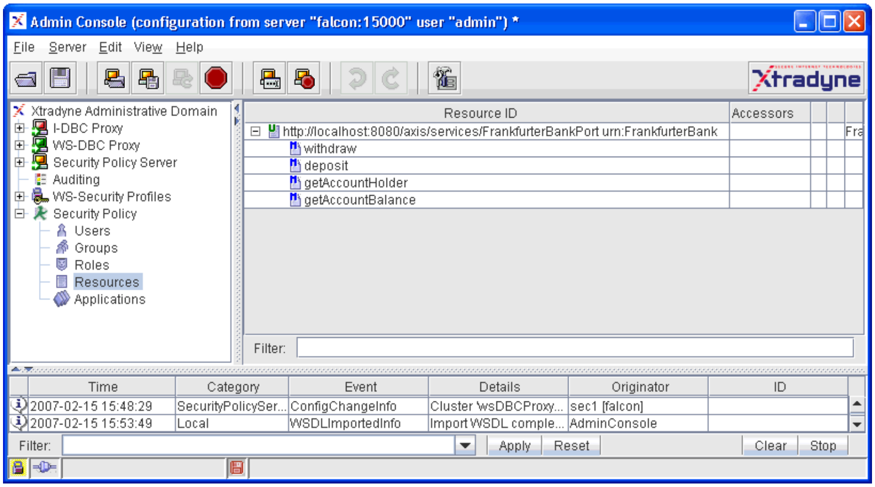
Click the OK button and don’t forget to write the configuration to the Security Policy Server! Let’s try if our simple policy works: Restart the Frankfurter Bank Client and choose UserID Password authentication. Enter the UserID and password of user `meyer`. View the accounts and deposit and withdraw money.

Try to login with a wrong password or another UserID, you should get the “No Permission” window again.

Also try the following: Define a user with IP-Address authentication and give him access to the resources. Try to login with this user and access the bank accounts. This should fail, as you defined that the minimal required authentication method for accessing the Frankfurter Bank resource is Basic Authentication (User ID/Password).

3.6 *Define a more fine-grained policy*

As a next step we want to do access control on operation level. To do so, we will import the application’s WSDL again - this time selecting all the operations. After import the resource appears on the “Resources” table.

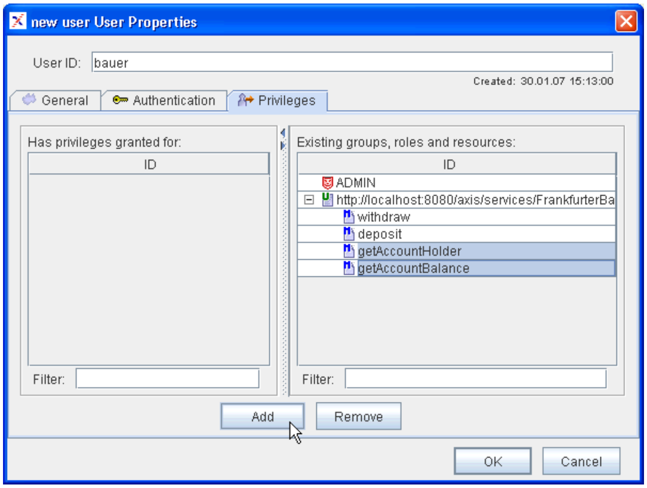


Resource ID	Accessors			
http://localhost:8080/axis/services/FrankfurterBankPort urn:FrankfurterBank				Fra
withdraw				
deposit				
getAccountHolder				
getAccountBalance				

Time	Category	Event	Details	Originator	ID
2007-02-15 15:48:29	SecurityPolicySer...	ConfigChangeInfo	Cluster wsDBCProxy...	sec1 [falcon]	
2007-02-15 15:53:49	Local	WSDLImportedInfo	Import WSDL comple...	AdminConsole	

Fig. 23. Frankfurter Bank Example: Imported Resource with operations

Now define a new user entry for account manager Mr. Bauer (e.g., with id `bauer`). Add UserID/Password authentication for this user and give Mr. Bauer all rights except for the operations deposit and withdraw. (Hint: deselect the operations withdraw and deposit by pressing the **CTRL** key while clicking on the operation, cf. figure 24).



