SECURE INTERNET TECHNOLOGIES

Xtradyne

Web Services Domain Boundary Controller v 3.1

Administrator's Guide

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

Contents 3

Preface 19

Audience 19

Document Conventions 21

PrismTech Customer Support 22

When contacting customer support 22
How to contact PrismTech customer support 22
Encrypting DBC Configuration Files for Support 22
Making Screenshots for Support 22

Part 1 Concepts and Components

Chapter 1 Introduction 25

1.1 Web Services and SOAP 25
1.1.1 Web Services Security 29
1.1.2 Message-oriented Security and Security Assertions 29
1.2 Securing Web Services with the WS-DBC 31
1.2.1 Product Features 32
Exporting Policies in XACML format 33
1.3 WS-DBC Components 34
1.3.1 WS-DBC Proxy 34
1.3.2 Security Policy Server 36
1.4 Scalability and High Availability 37
1.4.1 Scalability 37
1.4.2 High Availability 38
High Availability with Hot Standby 38

High Availability with Traffic Redirection 39
1.5 Features of the WS-DBC Enterprise Edition 40
1.6 References 40

Chapter 2 Security Functionality and Use Cases 41

2.1 Introduction 41
Auditing 42
2.2 Use Cases 42
2.3 Use Case 1 – Single target-side WS-DBC 43
2.3.1 Message Authentication 43
2.3.2 Content Inspection and Access Control 44
Content Inspection 44
Access Control 45
SAML Injection 45
Message Protection 46
Summary 46
2.4 Use Case 2 – Sender-side and target-side WS-DBCs 47

Chapter 3 Advanced XML Processing 49

3.1 Web Services Description Language (WSDL) 49
3.1.1 Using the WS-DBC in combination with WSDL 50
3.1.2 WSDL and HTTP GET requests 51
3.2 XML Schema Validation 51
3.3 Configuration of XML Schema Validation 53
Schemas importing other Schemas or DTDs 54
3.3.1 Limitations of Schema Validation 54
3.4 Finding out which schemas are necessary 56
3.4.1 Writing a Schema file 57
3.4.2 References 58

Part 2 Installation and Configuration

Chapter 1 Preparing to Install 61

1.1 Prerequisites 61
1.1.1 Component Specifications 62
DBC Proxy 62
Security Policy Server 62
Administration Console 62
1.2 Typical Deployment Scenarios 63
1.2.1 Internal Communication Links and Trust Relationships 63
1.2.2 Scenario 1: Screened Host Firewall with WS-DBC Proxy 65
Firewall Configuration 66
1.2.3 Scenario 2: WS-DBC Proxy in the DMZ 66
Exterior Firewall Configuration 67
Interior Firewall Configuration 68

Chapter 2 Installing the DBC 69

2.1 Introduction 69
2.1.1 Mounting the CD ROM 71
2.2 Installer 72
2.2.1 Prerequisites 72
2.2.2 Choosing the Install Task 72
2.2.3 Choose the Install Folder 73
2.2.4 Choose a DBC User Name 74
2.2.5 Security Policy Server Configuration 74
2.2.6 Choose SSL Key Pair Information 75
2.2.7 DBC Proxy Configuration 76
2.2.8 Choose Key Archive 77
2.2.9 Choose License File 77
2.2.10 Configuration Summary 78
2.2.11 Uninstalling 78
2.2.12 What is Installed and Where: Security Policy Server 78
2.2.13 What is Installed and Where: DBC Proxy 80

2.2.14 Startup, Shutdown, and Restart 82
Security Policy Server 82
DBC Proxy 83
Determining the Status of the DBC Proxy or SPS 84
Finding out the Status with Linux/Unix Commands 85
2.3 Advanced Installation 85
2.4 Installation Steps 86
2.5 Installing the Security Policy Server (SPS) 87
2.5.1 Postinstallation Steps 88
Installing the License 88
Initial Configuration of the Management Network Interface 89
Adding a Proxy 89
Generating Keys 90
Overview of Keys and Certificates on the SPS 91
Generating Keys for a Cluster of Security Policy Servers 91
Checking Permissions for Key Files 92
The DBC Installation Account 92
2.6 Installing the DBC Proxy 93
2.6.1 Postinstallation Steps 95
Installing Keys and Certificates on the DBC Proxy Host 95
Overview of Keys and Certificates on the DBC Proxy 95
Initial Configuration of the Security Policy Server Interface 96
Starting the DBC Proxy 96
2.7 Deinstalling the Software 97

Chapter 3 Installation of the Admin Console 99

3.1 What's included in the Distribution? 99
3.1.1 Mounting the CD ROM on Linux and Solaris 99
3.2 Installation Steps 100
3.2.1 What is installed and where on Linux, Windows, and Solaris 101
3.3 Installing SSL Keys 101

Chapter 4 Configuration of DBC components 103

4.1 Admin Console – Introduction 103

4.2 Quick Start: Typical Configuration Steps 104
4.3 Administration Concepts 104
4.3.1 Configuration Data 105
Policy Versioning and Roll-back 105
Restoring Administrative Access Control Rules 106
4.3.2 Importing Configuration Files of previous DBC versions 107
4.3.3 Log Files and Log File Backup 108
4.4 First Start 109
Working Offline 110
4.4.1 Preferences 110
Configuring the Connection to the Security Policy Server 111
Configuring the Event Browser 111
Write Configuration – Properties 112
Passphrase Prompt 114
4.4.2 General Navigation 115
Tool Bar Icons 115
Admin Console Menus 116
4.5 General Organization of the Administration Console 117
4.5.1 Audit Event Browser 119
4.5.2 Activate a Configuration on the DBC Proxy or SPS (Cluster) ... 119
Conflicts When Writing to the SPS 120
When to Restart the DBC Proxy / Security Policy Server 120

Chapter 5 DBC Proxy Cluster Configuration 123

5.1 DBC Proxy Cluster 123
5.1.1 I-DBC Proxy Cluster - General 124
IIOP Proxy Engines 125
Security 125
5.1.2 I-DBC Proxy Cluster – CSIv2 126
Target Security Service 126
Client Security Service 127
5.1.3 I-DBC Proxy Cluster - PAM 127
5.1.4 I-DBC Proxy Cluster - Advanced 129
Preferred Security Policy Server 129

Access Session Management 130
Setting GIOP Connection Timeouts 130
5.2 WS-DBC Proxy Cluster 131
5.2.1 WS-DBC Proxy Cluster – General 132
WS-DBC Proxy Cluster Name 132
Number of SOAP Proxy Engines 132
Maximum Size of an HTTP Message 133
Maximum Start line of an HTTP Message 133
Verbosity Levels on HTTP/SOAP Errors 133
5.2.2 WS-DBC Proxy Cluster – PAM 133
5.2.3 WS-DBC Proxy Cluster – Advanced 133
Preferred Security Policy Server 134
5.3 DBC Proxy Configuration 135
5.3.1 DBC Proxy 136
5.3.2 DBC Proxy Network Interfaces 136
5.3.3 NAT Addresses for I-DBC Proxy Interfaces 138
5.4 External and Internal Interface Overview 140
5.5 External Interface 141
5.5.1 Virtual Address 142
5.5.2 Acceptors 143
Acceptor Details 144
SSL Acceptor Details 145
5.5.3 NAT Port Mappings for the I-DBC Proxy 145
Defining Port Mappings for the I-DBC Proxy Cluster 145
Bind Acceptors to NAT Addresses for Direct Routing. 146
5.5.4 Connectors 146
5.5.5 External I-DBC Interface - Advanced 147
5.6 Internal Interface 148
5.7 Management Interface 149
5.8 Replication Interface 149
5.9 Resource Mappings 150
5.10 Services 152
DBC Monitoring 152
I-DBC: CORBA Name Service 152

I-DBC: Flow Control 153
Connection Limiter 153
5.11 IOR Proxification 154
5.11.1 Initial IORs 154
5.11.2 Advanced Features 157
Editing the Proxified Object Key 157
Editing the Proxified Type ID 157
5.11.3 Proxification Options 158
Proxification Options 159
Pass Through Options 159
5.12 Address Translation 159
5.12.1 Outgoing Connections to Servers 160
Configuring Address Mappings for Outgoing Connections 161
Defining and Deleting Address Mappings 161
5.13 Services 163
DBC Monitoring 163
I-DBC: CORBA Name Service 163
I-DBC: Flow Control 164
I-DBC: Connection Limiter 164

Chapter 6 SSL and WS-Security Profiles 165

6.1 SSL Profiles 165
6.1.1 SSL Profiles – Protocol 166
Profile Name 167
SSL Version 167
Ciphersuite 168
Peer Authentication 168
6.1.2 SSL Profiles – Key & Certificate 169
Private Key 170
6.1.3 Trusted CAs 171
6.1.4 SSL Profiles – OCSP 171
6.1.5 Importing Keys and Certificates from JAVA-Keystore 173
6.2 WS-Security Profiles 175
6.2.1 Signature Creation Profile 176

General Tab 176
Private Key 176
Public Key Certificate 177
Advanced Tab – Signature Settings 177
6.2.2 Signature Verification Profile 178
General Tab 178
Advanced Tab 179
6.2.3 XML Encryption Profile 180
6.2.4 XML Decryption Profile 181
Private Key 181

Chapter 7 Security Policy Server (Cluster) 183

7.1 Single SPS and SPS Cluster 183
7.2 Security Policy Server Cluster Properties 184
7.3 Security Policy Server 185
7.3.1 Security Policy Server Name 185
7.3.2 Management Network Interface 186
7.3.3 NAT between the DBC Proxy and the SPS 187
Using Host Names or IP Addresses 187

Chapter 8 Audit Policy 189

8.1 Introduction 189
8.1.1 Audit Events 189
Audit Event Types 190
8.1.2 Event Flow 190
8.2 Audit Policy 190
Audit Event Categories 192
8.2.1 Event Consumer 193
8.2.2 Event Priorities 194
8.3 SNMP Support 194
8.3.1 Mapping between events and trap messages 194
8.3.2 Activating SNMP trap generation 195
8.3.3 Customizing HP Openview NNM Alarm Browser 195

Chapter 9 Expert Mode 197

9.1 Dictionaries - An Introduction 197
9.2 The Dictionary Explorer 197
9.3 Editing Entries 198
9.4 Insert New Entries 199
9.5 Copy, Cut, Paste, and Delete Entries 199
9.6 Importing and Exporting Dictionaries 199

Chapter 10 Installing Keys and Certificates 201

10.1 Trust Establishment 202
10.2 Certificates and Certification Authorities 202
10.2.1 Trust Stores and Trusted CAs 203
Trust Store for the Application Connections 203
Trust Store for the Control and Admin Connections 203
10.2.2 Application Connections 204
Making the DBC Proxy Trust External Certificates 205
Integrating the DBC with Applications 206
Client Keystores 206
10.2.3 Control and Administration Connections 207
Checking the Validity of Keys and Certificates 210
10.3 Replacing Keys and Certificates 210
10.3.1 Replacing External and Internal Proxy Keys 211
Defining SSL Profiles 211
Selecting SSL Profiles for Specific Interfaces 211
An Example SSL Profile 211
Protocol 212
Key and Certificate 212
CA Certificates (Trusted CAs) 214
10.3.2 Defining WS-Security Profiles 214
Checking Permissions for Key Files 215
10.3.3 Replacing Control and Administration Connection Keys 215
10.3.4 Changing the Certificate Encoding Format 216

Chapter 11 Troubleshooting 217

11.1 Encrypting the DBC Configuration File for Support 217
11.2 How to Diagnose Problems: Logging 217
11.2.1 Determining the DBC Proxy / SPS Status 218
Useful Status Scripts 218
11.3 Scripts Do Not Work 219
11.3.1 Checking Permissions for Key Files 220
11.4 When to Restart the DBC Proxy / SPS 220
11.5 Frequently Encountered Problems 220
Access Denied 220
11.5.1 Internal Server Error 221
11.6 SSL Connection Problems 221
SSLAuthenticationCertificateFailure 221
SSLTransportHandshakeFailure 221
SSLTransportCertificateFailure 222
11.7 Callback Configuration 222
11.8 License Errors 223
11.9 Problems with the Installers 224
11.9.1 Installer Exits with Exception (Linux/Solaris) 224
11.9.2 Installer does not start 224
Access Control on the X Display 224
Included VM Could Not be Unarchived 224
JAVA_FONTS Variable Not Set Correctly 225
11.10 Admin Console 225
11.10.1 Problems With Starting the Admin Console 225
11.10.2 Linux, Solaris Startup 226
11.10.3 Problems With Logging On to the SPS 226
Server not reachable (`org.omg.CORBA.TRANSIENT`) 226
Server not reachable (`org.omg.CORBA.COMM_FAILURE`) . 227
User has no access (`org.omg.CORBA.NO_PERMISSION`) . 227
11.10.4 Problems with Adding SSL Certificates 227
11.11 Miscellaneous 227
11.11.1 Firewall Configuration – TCP Connection Timeouts 227

Finding out and setting TCP keep-alive times 228
11.12 Using Logrotate on Solaris causes sparse DBC log files 229
Workaround 229
11.13 My CORBA Application Doesn't Run With the I-DBC 230