User ID: bauer

Created: 30.01.07 15:13:00

ADMIN
http://localhost:8080/axis/services/FrankfurterBa
withdraw
deposit
getAccountHolder
getAccountBalance

Fig. 24. Grant permissions for operations

Now login as user bauer (with UID/Password authentication) and try to deposit and withdraw money. You should get the “No Permission” window.

3.7 Using SSL

Let’s run the application across the DBC using SSL. Note that giving a complete introduction on SSL is beyond the scope of this section, therefor, when reading this section you should be familiar with SSL basics.

Frankfurter Bank Keys and Certificates

The Frankfurter Bank uses its own PKI infrastructure with its own CA certificate. For convenience, the Frankfurter Bank application comes with two pre-defined keystores: one for each of the account managers (i.e., the clients). These keystores are located in the directory keystores/ in the Frankfurter Bank installation. The password for the keystores is “changeit”.

To list the contents of the keystores you can use the Java keytool:

```
keytool -list -keystore Meyer.jks
```

Each keystore contains the a key for the respective user and the CA certificate that signed the key.

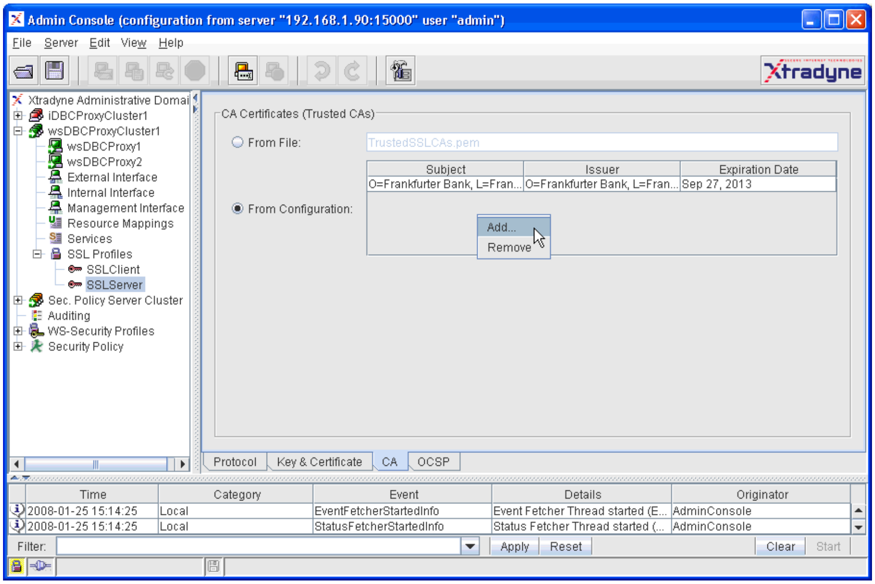
Establish Mutual Trust between DBC and FB Client

Up to now the DBC doesn’t know anything about the Frankfurter Bank’s CA certificate. So when a Frankfurter Bank client connecting to the DBC with SSL presents its key, the DBC won’t trust this client as the CA certificate that issued the key is unknown to the DBC. The same applies to the other direction: the Frankfurter Bank application doesn’t know anything about the DBC CA certificate, therefor it won’t accept the DBC as communication partner.

To establish mutual trust between the two communication partners the Frankfurter Bank CA certificate has to be added to the list of trusted CA certificates in the DBC. And the DBC’s CA certificate has to be added to the Frankfurter Bank Client’s keystore.

As a first step we have to add the Frankfurter Bank CA certificate (located in `keystores/FBCA.pem`) to the DBC’s list of trusted certificates. To do so, bring up the AdminConsole and go to the **WS-DBC Proxy Cluster ➜ SSLProfiles ➜ SSLServer** Panel and select the “CA” tab. In the “Certificate” section, select the “From Configuration” check box. Then right-click into the table and choose **Add...** from the context menu. Select

Frankfurter Bank’s CA certificate (`keystores/FBCA.pem`) in the file chooser. The CA certificate will appear in the table (cf. screenshot in figure 25).