Part 3 Managing Security Policies and Deploying Web Services

Chapter 1 Security Policies 235

1.1 Access Control 235
1.1.1 Access Control Policy 235
1.2 Defining Security Policies 237
1.3 Security Policy 238
1.3.1 Security Policy Storage 238
Use LDAP 238
LDAP Server 239
1.3.2 LDAP Server: Prerequisites 239
DBC Base DN 239
LDAP Account 239
iPlanet: Prerequisites 239
Active Directory: Prerequisites 240
Generic LDAP 240
1.3.3 Configuring the LDAP Server 240
1.4 General Navigation 241
Ambiguous User/Group/Role IDs 241
1.5 Users 241
1.5.1 User Properties – General 242
1.5.2 User Properties – Authentication Mechanisms 242
CSIv2 (I-DBC Proxy only) 243
Authentication via SAML Assertion 243
SSL X.509 Certificate Authentication 243

User ID/Password Authentication 244
Changing the Admin User's Password 245
IP Address 245
Using the Subnet Mask 246
Number of Access Sessions and Access Session Hierarchy 247
1.5.3 User Properties – Privileges 250
1.5.4 User Properties – Constraints 251
Activation/Expiration Date 252
Time Frame Constraint 253
1.6 Groups 254
1.6.1 Group Properties – General 256
1.6.2 Group Properties – Members 256
1.6.3 Group Properties – Privileges 257
1.7 Roles 258
1.7.1 Role Properties – General 259
1.7.2 Role Properties – Actors 259
1.7.3 Role Properties – Permissions 260
1.7.4 Role Properties – Administration 261
1.8 Resources 263
Adding a Resource 263
1.8.1 WSDL Exposure Wizards 264
1.8.2 Resource Properties 266
1.8.3 Resource Properties – General 267
Resource ID 267
Use Resource ID as Service Locator 267
Allow WSDL GET Requests 268
1.8.4 Resource Properties – Incoming Policy 269
Required Authentication 269
Required Transport Layer Protection 270
Process HTTP Basic Authentication 270
Process UsernameToken Authentication 270
Signature Keys / XML Decryption Keys 271
Require Signature on SAML 271
Incoming Requests/Incoming Responses 271
Preparing Schema Files for Use by the WS-DBC 272

The wsdl2schema and schematest Utility 272
Schemas Included in the WS-DBC Installation 273
Enable SOAP Attachments 273
1.8.5 The Xtradyne SOAP Attachment Filter Servlet 274
1.8.6 Resource Properties – Outgoing Policy 277
Sign Requests 277
Encrypt Requests 279
Authentication Token 279
Inject SAML Assertion 279
Include SAML Assertion in Signature 280
Preserve HTTP Authorization Header 280
Preserve UsernameToken Authorization Header 280
Timestamp Creation 280
Verbosity Levels 281
1.8.7 Resource Properties – Interface 281
Interface Properties 282
Operations Table 283
Add/Edit an Operation 283
Operation 283
SOAP 283
SOAP Requests / SOAP Responses 284
Define Operation Parameters for Web Service Resources 284
1.8.8 Resource Properties – Filters 285
Creating Filter Expressions 286
Filter Properties 286
Standard Filter Terms 286
XPath Expressions 287
Combining Filter Expressions 288
Filter Templates 288
1.8.9 Exporting and Importing Filter 289
1.8.10 Resource Properties – Accessors 291
Configuring Accessor Lists 291
Public Access 293
1.9 Applications (Application Domains) 295
Adding Applications 295

Application Properties – General 296
Application Properties – Administration 296
Example 298
Moving Global Roles and Resources to an Application 299

Chapter 2 Regular Expressions 301

2.1 WS-DBC – Regular Expression Syntax 301
2.1.1 Quantifiers 302
2.1.2 Atoms 302
2.1.3 Character Classes 303
2.1.4 Single Character Escapes 304
2.1.5 Category Escapes 304
2.1.6 Block Escape 306
2.1.7 Multi-Character Escapes 306
2.1.8 Examples 307
Define a Regular Expression for an IP-Address 307
Define a Regular Expression for a Date 307
Regular Expressions for Sets 308

Chapter 3 Protecting Web Services – Example 309

3.1 Model Engineering 310
Extended Model 310
Define Accessors to the protected Resources 312
3.2 Content Filtering 314
Parameter Filter Rules for the `transfer` operation 315
Parameter Filter Rule for the Date Format 317

Part 4 Appendices

Appendix A Audit Events 321

A.1 Events in alphabetical order 321

A.2 Important Audit Events 332
ADFRequestDeniedFailure 332
AuthenticationBasicAuthenticationFailure 333
AuthenticationSAMLAssertionFailure 333
PolicyRepositoryRetrieveFailure 333
ProxyResourceMappingFailure 334
ProxyHTTPConnectionFailure 334
ProxyInvalidHTTPFailure 335
SSLAuthenticationCertificateFailure 335
SSLTransportHandshakeFailure 336
XMLSchemaValidationFailure 337
XMLDSigVerificationFailure 337

Appendix B Error Messages and System Exceptions 339

B.1 DBC and SPS Error Messages 339
B.1.1 Error Message Format 340
B.1.2 Common Error Messages 340
License not found 341
Address lookup failed 341
Error while reading key file 341
Address already in use 342
Server Socket bind failed 342
System Exception “No Permission” 342
Cannot set SSL private key 343
Invalid key file format 343
Cannot open SSL certificate file 343
Unable to get local issuer certificate 344
Client does not accept Server’s certificate 344
B.2 CORBA System Exceptions and Minor Codes 344
B.2.1 BAD_OPERATION 345
B.2.2 BAD_PARAM 345
B.2.3 COMM_FAILURE 346
B.2.4 INITIALIZE 347
B.2.5 INTF_REPOS 347
B.2.6 IMP_LIMIT 348

B.2.7 INTERNAL 348
B.2.8 NO_PERMISSION 349
B.2.9 MARSHAL 350
B.2.10 NO_RESOURCES 351
B.2.11 NO_IMPLEMENT 351
B.2.12 OBJECT_NOT_EXIST 352
B.2.13 TRANSIENT 352
B.3 HTTP Error Messages 353
B.4 SOAP Error Messages 353

Appendix C SSL Ciphers 355

C.1 Cipher Suite String Format 355
Cipher Strings 356
C.2 Cipher Suites Offered by the DBC 358

Index 359

Preface

Existing firewall installations cannot adequately protect XML Web Service applications because these use HTTP. While HTTP as a transport protocol enables the flexible integration of resources into interoperable applications, firewalls will allow HTTP to pass through uninspectedly. Without additional protection, severe security breaches are possible that would compromise the integrity, confidentiality, or availability of critical company data.

The *Xtradyne Web Services Domain Boundary Controller™ (WS-DBC)* provides a comprehensive security solution for Web Services. It enables Web Services-based applications to interact across network boundaries.

The WS-DBC acts as a proxy server protecting SOAP-to-SOAP communication. It provides authentication, cryptographic message protection, fine-grained access control, content inspection, various XML-related checks, and security audit. Additionally, the Web Services DBC offers the insertion of SAML assertions into messages and the signing and verifying each of the SOAP messages.

Audience

This guide is aimed at administrators managing the Xtradyne Web Services Domain Boundary Controller environment. It comprises three parts:

- Part 1, “Concepts and Components” discusses the concepts of the WS-DBC including:
  - an overview of Web Services and related security issues (chapter 1 on page 25),
  - a description of the WS-DBC’s core functionality and use cases (chapter 2 on page 41),
  - a discussion of advanced aspects of XML Processing in the WS-DBC, viz. client bootstrapping using WSDL, and XML Schema Validation of SOAP messages (chapter 3 on page 49).
- Part 2, “Installation Guide” describes the set up and configuration of the WS-DBC System including:
  - a general product overview with examples for typical deployment scenarios (chapter 1 on page 61),

  - detailed installation instructions for the WS-DBC and the Security Policy Server (chapter 2 on page 69),
  - detailed installation instructions for the Administration Console (chapter 3 on page 99),
  - configuration instruction for the DBC using the Admin Console (chapters 4 to 9 starting on page 103),
  - description of trust relations between DBC components and the required keys and certificates used within the DBC (chapter 10),
  - a troubleshooting guide (chapter 11 on page 217).
- Part 3, “Managing Security Policies and Protecting Applications” on page 229 describes:
  - how access control policies are managed in the DBC and how to use the Admin Console to define such a policy (chapter 1 on page 235),
  - how to define regular expressions (chapter 2 on page 301),
  - the example application Frankfurter Bank that may be used to test a WS-DBC configuration (chapter 3 on page 309).
- Part 4, “Appendices” provides detailed descriptions of different topics related to the Domain Boundary Controller:
  - appendix A on page 321 lists the audit events used in the WS-DBC System,
  - appendix B on page 339 lists system exceptions and error messages,
  - appendix C on page 355 gives more background information on SSL and the ciphers recommended to be used with the WS-DBC.

For further reading, please refer to the Deployment Guide which discusses various topics related to the deployment of DBCs, like high availability and scalability requirements, hardening the operating system, performance monitoring etc.

Document Conventions

This guide uses the following typographical conventions:

This font	is used for:
`courier`	filenames and Unix commands
`courier`	URLs and e-mail addresses (e.g., `http://www.xtradyne.com`)
Arial Bold	menu selections / menu items and key-board short cuts (e.g., **CTRL-C**)
simple emphasis	new terms

PrismTech Customer Support

When contacting customer support

When contacting customer support please have the following information available:

- Your name, title, company name, email address, fax, and telephone number.
- Name and version of the product.
- Operating system and version.
- Severity level.
- Brief description of the problem.
- Details of any error messages or exceptions raised.

How to contact PrismTech customer support

PrismTech offers different levels of customer support. PrismTech customer support can be contacted by

- filling in the web form at:
  `http://www.xtradyne.com/services/problem_report.htm`
- sending an email to `support@xtradyne.com`

Furthermore PrismTech offers priority support – silver and gold support. Access information for silver or gold support are part of the support contract.

Encrypting DBC Configuration Files for Support

For diagnosing problems it might be helpful to send the DBC configuration file to the PrismTech support team. As configuration files contain confidential information like keys and certificates, you may use the **File ➜ Export ➜ Encrypt for support...** facility of the Admin Console. This will encrypt all the sensible information contained in the config file. For details please refer to page 217.

Making Screenshots for Support

Additionally, it might be helpful to send a screenshot of a certain configuration panel of the Admin Console to the PrismTech support team. To do this, you may use snapshot feature of the Admin Console, choose **Help ➜ Capture Screen** from the menu bar.

PART

1 *CONCEPTS AND COMPONENTS*

In this part of the administrator's guide, we present the central concepts of the Web Services DBC. Once these concepts have been explained, we describe how to deploy the Web Services DBC and administer Web Services security in parts 2 and 3.

The concepts covered in this part include:

- *A general introduction to the topic of Web Services and SOAP security (chapter 1).*
- *A description of the primary usage scenarios of SOAP security domain boundary control (chapter 2).*
- *A more detailed treatment of SOAP/XML processing (chapter 3).*

CHAPTER

1 Introduction

Modern enterprises can benefit from making existing applications available as Web Services, which offer easier integration and more cost-effective solutions, both for external and internal service users. However, the use of Web Services also causes a higher degree of exposure of what used to be internal data and functions – and thus results in higher risks for sensitive company assets. Therefore, the ability to control the risks of the open architectures induced by Web Services is a prerequisite for using Web Services successfully and securely.

This chapter provides an overview of Web Services and the related security issues and introduces the Web Services DBC, a complete security solution to protect enterprise resources that are exposed as Web Services.

1.1 Web Services and SOAP

Web Services and SOAP

Web Services are modular, self-contained software components that are accessed over "the Web". This means that Web Services are used by exchanging SOAP messages over the standard protocol HTTP. The format of SOAP messages and their binding to HTTP is defined by the W3C using the *Extensible Markup Language* (XML): SOAP messages

are XML documents embedded in an *envelope*. Figure 1 illustrates the general format of SOAP messages.

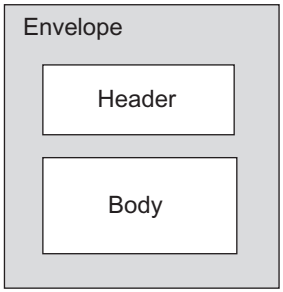


Fig. 1. SOAP message format

a Web Service example

As an example for applying Web Services, consider a company providing express delivery services in a city. The example company uses an internal management platform for placing orders, assigning delivery tasks, calculating per-delivery messenger compensations based on items and distances, and managing employee data. In order to extend the business such that freelance messengers or subcontractors may manage their compensations and calculate earnings from their external offices, the company decides to provide the management application as a Web Service. Figure 2 illustrates the flow of a SOAP request message from a messenger client to the order management service and back.

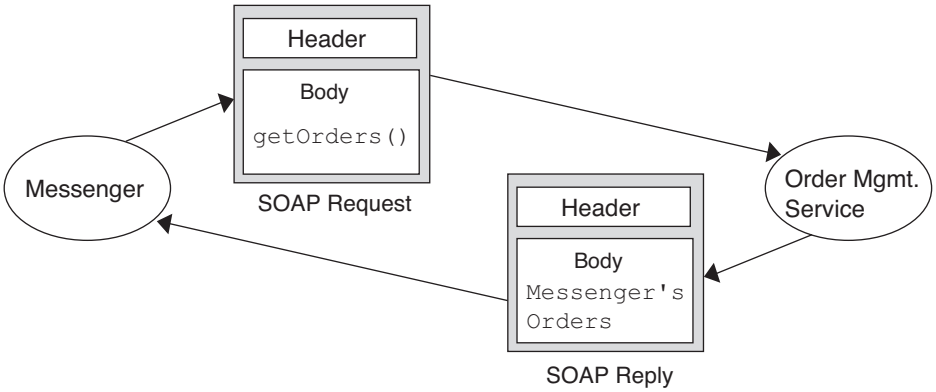


Fig. 2. SOAP message flow.

Figure 3 shows the actual XML content of a SOAP message for the example Web Service. The message is a remote procedure call (RPC) for the operation `getOrders()`:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
       <ns1:getOrders xmlns:ns1="urn:CityCycle">
       </ns1:getOrders>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 3. An example SOAP request.

The SOAP request in figure 3 consists of a SOAP envelope without a SOAP header, and the message body. The SOAP body has a single child element for the request `<getOrders>`. This simple request does not have any parameters.

The order management Web Service returns a list of orders that might look like this:

OrderID	**Messenger**	**Done**	**Distance**	**Description**
bananas	Rees	no	36 km	45 Mediterranean Avenue – 166 Kentucky Avenue
secrets	Bush	no	6 km	120 Atlantic Avenue – 109 Marvin Gardens
oranges	Rees	no	12 km	3 North Carolina Avenue – 1090 Boardwalk

The corresponding SOAP response would look as in figure 4

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <SOAP-ENV:Body>
      <ns1:getOrdersResponse
         xmlns:ns1="urn:CityCycle"
         SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <return xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns2:Array"
            xmlns:ns3="urn:xml-soap-citycycle"
            ns2:arrayType="ns3:order[1]">
            <item xsi:type="ns3:order">
               <orderId xsi:type="xsd:string">bananas</orderId>
               <done xsi:type="xsd:boolean">false</done>
               <km xsi:type="xsd:long">36</km>
               <orderText xsi:type="xsd:string">
                  45 Mediterranean Avenue - 166 Kentucky Avenue
               </orderText>
               <assignedMessenger xsi:type="xsd:string">
                  Rees
               </assignedMessenger>
            </item>
                  ...
         </return>
      </ns1:getOrdersResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 4. A SOAP response to `getOrders()`

The body of the response message – just like the body of the request – has only a single child element, viz. the `<getOrdersResponse>` element. As a convention, SOAP RPCs use the same name for the return value as in the request body, and only add the suffix `Response`. The returned value is an array containing three orders, with only the first of those given in figure 4. The other two orders have been omitted for brevity here.

CORBA and Java/ RMI versus SOAP

The main difference between accesses to Web Services and traditional client/server scenarios such as CORBA or Java/RMI is that the participants in Web Services interactions are generally less tightly coupled. This means that senders and receivers need to make fewer assumptions about each other so that changes in either have only limited effects on the communication partner. Also, Web Services are typically much coarser-grained entities than remote objects, and interactions will in many cases be restricted to simple message exchanges rather than elaborate, application-level protocols involving, e.g., callbacks. In the example above, the target of the SOAP messages is the entire service, not an individual implementation object.

low protocol overhead

Web Services have been designed for light-weight, cross-domain usage from the beginning. Consequently, they avoid those features of other distribution platforms that would impose an undue amount of protocol overhead and complexity. As an example, the dynamic assignment of TCP ports to CORBA servers and the use of a sophisticated remote invocation protocol have made it difficult for CORBA's IIOP protocol messages to traverse packet filter firewalls at domain boundaries. These firewalls do not pose a problem for SOAP messages as these messages travel in HTTP requests and replies, which are typically allowed to pass through firewalls freely.

1.1.1 Web Services Security

packet filter firewalls cannot protect Web Services

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.

As an example, consider the order management Web Service again. This service is not intended to be used by the general public, not even by regular clients of the company. Rather, it was designed only for employees and subcontractors. Allowing unauthorized users to access the service may violate the integrity of the service, e.g., if a list of orders being processed could be manipulated. It may also lead to confidentiality breaches if private data such as the compensations earned by individual messengers can be leaked to observers. New mechanisms are required to reliably prevent security breaches that could otherwise compromise the integrity, confidentiality, or availability of critical company data.

1.1.2 Message-oriented Security and Security Assertions

Point-to-point vs. end-to-end security

In a general Web Services scenario a SOAP message may need to travel through an arbitrary number of intermediate message processors before reaching the actual target. These intermediates may be other Web Services, or infrastructure components like caching HTTP proxies, message routers, security proxies, firewalls, etc. If a connection-based mechanism such as the *Secure Socket Layer/Transport Layer Security* (SSL/TLS) is used to protect messages, a receiver of a message would need to trust all previous

intermediate peers to provide the appropriate security information when setting up SSL connections, and not to inappropriately modify or leak message contents.

In short, transport security mechanisms such as SSL/TLS can only provide *point-to-point* security. While this may be sufficient within a trusted network, it effectively prevents the use of Web Services for business processes that cross enterprise boundaries. What is really needed here are means to achieve *end-to-end* security across domain boundaries.

Web Services Security (WS-Security)

The emerging *Web Services Security* (WS-Security) standard by the OASIS consortium [WS-S] defines a general model for message-oriented security that can be enforced independently of the connection layer and in terms of individual messages. This means that a receiver can autonomously determine the level of trust for a message solely on the basis of the security information in this message, and independently of any intermediates that may or may not have previously processed this message. To achieve this, WS-Security defines generic XML data structures for message headers that can hold security tokens, such as passwords, Kerberos tickets, certificates, and digital signatures. In the WS-Security model, security tokens may be obtained from services that are called *security token services*. To provide message integrity and confidentiality, WS-Security relies on the additional standards *XML Digital Signature* [XMLDSIG] and *XML Encryption* [XMLEnc].

As an example, the sender of a message may want to claim that the message was sent by himself. He would do this by adding a security token that is trusted by the receiver and that asserts the message origin, e.g., as a role name or user identity. The receiver can then verify the validity of the token and check that it asserts a known sender.

Security Assertion Markup Language (SAML)

WS-Security does not define the actual contents and the internal format of security tokens but simply provides standardized SOAP message header elements to hold them. A complementing specification that fills this gap is the *Security Assertion Markup Language* (SAML) [SAML], also defined by OASIS. In SAML, security tokens are signed assertions that express statements by an authority over a subject. In the example, the sender of a message would obtain an assertion and add it to the message, or it could ask

a security token service to create an assertion and modify the message by adding that assertion at the same time. Figure 5 depicts this situation.

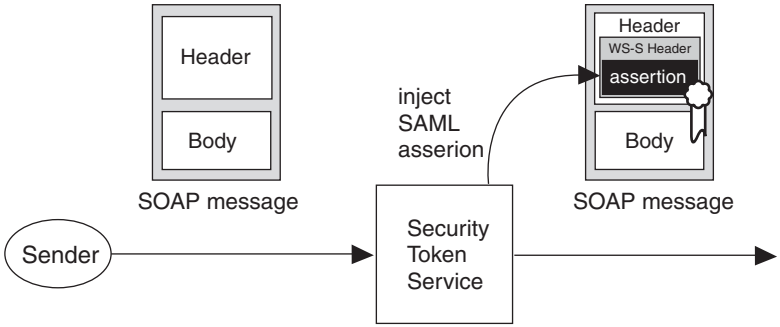


Fig. 5. Security Token Service adds a SAML assertion to a SOAP message

1.2 Securing Web Services with the WS-DBC

the WS-DBC provides full AAAA

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.

no modification of client or service implementation required

The WS-DBC is transparent to both sources and targets of SOAP messages. This means that neither client programs nor service implementations of messages need to be modified when WS-DBCs are interposed in the path of messages exchanged between them. From the perspective of a sender, a WS-DBC behaves exactly like the target service. Likewise, from the perspective of a target service receiving a SOAP message, the WS-DBC behaves exactly like a normal sender.

For message senders, the WS-DBC can play the role of a security token service and transparently obtain and insert security tokens into messages before forwarding these. To do so, the WS-DBC relies on standard WS-Security message headers and SAML assertions. Thus, message senders need not set up additional infrastructure components and integrate these into their applications.

1.2.1 Product Features

authentication

Access to Web Services can be restricted to authenticated clients. The WS-DBC supports the following mechanisms for client *authentication*:

- Anonymous access,
- Authentication by IP-address,
- HTTP basic authentication,
- Public key certificate-based client authentication (SSL),
- Authentication based on SAML assertions.

message protection

Communication between clients and the WS-DBC can be protected using SSL to ensure *integrity* and *confidentiality* in a point-to-point fashion. The end-to-end integrity and confidentiality of message headers and bodies is ensured by applying XML digital signatures and XML encryption.

content inspection

Services are protected from illegal, potentially malicious XML messages through:

- an XML well-formedness check,
- XML schema validation (optional),
- signature verification,
- content filter handling,
- virus scanning of SOAP attachments.

access control

The WS-DBC performs *authorization* for SOAP requests with access policies based on:

- the resource URL and application namespace,
- the operation name in the SOAP request,
- the arguments of an operation.

auditing

The WS-DBC provides *auditing* facilities, which can be configured to send event notifications.

enterprise management

The WS-DBC provides various enterprise management features like:

- policy versioning and rollback,
- concurrent administrator access,
- role-based access control of administrator actions,
- support for XACML as policy format,
- generation of XML schema files from WSDL documents (for schema validation),
- configurable error messages.

WS-DBC configuration and policy *administration* are conveniently managed using the Administration Console.

Exporting Policies in XACML format

The *eXtensible Access Control Markup Language* (XACML) is an XML-based language to describe access control policies. XACML is an OASIS standard and currently in version 1.1. The WS-DBC allows you to export the access control rules of a security policy in XACML format.

Exporting policy data in this way is useful

- for archiving policy data in a standardized, readable format.
- for visualizing policies based on automated translation of XACML to other formats, e.g., to HMTL. This can be done elegantly using tools such as XSLT processors.
- as a policy interchange format that forms the basis for cross-product policy integration.

While XACML is currently only supported by a limited number of security products, it is expected that this will change in the mid-term. Even without widespread, out-of-the box product support, however, XACML can be used as a design tool and a formalism for exchanging policy data between individual policy designers.

1.3 WS-DBC Components

The *Web Services DBC* is an infrastructure building block. It comprises the following components:

- the *WS-DBC Proxy*, which transparently intercepts SOAP messages, creates new SAML assertions and extracts information from existing ones, 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 WS-DBC components and managing security policies.

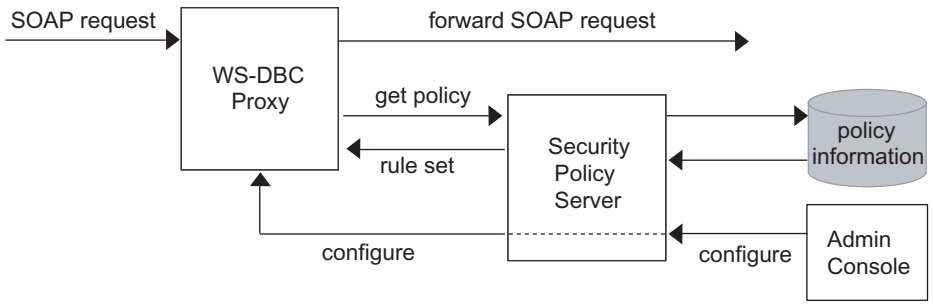


Fig. 6. WS-DBC components

Figure 6 illustrates how a SOAP request is processed by the WS-DBC. The remainder of this section introduces each WS-DBC component and its tasks in more detail.

1.3.1 WS-DBC Proxy

Policy Enforcement Point (PEP) and Policy Decision Point (PDP)

The WS-DBC Proxy typically runs on a dedicated host in a *demilitarized zone* (DMZ) of the network. It combines the functionality of a *Policy Enforcement Point* (PEP) and a *Policy Decision Point* (PDP)[1], as shown in figure 7. A PEP *enforces* decisions, i.e., it allows or denies an action on a resource. The PDP *makes* these access decisions and passes the result of the decision to the PEP for enforcement. The WS-DBC Proxy is thus both a security *mechanism* that enforces security policies (the PEP) and a *decision function* (the PDP), that consults policies to determine how the PEP should act.

1 The terms PEP and PDP are used as defined in the OASIS *Security Assertion Markup Language* (SAML) Specification.

Note that the WS-DBC Proxy must not be bypassable. The target Web Service must be deployed so that it receives SOAP messages *only* through the WS-DBC Proxy. This can be achieved, e.g., by setting up internal routers or packet filters in such a way that only the WS-DBC Proxy may communicate with the service. If the Web Service is reachable without going through the WS-DBC Proxy, the security of the service cannot be ensured.