Admin Console (configuration from server "192.168.1.90:15000" user "admin")

CA Certificates (Trusted CAs)

○ From File: TrustedSSLCAs.pem

● From Configuration:

Subject	Issuer	Expiration Date
O=Frankfurter Bank, L=Fran...	O=Frankfurter Bank, L=Fran...	Sep 27, 2013

Time	Category	Event	Details	Originator
2008-01-25 15:14:25	Local	EventFetcherStartedInfo	Event Fetcher Thread started (E...	AdminConsole
2008-01-25 15:14:25	Local	StatusFetcherStartedInfo	Status Fetcher Thread started (...	AdminConsole

Filter:

Fig. 25. Import the Frankfurter Bank CA Certificate

Save the new configuration to the SPS.

As the next step add the `ProxyCACert.pem` (located in `<INSTALLDIR>/wsdbc/etc/`) to Meyer’s and Bauer’s keystore. You may use the script `replaceDBCCert` located in the Frankfurter Bank’s `bin` directory:

On Linux/Solaris:

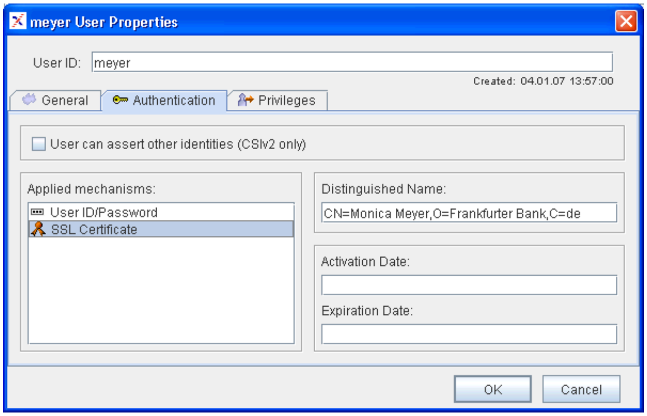
```
replaceDBCCert.bin ProxyCACert.pem
```

On Windows

```
replaceDBCCert.bat ProxyCACert.pem
```

Adapt the Access Control Policy

Now both communication partners trust one another. The next step is to adapt the access control policy, i.e., to add SSL as authentication mechanism for user `meyer`. To do this go to the user list and double-click on the user `meyer`. Then select the “Authentication” tab and right-click into the “Applied mechanisms” area. Select “SSL Certificate” and then configure the Distinguished Name (as depicted in figure figure 26).



meyer User Properties

User ID: meyer

Created: 04.01.07 13:57:00

□ User can assert other identities (CSIv2 only)

Applied mechanisms:

- User ID/Password
- SSL Certificate

Distinguished Name: CN=Monica Meyer,O=Frankfurter Bank,C=de

Activation Date:

Expiration Date:

Fig. 26. Add SSL as authentication mechanism for user `meyer`

The Distinguished Name (DN) has to be the same as the one contained in Mrs. Meyer’s key (i.e., “CN=Monica Meyer,O=Frankfurter Bank,C=de”).

You can view the DN by running the script `printcert.sh` on Linux/Solaris or `printcert.wsf` on Windows. The script is located in the `bin` directory. On Windows, type for example:

```
printcert.wsf ..\keystores\Meyer.jks
```

Alternatively, you may use the Java keytool. First export Mrs. Meyer’s key from the keystore and then print it:

```
keytool -export -alias monicameyer -keystore Meyer.jks
        -storepass changeit -file MeyerCert.pem
```

To print out the key:

```
keytool -printcert -file MeyerCert.pem
```

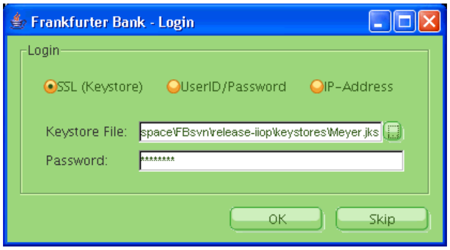
After entering the DN in the AdminConsole’s User Properties panel, press the OK button and save the configuration to the Security Policy Server.