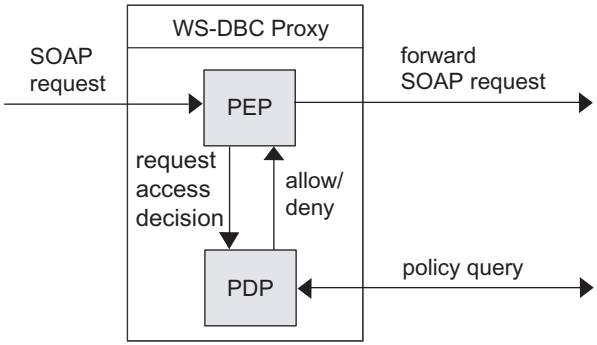


Fig. 7. WS-DBC Proxy functionality

In more detail, the WS-DBC Proxy carries out the following steps:

- obtain the user ID of the SOAP sender,
- authenticate that ID,
- determine the appropriate target SOAP resource,
- if the target SOAP resource is protected, confirm proper style of authentication and check whether sufficient access rights have been granted for this operation on this resource,
- if required, confirm that the SOAP message conforms with its XML schema,
- if required, check message content (according to defined filter rules or XPath expressions),
- if required, create a SAML assertion stating the SOAP sender's user ID, and insert the assertion into a WS-Security header,
- if required, sign the SOAP message,
- forward the message to the target resource.

1.3.2 Security Policy Server

Policy Retrieval Point (PRP)

The *Security Policy Server* (or *SPS* for short) is the component from which the proxy as a PDP requests its policy information. Hence, it functions as a *Policy Retrieval Point* (PRP), i.e., it provides a query interface to an internal policy repository that allows the WS-DBC Proxy to retrieve the policy that must be enforced in a given situation. Figure 8 illustrates the functionality of the Policy Server.

Because security policies are highly sensitive data, they are typically not stored in or retrieved from the DMZ, but rather managed in a more strictly protected internal network. While it is possible to run the Policy Server and the WS-DBC Proxy on the same host, it is not recommended.

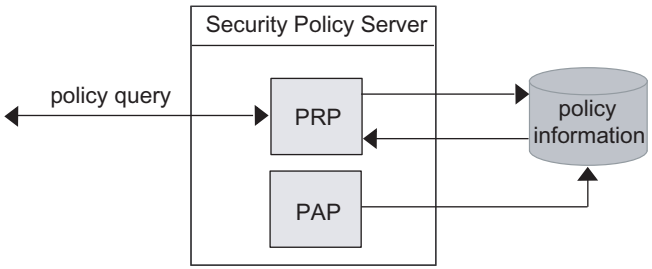


Fig. 8. Security Policy Server functionality

Policy Administration Point (PAP)

The Security Policy Server (SPS) also acts as a *Policy Administration Point* (PAP), which means that it provides an administration interface for the policy repository. This interface is accessed through the *Administration Console*, a graphical tool used by administrators to manage security policies and to configure the WS-DBC installation. The Admin Console communicates with the SPS through a secure SSL channel.

The Security Policy Server manages different kinds of policies:

- The *access control policy* defines which users have access to resources (either for the whole Web Service or, optionally, for individual operations offered by the Web Service).
- The *authentication policy* defines how the SOAP sender must authenticate.
- The *message protection policy* defines how a SOAP message will be cryptographically protected to ensure message integrity
- The *content inspection policy* defines whether and against which schemas SOAP messages will be validated. Additionally, it defines filter rules (including XPath expressions) against which the message content will be checked.
- The *audit policy* defines the events to be logged by the WS-DBC.

Because security policies are sensitive data, they are typically not stored in the Demilitarized Zone (DMZ), but rather managed in a more strictly protected internal network. For this reason, it is not advisable to run the Security Policy Server on the same host as the WS-DBC Proxy in a production environment. However, it is possible to do so in an evaluation setting.

1.4 Scalability and High Availability

This section describes the architectural concepts for the deployment of the WS-DBC in scenarios with high demands on service scalability and availability. For more details on how to configure WS-DBCs for these requirements see Appendix 1, “High Availability and Scalability” on page 11 of the Deployment Guide.

1.4.1 Scalability

The message processing capabilities of a single WS-DBC Proxy are sufficient in most cases. Figure 9 depicts a typical scenario with one WS-DBC Proxy and two clients.

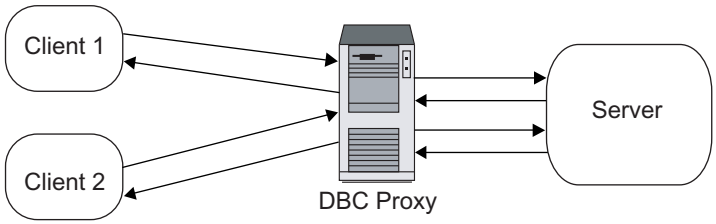


Fig. 9. Single WS-DBC Proxy, high load

For a very large number of clients, or clients with high throughput requirements, the system load of a single WS-DBC Proxy may reach a point where the processing speed is no longer sufficient. Hardware upgrades may remedy the situation, but systems designed

for high performance can be very expensive. An alternative approach is the use of multiple machines in conjunction with a *Traffic Redirector* (see figure 10).

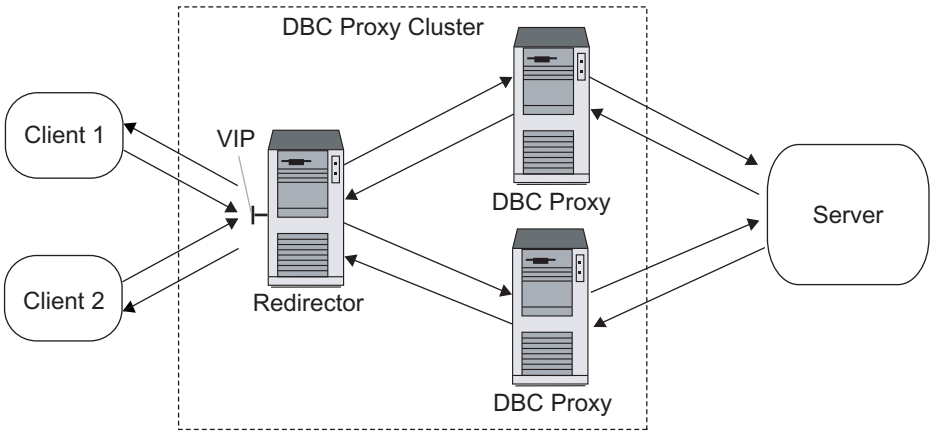


Fig. 10. Multiple WS-DBC Proxies, balanced load

A Traffic Redirector is a software add-on or dedicated device that employs a load-balancing algorithm to distribute client connections to a cluster of individual WS-DBC Proxies. The load of message processing is reduced on the individual WS-DBC Proxy machines in the cluster, allowing the deployment of less expensive hardware.

Traffic redirection is also useful for high bandwidth networks. A single WS-DBC Proxy may not be able to handle the full bandwidth by itself, but a cluster of WS-DBC Proxies can be deployed such that the sum of individual throughput capacity matches that of the network.

1.4.2 High Availability

Another important feature in mission-critical deployments is high availability. In case the WS-DBC becomes unavailable, the Web Services behind it can no longer be reached by clients, so downtime of the WS-DBC is as costly as downtime of the Web Services. There are two approaches to ensuring high availability of the WS-DBC: *hot standby* and *traffic redirection*.

High Availability with Hot Standby

If no load balancing is required, the simplest and most cost-effective approach to high availability is *hot standby*. In this case, no traffic redirector component is required, only

one additional machine with a standby, secondary WS-DBC installation is necessary. In the simplest scenario with only two machines, the WS-DBC Proxy and Policy Server are installed co-located on the same host.

During normal operation the primary WS-DBC host answers requests on a virtual IP address. The standby WS-DBC host does not process requests but only *monitors* the primary WS-DBC host. Once the primary WS-DBC host as a whole or the Proxy component running on this host becomes unavailable, the secondary host takes over the primary host's IP address and starts processing requests until the primary WS-DBC host is operational again. This behavior requires cooperation between the WS-DBC installation and a network-level failover software package in order to move the virtual IP address from the primary to the secondary (or standby) WS-DBC host.

Clients will notice a termination of active connections during the failover procedure but can resume normal application operation after the standby WS-DBC has taken over. For more details, please see Appendix 1, "High Availability and Scalability" on page 11 of the Deployment Guide.

High Availability with Traffic Redirection

The cluster architecture described above, though primarily focused on scalability, can also be used to meet the availability demands.

Most traffic redirector products are capable of *monitoring*, or may be coupled with a third party monitor product. The monitor uses various mechanisms to determine the availability of individual cluster machines, as well as the services running on them. If a machine or service becomes unavailable due to hardware/software/network failures, the monitor will tell the redirector to remove the machine from its distribution list.

As client connections are distributed among the cluster machines, service continues uninterrupted for clients on other machines, and only those clients that were attached to the failing machine are affected at all. For the latter, any existing transport sessions are aborted, but they can immediately be re-established (now on another cluster machine). To clients it appears as if the failed WS-DBC machine performed a fast reboot.

Note that this architecture cannot provide completely uninterrupted services, and thus cannot replace a transaction monitor in critical applications.

Once the WS-DBC machine becomes available again, the monitor will notice this and tell the redirector to add it to its distribution list again. Subsequently the machine may be used for new client sessions.

1.5 Features of the WS-DBC Enterprise Edition

The WS-DBC is available in the standard or the enterprise edition. The following features are only available in the **Enterprise Edition** of the WS-DBC:

- **Linear Scalability, High Availability**: The WS-DBC supports several clustering technologies for load balancing and high availability.
- **Message Filtering**: Administrators can conveniently define expressive message filters to enforce content-based access control and thus thwart application-level attacks, such as SQL injection.
- **SNMP Support**: Audit events can trigger SNMP traps to allow for integration with system management tools.
- **Enterprise Security Policy Server**: A single instance of the Security Policy Server is capable of controlling multiple WS-DBCs as well as I-DBCs.
- **Policy Versioning and Roll-back**: The WS-DBC internally versions policy and configuration data and supports roll-backs to previous versions in case of administrator errors.
- **Support for Multiple, Concurrent Administrators and Role-based Administration Rights**: The WS-DBC is designed for enterprise deployment and fully supports concurrent administrator access, which is controlled by role-based definition of administrator permissions.

1.6 References

[SOAP] SOAP Simple Object Access Protocol (SOAP) Version 1.1, May 2000, http://www.w3.org/TR/SOAP

[WS-S] Web Services Security, http://www.oasis-open.org/committees/wss

[SAML] Assertions and Protocol for the OASIS Security Assertion Markup Language, May 2002, http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf

[XMLDSIG] XML-Signature Syntax and Processing, W3C Recommendation, February 2002, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

[XMLEnc] XML Encryption Syntax and Processing, W3C Candidate Recommendation, August 2002, http://www.w3.org/TR/2002/CR-xmlenc-core-20020802/

CHAPTER

2 Security Functionality and Use Cases

This chapter describes the WS-DBC's core functionality in more detail and presents two common use cases. In the first use case the WS-DBC is deployed next to the target services that it protects. In the second use case, another WS-DBC is employed at the sender side for client-side security and to facilitate business-to-business (B2B) scenarios.

2.1 Introduction

The security functionality supported by the WS-DBC falls into the following categories:

- message authentication,
- content inspection,
- access control,
- message protection, and
- auditing.

This chapter examines these security functions in the context of the two main use cases of the Web Services DBC and roughly follows the logical path of a SOAP message travelling from a sender through the WS-DBC to the receiver. Because auditing is performed along the whole path, it is summarized in advance in the following subsection.

Auditing

Any critical system must be monitored during operation. Monitoring is essential in order to detect when something starts going wrong so that countermeasures may be taken. If a system malfunction cannot be prevented, or if it is detected later, it is important to have records with traces of events that occurred in a system. Analyzing these traces helps finding out exactly what went wrong afterwards, and why it could happen. With respect to security, analyzing records of security breaches can show how a given attack happened, and why it was not prevented by security mechanisms and the policies enforced by them. Thus, new security mechanisms may be found necessary, or incorrect or incomplete policies may be identified and fixed.

These written traces of events are called *audit logs*. An audit log is a record of security-relevant events that the system observed and that can be analyzed to determine the effectiveness of security services as well as the reasons and circumstances of system failures. For more details about auditing and the specific audit policies that can be specified in the WS-DBC, see chapter “Audit Policy” on page 189. The list of auditable events that the WS-DBC can generate can be found in appendix “Audit Events” on page 321.

2.2 Use Cases

target-side WS-DBC

In the typical case, a WS-DBC is set up in a domain to protect Web Services that are hosted in that domain. Such a WS-DBC is conceptually part of the target services that it protects and hence called a ***target-side*** WS-DBC.

sender-side WS-DBC

Both the development and the management of client software can be greatly simplified by using an additional ***sender-side*** WS-DBC. A sender-side WS-DBC can be deployed to transparently insert SAML security assertions into outbound messages from clients that are themselves SAML unaware. It thus supports centralized deployment and management of security information in the client domain, which simplifies the establishment of business-to-business (B2B) relationships. Moreover, it provides audit, content inspection, and authorization functionality on the client side. Finally, a sender-side WS-DBC can be used to integrate third-party security services based on WS-Security and SAML as these become available.

The differences between a setting with a single target-side WS-DBC and a setting with an additional sender-side WS-DBC are presented in more detail in the following sections.

Note that the use cases assume trusted network transport between WS-DBCs, i.e., either a trusted intranet situation or transport-level security using SSL/TLS.

2.3 Use Case 1 – Single target-side WS-DBC

In this basic use case, a web service provider operates a target-side WS-DBC to protect his or her services.

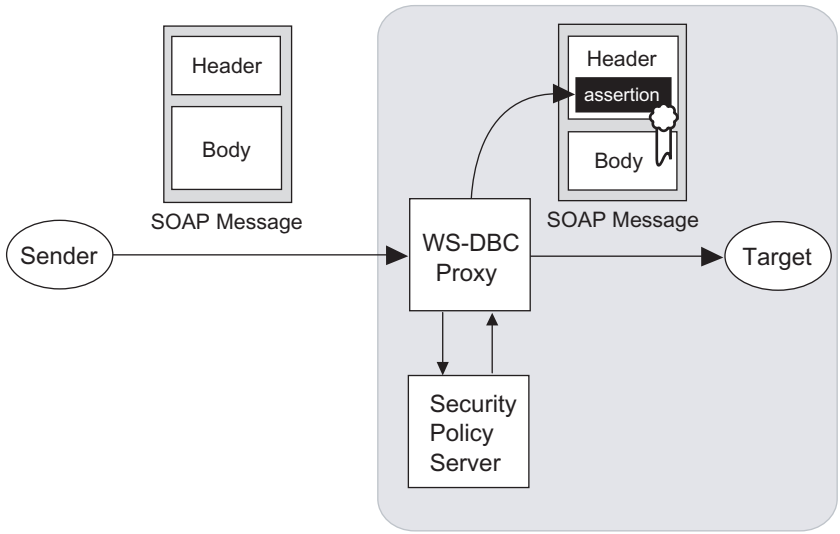


Fig. 11. A single, target-side WS-DBC

2.3.1 Message Authentication

identification and verification of a message's origin

Before the WS-DBC can make policy decisions such as an authorization decision, it needs to retrieve the security information upon which a decision can be based. Any such information must either come from a completely trusted source or first be verified or *authenticated*. For message-based security, it is necessary to authenticate the security-relevant information about messages. The most important information that must be identified and verified here is the origin of a given message.

authentication via SAML assertions

In the message-oriented approach of Web Services security the receivers of SOAP messages rely on security information in the message envelope itself. The standardized way to denote the origin of a message is to attach a small XML document to that message which contains an assertion. The OASIS *Security Assertion Markup Language* (SAML) specification defines a standardized XML format for attributes, and this is the format in which the WS-DBC expects to receive information about the origin of a message. As mentioned in chapter 1, SAML attributes travel inside special headers as defined by Web Services Security (WS-Security).

Not every Web Service client is SAML-aware and capable of providing SAML assertions within messages headers. Frequently, as in this use case, a SOAP message arriving at the WS-DBC will not contain any SAML assertion. To authenticate such a message, the WS-DBC must rely on credentials provided by one of the following, alternative authentication mechanisms:

- authentication via the client's source IP-address,
- HTTP basic authentication: user ID and passwords transmitted in HTTP headers,
- SSL-based public key authentication: X.509 public key certificates transmitted in the SSL layer of HTTPS communications,

If a WS-DBC authenticates a message sender using its IP-address, HTTP basic authentication, or SSL and the target resource is configured for SAML injection, it will create a SAML assertion in a later step that asserts the message origin in the form of a user identity attribute. In other words, the WS-DBC provides an entry point for SOAP messages to the Web Services Security model. As described in the next use case in section 2.4, a WS-DBC on the sender side can be used purely for this purpose.

If no authentication information is provided, the message is treated as anonymous. In this case, no SAML assertion is created. The WS-DBC can be configured to accept or reject anonymous messages. In the present use case, no SAML assertions are found, so the sender has to use any of the other available mechanisms, or else the message is regarded as anonymous. If authentication is attempted but fails, the WS-DBC Proxy will generate appropriate audit events.

2.3.2 Content Inspection and Access Control

After the WS-DBC has authenticated a SOAP message, it analyzes the message content and checks authorizations based on the authenticated message origin.

Content Inspection

XML well-formedness and optional schema validation

Message analysis is based on XML processing of the message. This processing consists of a basic, obligatory check for XML well-formedness, and an optional XML schema validation step. Both steps protect the receiver of the SOAP message from XML content which could either inadvertently or maliciously damage the receiver. Identifying and filtering SOAP messages that cannot be processed by the receiver can also reduce the load on the target. Because the XML schema validation step can be computationally expensive, it may be skipped in contexts that do not require this level of protection. More details on XML processing in the WS-DBC can be found in chapter 3 on page 49.

When a SOAP request is rejected at the XML processing stage, the WS-DBC will send an audit event that describes the reason for the failure. It will also send a SOAP fault to the SOAP client. A list of error messages generated by the WS-DBC can be found in appendix B, “Error Messages and System Exceptions” on page 339.

parameter checking

The WS-DBC can perform content inspection by validating the parameters of incoming messages. Flexible filter rules (including XPath expressions) for operation parameters can be defined with the Admin Console. Messages carrying arguments that do not conform to the filter rules will be rejected.

virus scanning of SOAP attachments

SOAP attachments can contain binary content that may be virus-ridden. The WS-DBC supports scanning of these SOAP attachments by allowing administrators to

- generally allow or disallow messages with attachments,
- configure external filter programs for individual MIME types that will be called whenever content of that type is encountered in attachments.

Access Control

retrieving policy information

After the authentication and analysis stages, the WS-DBC knows the message origin. The WS-DBC also knows the name of the single XML element in the SOAP body which denotes the operation that is to be invoked in the case of SOAP-RPC, and the target service that it protects. Based on this information, the WS-DBC Proxy retrieves applicable authorizations from the Security Policy Server – its policy retrieval point (PRP) – to make an access decision.

fine-grained access control policies

The Security Policy Server is also a policy administration point (PAP) and thus provides administrative interfaces to specify access control policies, which contain the authorizations for messages. The underlying access control model provides additional concepts for flexible and scalable access policies, viz. *groups* and *roles*. Please see part 3 “Managing Security Policies and Deploying Web Services” on page 233 for details on the specification of access policies.

An access decision is either a “permit” or a “reject”. If the message is to be rejected, the WS-DBC Proxy generates an audit event and a SOAP fault, and discards the message. If it is permitted, the WS-DBC forwards the message but first may need to insert SAML assertions and to apply message protection to it.

SAML Injection

Before a message is forwarded to the target, the WS-DBC may transform the message by adding header elements and applying signatures. Depending on the message forwarding policy for a particular resource, the WS-DBC Proxy creates an assertion containing the sender’s identity (authenticated user ID). If the message already contains a SAML

assertion, this assertion is passed on. Subsequent message processors (other gateways or security at the target service) can enforce their own policies based on this information.

message forwarding policy

If it is known that there are no SAML-aware message processors downstream from the WS-DBC Proxy, it may not be necessary to insert the assertion. Skipping this step reduces message processing time in the WS-DBC Proxy and application server.

Message Protection

Messages that are to be forwarded may require protection of their confidentiality and integrity to prevent eavesdropping, forgery, or sabotage on the communication path between the WS-DBC and the target.

Message *integrity* protection has two aspects: message body integrity and message header integrity. Both kinds of protection are achieved by applying digital signatures that comply to the XML Digital Signature standard specified by W3C.

digital signatures

By digitally signing a digest of the SOAP body and the SAML assertion, the asserting WS-DBC allows receivers to decide whether or not they want to accept the signer as an attribute authority. In other words, receivers can check the signature and determine how much they trust the signer. The signature also ensures that the SAML assertion cannot be copied and reattached to another message by potential attackers. Moreover, any modifications to either the message body or the assertion can be detected by the receiver.

XML encryption

With XML encryption, the WS-DBC supports end-to-end confidentiality protection that can be persisted together with messages.

SSL protection

To prevent eavesdropping, a sender and a target WS-DBC can be configured to use SSL on the transport layer between sender and sender-side proxy (as in the use case described in section 2.4, page 47), between sender and target proxy, and between target-side proxy and target service. Note that each point-to-point connection in the call chain needs to be considered to prevent eavesdropping, not just the connection to, and within, the WS-DBC operating environment.

Summary

In summary, the tasks performed by the WS-DBC as presented in this use case with a single target-side WS-DBC, are the following:

- identification of message origin (authentication) using non-SAML information,
- XML validation of the SOAP message (XML well-formedness check and XML schema validation),
- authorization of the RPC in the SOAP message (including parameter checks and XPath filtering),

- (optionally) creating a SAML assertion,
- (optionally) digitally signing the SOAP message to bind the SAML assertion to the SOAP body,
- logging of auditable events,
- message forwarding.

2.4 Use Case 2 – Sender-side and target-side WS-DBCs

sender-side WS-DBC as attribute authority

In this second, more advanced use case an additional WS-DBC at the sender-side forwards messages to a target WS-DBC. The main purpose of a sender-side WS-DBC is to add SAML assertions to outgoing messages that were sent by security-unaware sender software. This scenario is depicted in figure 12.

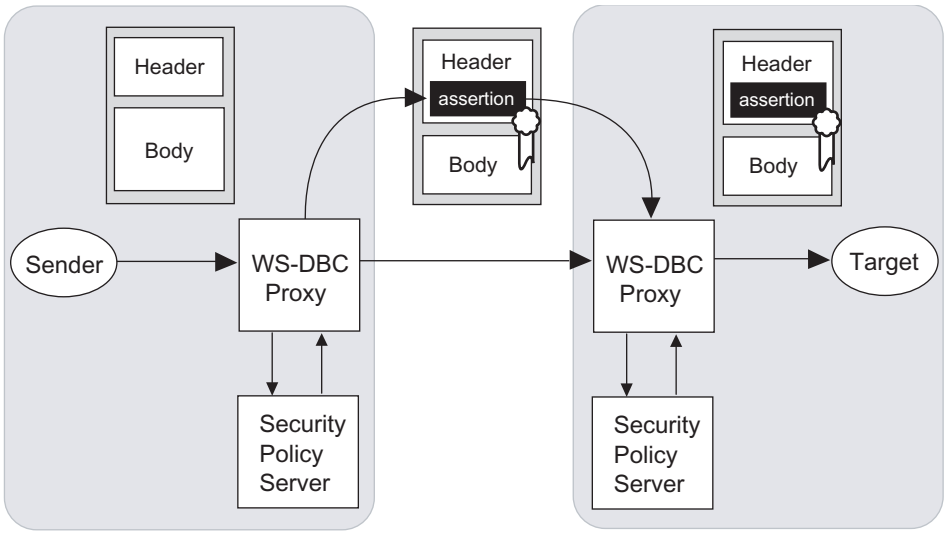


Fig. 12. A sender WS-DBC as an attribute authority.

steps carried out by the sender-side WS-DBC

The sender-side WS-DBC plays the role of a security token service in the WS-Security model, or SAML *authority* that can verify and vouch for the sender. As described in use case 1 the sender-side WS-DBC must rely on the communication context to authenticate the client. If the authentication process was successful, a SAML assertion containing the subject identity is created, and added to the message. The sender-side WS-DBC can also perform auditing and outgoing access control.

A main advantage of this scenario is that the sender-side WS-DBC maintains control of the user authentication (authentication data and authentication mechanism). For example, such a setting may permit SOAP senders to authenticate with the sender WS-DBC using HTTP basic authentication even without underlying SSL to protect passwords if the sender domain is sufficiently trusted. The sender-side WS-DBC will then create SAML assertions as a means of authenticating with the target WS-DBC. Optionally, the message is digitally signed.

SAML assertions inserted by the sender-side WS-DBC are used by the target WS-DBC to authenticate the sender. The target WS-DBC validates the injected assertion and passes it on after performing its usual authorization and auditing functions. Its additional tasks are:

- verify the sender's XML digital signatures over the SAML assertion and the message body, and determine whether the signer is trusted,
- extract the user identity from the SAML assertion,
- forward the unmodified SOAP message (no additional signatures or assertions are created).

CHAPTER

3 Advanced XML Processing

This chapter discusses advanced aspects of XML Processing in the WS-DBC, viz. client bootstrapping using WSDL, and XML Schema Validation of SOAP messages. A general understanding of Web Services and XML schemas is required for this chapter.

Section 3.1 explains how the WS-DBC deals with SOAP toolkits that rely on WSDL for client bootstrapping, while section 3.2 introduces XML schema validation. Section 3.3 explains how schemas are loaded by the WS-DBC Proxy. It also discusses the limitations of schema validation. Section 3.4 describes how to find out which schemas are necessary, and lists the schemas included in the WS-DBC installation.

3.1 Web Services Description Language (WSDL)

WSDL is an XML-based language for writing interface descriptions for Web Services. WSDL documents contain descriptions of operation names and parameter types, and additionally location information that allows clients to find a Web Service (i.e., a URL).