Start the Client with SSL

Start the Frankfurter Bank Client with `https`. For example:

`FBClient -u https://<HOST>:8001/axis/services/FrankfurterBankPort`, where `<HOST>` has to be replaced by the host on which the WS-DBC is installed (the port 8001 is the default SSL port of the WS-DBC Proxy).

Choose “SSL (keystore)” as authentication method. Then select a keystore (e.g. `keystores/Meyer.jks`) with the file selector and enter the keystore password (“changeit”).



Frankfurter Bank - Login

Login

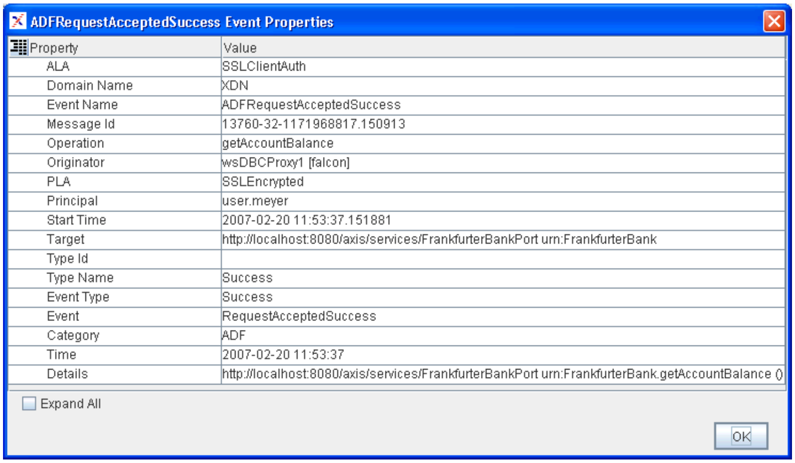
● SSL (Keystore) ○ UserID/Password ○ IP-Address

Keystore File: space\FBsvn\release-iiop\keystores\Meyer.jks

Password: ********

Fig. 27. Frankfurter Bank Client: SSL Authentication

In the Admin Console’s event browser you will find the “ADFRequestAcceptedSuccess” event which indicates that the DBC’s Access Decision Function (ADF) granted access. Double-clicking on the event will bring up the event details panel depicted in figure 28.



ADFRequestAcceptedSuccess Event Properties

Property	Value
ALA	SSLClientAuth
Domain Name	XDN
Event Name	ADFRequestAcceptedSuccess
Message Id	13760-32-1171968817.150913
Operation	getAccountBalance
Originator	wsDBCProxy1 [falcon]
PLA	SSLEncrypted
Principal	user.meyer
Start Time	2007-02-20 11:53:37.151881
Target	http://localhost:8080/axis/services/FrankfurterBankPort urn:FrankfurterBank
Type Id	
Type Name	Success
Event Type	Success
Event	RequestAcceptedSuccess
Category	ADF
Time	2007-02-20 11:53:37
Details	http://localhost:8080/axis/services/FrankfurterBankPort urn:FrankfurterBank.getAccountBalance ()

□ Expand All

Fig. 28. Event Properties: ADFRequestAcceptedSuccess

Among other information, the event details show that the requested operation is: `getAccountBalance` and that the principal was recognized as user `meyer`.

We suggest that you play around a bit with the example. For a more detailed example on how to define access control policies, please refer to (chapter "Deploying Web Services – An Example" on page 305 of the Admin Guide).

3.8 Summary

We have seen how to define a very simple security policy. The DBC offers to define far more complex security policies. For example, the WS-DBC can perform content inspection by validating the parameters of incoming messages. Flexible filter rules for operation parameters can be defined with the Admin Console. Messages carrying arguments that do not conform to the filter rules will be rejected. For more details, please refer to the Administrator's Guide.