Some SOAP toolkits use a special mechanism based on the *Web Services Description Language* (WSDL) for the initial bootstrapping of clients. Even though this mechanism is not standardized, the WS-DBC supports this way of locating services. This section describes how WSDL is supported by the WS-DBC.

The WSDL description for a service is created by a Web Service developer to give client programmers the interface definition that client applications must conform to. There is a variety of usage scenarios for WSDL employed by the different Web Services frame-

works that are beyond the scope of this manual and will not be discussed. For further information on WSDL, please refer to [WSDL] [UDDI]. We will present some common scenarios and explain the issues involved with WSDL and deploying a WS-DBC:

- **Out-of-band transmission of WSDL**: A WSDL description can be turned into a programming language specific stub (e.g., a Java stub) automatically. If the client's Web Service SDK contains such a tool the service can be interfaced easily.
- **Publishing WSDL on UDDI servers**: A WSDL document can also be published on a server providing *Universal Description, Discovery and Integration* (UDDI) services [UDDI]. Such a server is not necessarily located in the same domain as the Web Service itself.
- **Runtime transmission of WSDL**: Other Web Services frameworks use the WSDL for bootstrapping purposes and download WSDL descriptions dynamically. Stubs are created at run-time, before interacting with specific Web Services. The exact steps carried out by clients depend on the specific toolkit.

3.1.1 Using the WS-DBC in combination with WSDL

Any URLs pointing to protected services[1] need to be modified to point to the WS-DBC Proxy rather than the original target service. Otherwise, clients would attempt to contact the service directly using that URL, which must fail:

- When transmitting WSDL out-of-band or when publishing WSDL files using UDDI, the service provider or publisher must take care that the URL in the WSDL points to the WS-DBC that protects the Web Service instead of the real service host.
- When WSDL documents are dynamically downloaded by the client, a similar modification must be performed. First, the URL that clients use to download service descriptions must point to the WS-DBC because, again, the real Web Service host is not directly reachable by the client. The WS-DBC forwards the request to retrieve the WSDL document to the target, but it modifies the URL in the returned WSDL file in transit to point back to the WS-DBC. Thus, the complete bootstrap is totally transparent to the client application.

proxification of WSDL documents

As just described, the WS-DBC modifies the port description in WSDL documents in transit, depending on the resource mappings configured with the Admin Console. It will perform this manipulation on all WSDL documents that contain URLs in their `<port>`

1 Such a target URL is called "port" in WSDL.

element that match a resource entry in the WS-DBC's resource mappings. Figure 13 illustrates the modifications that the WS-DBC Proxy carries out on a WSDL document.

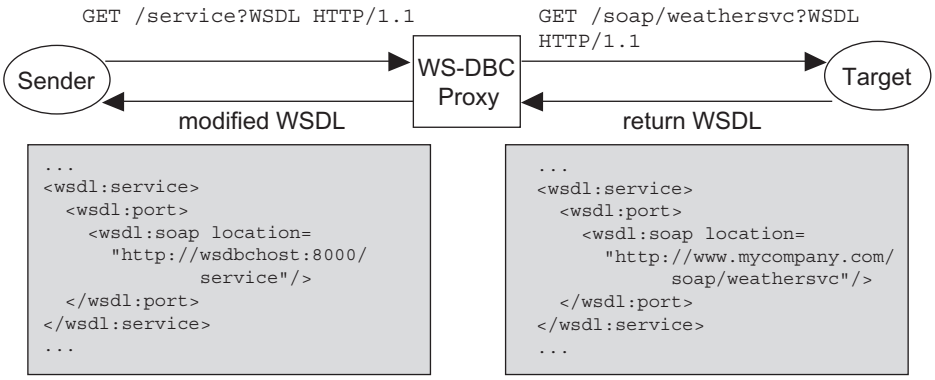


Fig. 13. WSDL modified by the WS-DBC Proxy

In this example we assume a resource mapping from the path `/service` to the resource `http://www.mycompany.com/soap/weathersvc`. Note that the WS-DBC Proxy replaces the host address, port number, and URL path in the WSDL document according to the configuration of the resource mapping (please see section 5.9 on page 150 on how to configure these mappings with the Admin Console). The client transparently continues to connect to the WS-DBC instead of the real service host.

3.1.2 WSDL and HTTP GET requests

As explained above, WSDL definitions are retrieved using plain HTTP GET requests rather than through SOAP messages, which are send in HTTP POST requests. By default, the WS-DBC does not allow HTTP GET requests in order to restrict the exposure of resources to SOAP-only messages. However, this restriction has to be relaxed when WSDL files are to be retrieved. The URL that points to the WSDL document of a certain Web Service can be configured in the WS-DBC, so that a WSDL GET request to this URL will be allowed.

3.2 XML Schema Validation

XML Processing

Whenever the WS-DBC Proxy receives an HTTP request containing an XML document in the body, it tries to parse this content. For SOAP requests, this is necessary in order to

determine the requested operation, so fine-grained access control can be performed. For best performance, XML in HTTP replies are only optionally parsed by the WS-DBC Proxy.

ensuring XML well-formedness

In all cases, the well-formedness of SOAP requests traversing the WS-DBC Proxy is ensured. However, syntactically well-formed messages may still carry a content that can be used for application-layer attacks. For example, if it is known that a particular part of the message body, say a string value, is used by the target service to construct a database query in SQL, then an attacker could attempt to carry out attacks known as "SQL injection attacks". In this case, the attacker would know that a given parameter will eventually be interpreted by an SQL processor. The attack consists of providing a string like `"Paul; DROP TABLE"` that, when interpreted by the database software, executes a security-critical operation, e.g., dropping a table in the database.

Application-layer attacks like these cannot be prevented on the network level. Protection is only possible using application-level information that describes which message values it is safe to accept – and thus which values are to be rejected. An important step towards stricter application-level restrictions on message contents is applying XML schema validation to SOAP messages.

schema validation

As an additional layer of protection for the target of the message, the WS-DBC Proxy can validate SOAP messages against XML schemas. XML schemas contain definitions of types and elements for a given target namespace and support enforcing stricter restrictions on message contents. In essence, XML schemas can be used to provide powerful type-checking including, e.g., matching string values against given patterns, limiting the range of integer values, or requiring particular record structures. Validating a SOAP message against one or more schemas ensures that the validated parts of the XML document comply with the structure that was specified by these schemas, i.e., that these parts are as expected by the receiver.

An important prerequisite is the existence of applicable XML schemas for SOAP-based applications. These schemas must be supplied by the application itself and cannot generally be deduced by security administrators. The stricter the restrictions imposed by schemas, the better the protection against application-level, content-based attacks like the one outlined above.

For example, an XML schema can define derived string data types that must not contain the "`;`" character followed by other characters, which would prevent SQL injection attacks like the one sketched above. By checking string values against the derived string type, the WS-DBC can prevent that such a string will eventually reach an SQL processor which expects to insert the string directly into a database query.

Schema validation can be turned on per resource using the Admin Console. Please refer to section "XML Schema Validation for incoming SOAP message requests and

responses can be enabled by activating the checkbox (an XML wellformedness check is always carried out).” on page 271 on how to enable schema validation for a certain resource.

3.3 Configuration of XML Schema Validation

To schema-validate an XML document, one or more schemas must be made available to the validating XML parser in the WS-DBC Proxy. In theory, there are two options:

dynamic downloading vs. static pre-loading

- *Dynamic downloading* of schemas:
  An XML instance document may have an attribute `schemaLocation` which contains mappings between XML application namespaces and the locations of corresponding schema documents. This, however, poses a risk when the parser blindly starts to download a schema from an untrusted location. In the simplest case, an attacker may just define his own schema so that the message is found to be valid even though it contains different content than what would actually be expected. Therefore, the WS-DBC Proxy disregards any `schemaLocation` attributes in SOAP messages and does not download any new schemas on demand so that offending messages can be reliably flagged as invalid.
- *Static pre-loading* of schemas from a defined file system location on the WS-DBC Proxy host:
  In this case, all available schemas will be loaded once at startup of the WS-DBC Proxy from a configurable location – ignoring any `schemaLocation` attribute that maybe present in the message’s XML text. The WS-DBC Proxy supports only this method, mixing local and remote schemas is not supported.

The default location for schema descriptions (files with the extension `.xsd`) in the WS-DBC Proxy is `<INSTALLDIR>/wsdbc/adm/schemas`. This path is freely configurable per Web Services resource via the Admin Console (see also “XML Schema Validation for incoming SOAP message requests and responses can be enabled by activating the checkbox (an XML wellformedness check is always carried out).” on page 271).

Note that if the loading of a schema file fails, the WS-DBC Proxy will print an error message and continue to load the next schema, i.e., the offending schema will not be used for validation. Please check the event log to make sure all schemas were successfully loaded if you experience problems with XML schema validation. Also note that schema files that are semantically incorrect may lead to the rejection of messages when schema validation is used! Schemas should always be checked for syntax and semantic correctness with the `schematest` utility (cf. Chapter 5 “WS-DBC Tools” on page 49 of the Deployment Guide).

Schemas importing other Schemas or DTDs

Another important point to be considered is that while pre-loading schemas, these are allowed to import other schemas and DTDs from any location, including locations in the internet. Again, schemas from remote locations will not be downloaded for security. Therefore, schemas should be carefully checked for `<import>` statements and required schemas installed on the WS-DBC Proxy hosts before the WS-DBC is configured to pre-load them.

3.3.1 Limitations of Schema Validation

There are two limitations of schema validation: namespaces for which no schema is present, and elements without explicit namespace qualifications. The first case, elements in a namespace for which no schema document is present, will *not* be treated as a validation failure and thus go unnoticed. In the second case, whenever an element is not namespace-qualified, it cannot be validated unless it conforms to the parent element's schema. If the parent element does not place any requirements on its children, then unqualified elements will be treated as valid.

The following example demonstrates this behavior. The given schema defines one element `MyElement` in namespace `http://www.my.com/` that can have arbitrary children which may or may not be namespace-qualified.

Example Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.my.com/"
            xmlns="http://www.my.com/WS-DBC"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            version="0.2">
    <xsd:element name="MyElement">
        <xsd:complexType>
           <xsd:sequence>
              <xsd:any processContents="lax"
                       minOccurs="0"
                       maxOccurs="unbounded"
                       namespace="##any"/>
           </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Example: Valid Document 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<myns:MyElement xmlns:myns="http://www.my.com/"/>
```

This document is an example for the use of `MyElement` without any child elements. This is possible because child elements are defined as optional (`minOccurs=0`) in the schema.

Example: Valid Document 2:

```
<?xml version="1.0" encoding="UTF-8"?>
<myns:MyElement xmlns:myns="http://www.my.com/">
<UnqualifiedOtherElement>
...
</UnqualifiedOtherElement>
</myns:MyElement>
```

In this document, `MyElement` has a child element that is itself not qualified by a namespace - and therefore has no applicable schema document.

Example: Valid Document 3:

```
<?xml version="1.0" encoding="UTF-8"?>
<myns:MyElement xmlns:myns="http://www.my.com/">
<yourns:QualifiedOtherElement xmlns:yourns="http://
www.you.com/">
...
</yourns:QualifiedOtherElement>
</myns:MyElement>
```

In this final example, `MyElement` has a child element that is qualified by a different namespace, but no schema is deployed for this namespace.

While this might seem dangerous at first glance, the overall risk is limited when a few basic rules are followed. For example, when an attacker inserts his own qualified or unqualified element into the SOAP message, this may go completely unnoticed by the target service because it will usually not try to access elements it doesn't know. So when all schemas for elements the target service understands are present, unknown elements should not pose a threat.

If this behavior is considered too risky in any given circumstances, then usage of the `processContent="lax"` directive in the schema's `<any>` elements should be avoided and replaced by the default value `strict`. Also, the `namespace` attribute in these elements should define a real namespace and not contain the generic values `##any` or `##other`.

For more detailed information please consult common XML Schema literature, or the XML Schema language specification [XSD0, XSD2].

3.4 Finding out which schemas are necessary

For the WS-DBC to be able to validate a message against one or more schemas, the WS-DBC must know about these schemas. A number of general schemas are defined by standards bodies such as W3C and OASIS. The standard schemas that the WS-DBC considers as predefined are included in the installation and listed in section "Schemas Included in the WS-DBC Installation" on page 273.

The definition of application-specific schemas, however, is the responsibility of Web Service developers or administrators. WS-DBC administrators that want to enforce XML schema validation should contact these developers or administrators if application-specific schemas were not provided on deployment of the service in the WS-DBC.

Since these schemas may reference other schemas, it is important to carefully inspect and retrieve all required schemas. Missing schema definitions may result in rejection of SOAP messages by the WS-DBC if schema validation is turned on for a resource (Internal Server Error).

There is currently no direct support for automatic analysis of messages with regard to the namespaces that occur in the message. Therefore, different sources of information should be considered:

- The documentation of the target Web Service platform.
- The documentation of the SOAP toolkit at both sides, target and sender.
- The documentation of the Web Service, such as an WSDL description
- A network traffic inspection tool (“sniffer”) that allows you to view the messages as sent “over the wire”.

3.4.1 Writing a Schema file

XML schema files will not always exist for a given application, so it may become necessary to write one. The following steps should be carried out:

1. Create an empty file `service.xsd` for a service which begins:
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="myns"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="myns" elementFormDefault="unqualified"
 attributeFormDefault="unqualified">
```
2. For each operation `foo` provided by the target service, create an element definition:
```
<xsd:element name="foo">
  <xsd:complexType>
  ... <!-- define any foo params in a sequence here-->
  </xsd:complexType>
</xsd:element>
```
3. Make sure the schema is syntactically well-formed by running the `schematest` command (please refer to Chapter 5 “WS-DBC Tools” on page 49 of the Deployment Guide for a detailed description):
```
c:> <INSTALLDIR>/tools/bin/schematest -I service.xsd
Successfully parsed schema file myschema.xsd for target
namespace myns
```

Writing a correct schema is not always easy, and some of the decisions you have to make in the schema file, such as whether you want element and attribute names to be qualified or not, depend on the SOAP toolkits that clients will use.

3.4.2 References

[WSDL] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl

[UDDI] Universal Description, Discovery and Integration of Business for the web, http://www.uddi.org/

[SAML] Assertions and Protocol for the OASIS Security Assertion Markup Language, May 2002, http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf

[XMLDSIG] XML-Signature Syntax and Processing, W3C Recommendation, February 2002, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

[XSD0] XML Schema Part 0: Primer, W3C Recommendation, 2 May 2001, http://www.w3c.org/XML/Schema

[XSD2] XML Schema Part 2: Datatypes, W3C Recommendation, 2 May 2001, http://www.w3c.org/XML/Schema.

PART

2 *INSTALLATION AND CONFIGURATION*

This part is aimed at system administrators who wish to set up the Web Services DBC software. We assume that you are already familiar with the architectural concepts described in the first part of this handbook.

The first chapters of this part list hardware and software prerequisites for installing the WS-DBC, present some common deployment scenarios, and describe how to install the Web Services DBC. Chapters 4 to 9 describe in detail how to configure the WS-DBC using the Administration Console. Chapter 10 presents how key management works in the WS-DBC, and chapter 11 is a troubleshooting guide.

CHAPTER

1 Preparing to Install

This chapter lists hardware and software prerequisites for installing the DBC and presents some common deployment scenarios which help to plan the installation of the DBC software.

1.1 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.

1.1.1 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. Chapter 6 “Hardened System” on page 55 of the Deployment Guide on the CD-ROM provides suggestions on how to set up a hardened Linux configuration.

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 128 MB RAM and 128 MB free disk space (recommended are 256 MB).

1.2 Typical Deployment Scenarios

This section describes some typical deployment scenarios and should help to identify where to install the different DBC components according to your requirements. When considering deployment scenarios, it is important to understand the internal communication links between the DBC Proxy, Security Policy Server, and Administration Console. These links are explained in the following section.

1.2.1 Internal Communication Links and Trust Relationships

control connection

The DBC Proxy communicates with the Security Policy Server in order to request policy information. This communication link, which is called *Control Connection*, has higher security requirements than external application connections to the DBC Proxy. This is because the security enforced by the DBC Proxy depends on the integrity of the security information that is communicated over this link, as does the audit log that is written by the Security Policy Server.

administration connection

The Administration Console communicates with the Security Policy Server to display and edit the configuration and policy data managed by the Security Policy Server. The communication link is called *Administration Connection*. Both kinds of communication links are illustrated in figure 1, “Internal DBC communication links”. Note that this illustration depicts *logical* components, i.e., all three components could be installed on a single host rather than physically distributed.

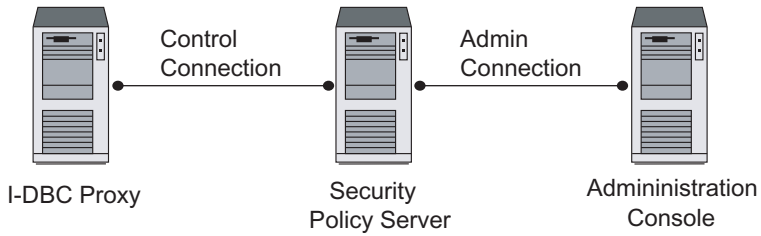


Fig. 1. Internal DBC communication links

The control connection is a pair of TCP connections, one in each direction, addressing a known host and port. The admin connection is a single TCP connection from the Admin Console to the Security Policy Server. The protocol spoken on these connections is IIOP.

security requirements

Since all communications within the DBC are security sensitive, each component must authenticate when communicating with another component so that trust between components can be established. Otherwise a rogue component masquerading as the Security Policy Server could, for example, provide the DBC Proxy with spoofed policy information, which would enable severe security violations. Likewise, access to the Security Policy Server through the Administration Console must obviously be restricted to trusted clients to prevent illegitimate modifications of the policy data. Whether confidentiality of policy and configuration data is an issue depends on the environment in which the DBC is deployed.

trust establishment

To prevent security violations, components generally establish trust by setting up their connections using SSL. The keys and certificates that are required for trust establishment are explained in more detail in chapter 10, “Installing Keys and Certificates” on page 201. SSL is also the standard mechanism used to ensure integrity and, if required, confidentiality of data.

Considering integrity requirements, the following configurations are possible for the control connection:

4. The DBC Proxy is connected via a dedicated management and control subnet with the Security Policy Server. This approach provides the highest degree of security in terms of integrity and availability.
5. The DBC Proxy uses an SSL-protected control connection on the internal part of the perimeter network. Any interior packet filtering router must be configured to allow the two TCP connections.
6. The DBC Proxy and Security Policy Server are collocated, in which case no specific security precautions have to be taken to secure the communication between the two components. Authentication is still required, of course.

Similarly, the Administration Console and the Security Policy Server could be collocated, in which case no integrity or confidentiality protection would be necessary for the administration connection. If the two components are deployed on separate hosts, the integrity of the administration connection would have be protected to avoid, e.g., man-in-the middle attacks that could compromise the integrity of policy data.

1.2.2 Scenario 1: Screened Host Firewall with WS-DBC Proxy

In the first scenario (depicted in Figure 2, “The screened host firewall with WS-DBC Proxy”), Web Services running on a host located in a protected network are to be made accessible from the Internet. In this scenario the only machine directly exposed to access from the Internet is the WS-DBC Proxy.

The firewall in this architecture needs to allow access to at least the HTTP(S) port of the WS-DBC Proxy host from selected internet clients. The WS-DBC Proxy forwards SOAP requests to the respective SOAP receiver. The SOAP receiver can be a standalone application, or a container environment such as a Web Server or J2EE Application Server.

Note that the SOAP receiver itself or the container environment that houses it must not be accessible directly from the internet. If it is possible that SOAP requests bypass the WS-DBC Proxy, no security can be enforced!

Network Address Translation (NAT) which is frequently employed in a scenario like this will be taken into account by the WS-DBC Proxy. NAT has the advantage that internal hosts are not reachable other than through a NAT device because internal addresses are not routable outside the protected network.

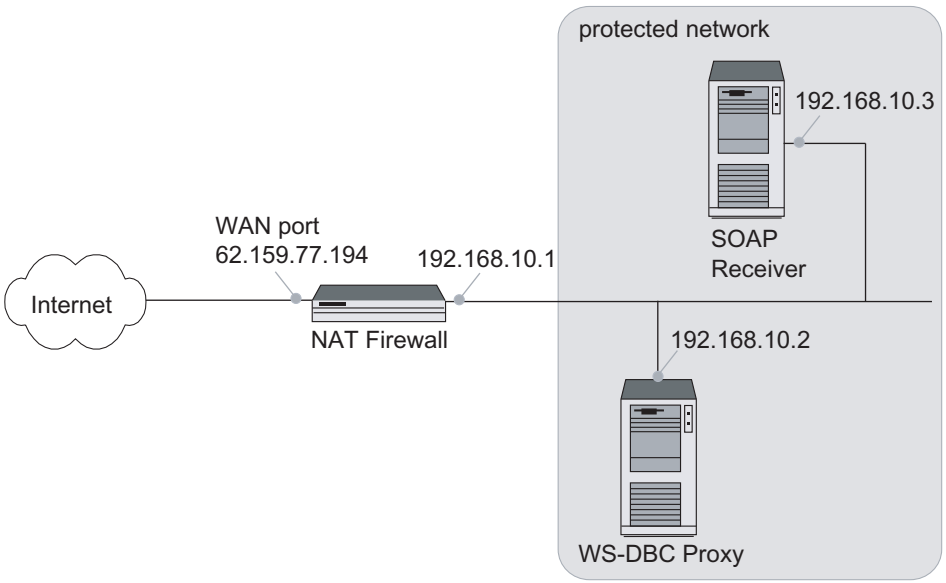


Fig. 2. The screened host firewall with WS-DBC Proxy

The Security Policy Server runs on a host located in the protected network. The WS-DBC Proxy contacts the Security Policy Server to decide whether a contacting client shall be allowed to access the Web Service (cf. “Internal Communication Links and Trust Relationships” on page 63). The Admin Console also runs on a host located in the protected network. In this scenario the Security Policy Server and the Admin Console could be run on the same host as the WS-DBC Proxy. Such a configuration is not recommended, however, because the most sensitive component, the Security Policy Server, would be hosted by a machine that is directly accessible from the Internet.

Firewall Configuration

A recommended firewall configuration for this scenario is:

- Allow traffic from selected internet sites to specific ports of the WS-DBC Proxy host (client initiated HTTP(S) connections).
- Allow traffic from the WS-DBC Proxy to internet systems.
- Reject all other traffic.

1.2.3 Scenario 2: WS-DBC Proxy in the DMZ

The second scenario presents a screened subnet firewall architecture. As shown in figure 3, “WS-DBC Proxy in the DMZ” two firewalls are used to create an outer, screened subnet or *demilitarized zone* (DMZ). This subnet contains the WS-DBC Proxy, which provides the gateway to SOAP receivers located inside the protected domain. The SOAP receiver can be a standalone application, or a container environment such as a Web Server or a J2EE Application Server.

The exterior firewall is the connection point to the internet. It restricts internet access to specific systems in the screened subnet and allows only these systems to access the internet. It blocks all other traffic from/to the internet. The interior firewall restricts access from the protected network to specific systems on the screened subnet and allows only these to access the protected network. It blocks all other traffic to the protected domain.

Again, note that the SOAP receiver itself, or the container environment that houses it, must not be accessible directly from the internet.

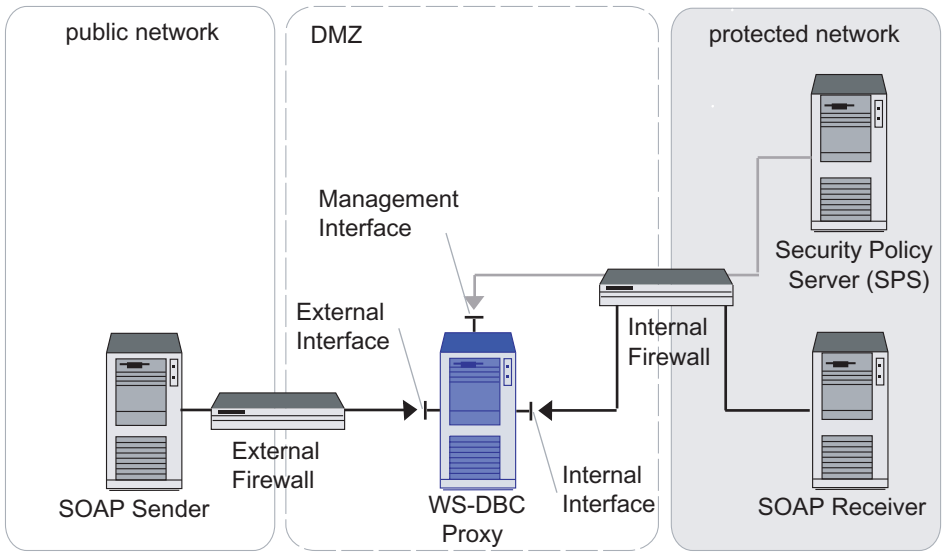


Fig. 3. WS-DBC Proxy in the DMZ

The Security Policy Server runs on a host in the protected network. The WS-DBC Proxy contacts the Security Policy Server to decide whether a contacting client shall be allowed to access the Web Service. In this scenario the Admin Console could be located on the same host as the Security Policy Server.

Exterior Firewall Configuration

As far as the access to application servers is concerned the exterior firewall would route traffic according to the following rules:

- Allow traffic from selected external sites to specific ports of the WS-DBC Proxy host (client initiated HTTP(S) connections).
- Allow traffic from the WS-DBC Proxy to internet systems.
- Reject all other traffic.

Interior Firewall Configuration

The inner firewall passes HTTP traffic to and from the WS-DBC Proxy to the screened subnet according to the following rules:

- Allow traffic from the WS-DBC Proxy to site systems (for example to the application server hosting the Web Services).
- Allow traffic from site systems to the WS-DBC Proxy:
  - specifically, the control connection from the Security Policy Server to the WS-DBC Proxy must be allowed (cf. “Internal Communication Links and Trust Relationships” on page 63).
- Reject all other traffic.

CHAPTER

2 Installing the DBC

This chapter guides you through the installation process for the DBC software. It provides a quick overview of the general installation process, followed by step by step instructions on how to install the DBC Proxy, the Security Policy Server, and the Administration Console.

The installation of a WS-DBC Proxy is similar to the installation of an I-DBC proxy. Therefore the following description applies to both types of Proxies.

2.1 Introduction

The Security Policy Server and the DBC Proxy can be installed on a Linux or Solaris environment. The Admin Console can be installed on Linux, Solaris, and Windows. 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 uses 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 glibc-2.2 / Linux glibc-2.3

- 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`
- The directory `resources/` contains:
  - RPM package for the **I-DBC Proxy**: `Xtradyne_IDBC-3.1.<x>.rpm`
  - RPM package for the **WS-DBC Proxy**: `Xtradyne_WSDBC-3.1.<x>.rpm`
  - RPM package for the **SPS**: `Xtradyne_SPS-3.1.<x>.rpm`
  - RPM package for the **SPS Client**: `Xtradyne_CLI-3.1.<x>.rpm` (a command line interface to the SPS, described in detail in Chapter 4 “SPS Client” on page 41 of the Deployment Guide).

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`
- The directory `resources/` contains:
  - Package for the **I-DBC Proxy**: `Xtradyne_IDBC-3.1.<x>.pkg`
  - Package for the **WS-DBC Proxy**: `Xtradyne_WSDBC-3.1.<x>.pkg`
  - Package for the **SPS**: `Xtradyne_SPS-3.1.<x>.pkg`
  - Package for the **SPS Client**: `Xtradyne_CLI-3.1.<x>.pkg` (a command line interface to the SPS, described in detail in Chapter 4 “SPS Client” on page 41 of the Deployment Guide).

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: Documentation

- 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.

2.1.1 Mounting the CD ROM

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

Mounting on Linux

Usually, a CD can be mounted to make it available to the system by typing:

`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.

Mounting on Solaris

Usually the CD is 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:

```
mkdir /cdrom/dbc_cd
```

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

2.2 Installer

This section describes in detail how to install the DBC using the installer (`IDBCInstall-3.1.<x>.bin` or `WSDBCInstall-3.1.<x>.bin` respectively). The manual installation of DBC components is described in detail in section “Advanced Installation” on page 85.

2.2.1 Prerequisites

To run the DBC installer root privileges are required. If running on a hardened system. make sure that `sh` can execute commands.

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 74).

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`).

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

2.2.2 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 4, “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.

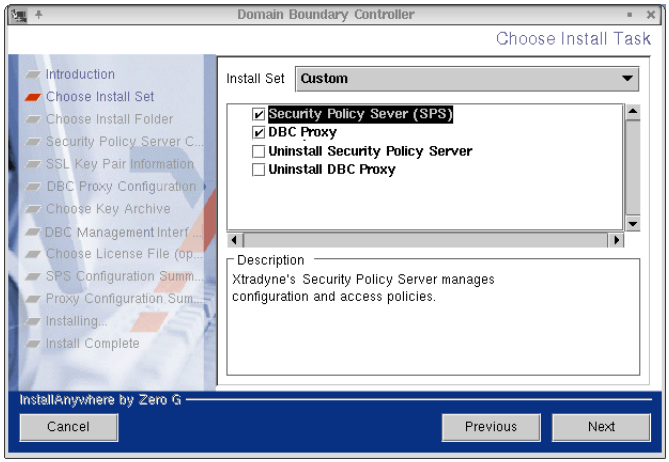


Fig. 4. Choosing Installation Tasks

2.2.3 Choose the Install Folder

Please enter an installation target directory, as shown in figure 5, “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.

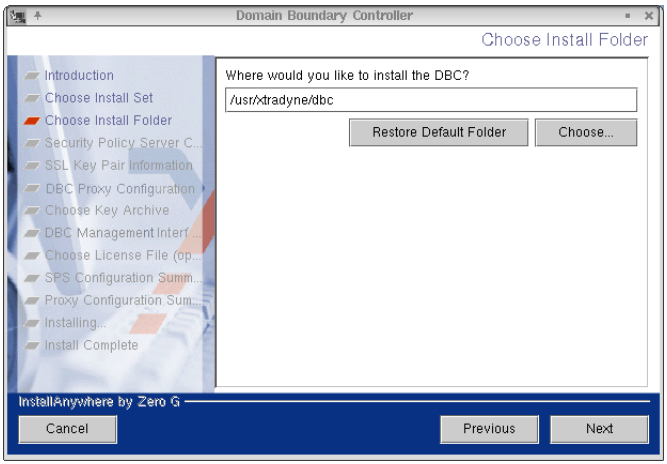


Fig. 5. Choosing an installation directory

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**).

2.2.4 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. For more details, please refer to “The DBC Installation Account” on page 92.

2.2.5 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 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 Security Policy Server can be contacted.

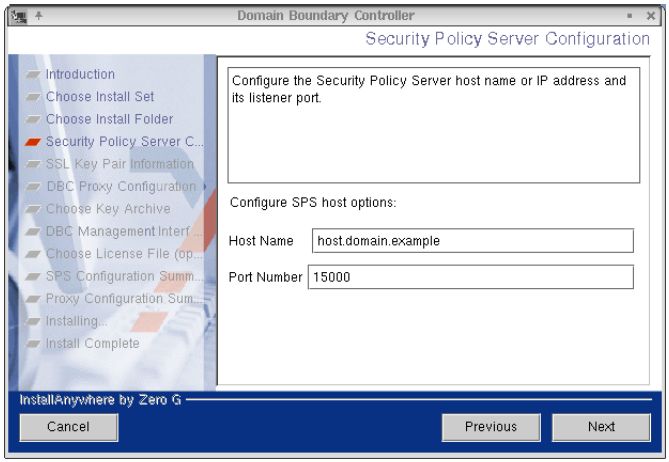


Fig. 6. Entering configuration data for the Security Policy Server

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).

2.2.6 Choose SSL Key Pair Information

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

During the 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.

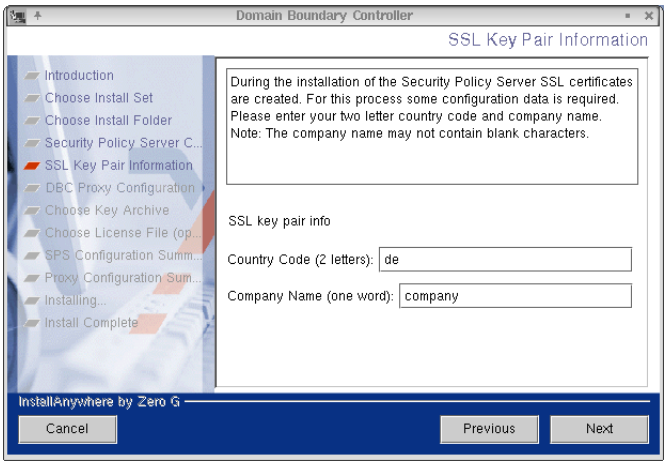


Fig. 7. SSL Key Pair information

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.

2.2.7 DBC Proxy Configuration

The next step in the installation process is to enter basic configuration information for the DBC Proxy, as shown in figure 8, "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.

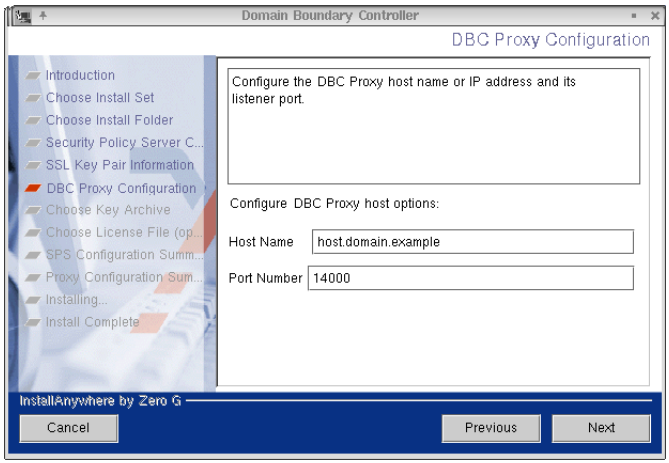


Fig. 8. Entering configuration data for the Security Policy Server

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). The default listener port for the **I-DBC** is 14000. The default listener port for the **WS-DBC** is 14001.

2.2.8 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 on how to install the keys manually.

2.2.9 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. Note that DBC and SPS will not start if no license file is present. Please see gray box “Installing the License” on page 88.

2.2.10 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.

2.2.11 Uninstalling

To uninstall the DBC Proxy or the Security Policy Server choose “Uninstall Security Policy Server” or “Uninstall DBC Proxy” in the installer. The DBC Proxy and the Security Policy Server will be stopped and all components of the selected product will be removed. Note that the DBC installation account (by default `xtradyne`) will not be removed. Please refer to page 97 on how to remove this user account.

2.2.12 What is Installed and Where: Security Policy Server

By default the Security Policy Server is installed in `/usr/xtradyne/sps` on Linux and in `/opt/xtradyne/sps` on Solaris. The table below lists the installed directories and files. Note that the `bin` directory contains some tools that are related only to the I-DBC Proxy or to the WS-DBC Proxy respectively. The I-DBC is the IIOP Domain Boundary Controller, Xtradyne’s DBC for CORBA, and the WS-DBC is the Web Services Domain Boundary Controller, Xtradyne’s DBC for Web Services.

Directory	**Description**
env.sh	Sets the appropriate shell environment (bash and sh) for Security Policy Server commands.
env.csh	Sets the appropriate shell environment (csh and tcsh) for Security Policy Server commands.

bin/	Contains the binaries.
bin/addproxy.sh	Script to add a Proxy to the configuration file. Used for initial configuration (cf. “Adding a Proxy” on page 89).
bin/cfrestore	Binary called by configrestore.sh (see below).
bin/checksps.sh	Script to check the status of the Security Policy Server.
bin/collectperfdata.sh	Helper script to output performance measurement data of a DBC Cluster. The results are written to a CSV file (`adm/PerfData.csv` by default).
bin/configrestore.sh	Script for restoring a security policy (cf. “Policy Versioning and Roll-back” on page 105).
dbcstat	used by `listconnections.sh`
bin/der2pem.sh	Shell script to convert key and certificate files from DER to PEM encoding (cf. “Changing the Certificate Encoding Format” on page 216).
bin/dictionarymerge	Helper program for the setup process.
bin/event2trap.sh	Tool to generate SNMP traps from Xtradyne event messages.
bin/generateior	Tool to generate initial IORs (I-DBC).
bin/generatekeys.sh	Script to generate keys and certificates.
bin/getdictvalue	Helper program for the setup process.
bin/killprog.sh	Helper to kill a process.
bin/listconnections.sh	Lists current connections of the DBC.
bin/mergeior	Generates an IOR with multiple profiles from two IORs or more (I-DBC).
bin/openssl	This is used by the script `generatekeys.sh` for the actual creation of keys and certificates.
bin/printcert.sh	Tool for printing certificates in human readable form.
bin/printior	Tool for printing an IOR in a readable way (I-DBC).
bin/proxifyior	Tool for the setup process or command line proxification (I-DBC).
bin/relocate.sh	Relocate an SPS installation to a different directory.

bin/revertaac.sh	Tool to restore the administrative access control policy (cf. “Restoring Administrative Access Control Rules” on page 106).
bin/runsps.sh	This script sets some environment variables prior to executing the Security Policy Server. It is called by start script `xdn_sps`.
bin/setdictvalue	Helper program for the setup process.
bin/showciphers.sh	List possible SSL cipher suites.
bin/snmptrapgen	Tool for SNMP support.
bin/sps	The Security Policy Server program which is started by the script `runsps.sh` (see above).
bin/spsclient	Command line utility for use in scripts.
bin/spsconfig.sh	Provides the SPS with initial configuration (cf. “Initial Configuration of the Management Network Interface” on page 89).
bin/stripdict	Helper program to remove parts of the configuration file.
bin/xtradyne.sh	Collection of common things for Xtradyne scripts. This is sourced by all other scripts.
lib/	Dynamic and support libraries for the Security Policy Server.
adm/	Contains configuration information, logfiles and keys.
adm/dbc.config	Configuration file, initial configuration information.
adm/ldap_data/	Contains the access control data.
adm/XtradyneEvent-MIB.txt	MIB definition file for SNMP traps.
keys and certificates	Keys and certificates of the DBC installation are explained in chapter “Installing Keys and Certificates” on page 201.

2.2.13 What is Installed and Where: DBC Proxy

Unless a different directory is specified the I-DBC Proxy is installed into:

`/usr/xtradyne/idbc` on Linux and `/opt/xtradyne/idbc` on Solaris.

Unless a different directory is specified the WS-DBC Proxy is installed into:

`/usr/xtradyne/wsdbc` on Linux and `/opt/xtradyne/wsdbc` on Solaris.

The Proxy installation directory contains the following (note that the listing of the `bin` directory contains some binaries that are related to the I-DBC Proxy or to the WS-DBC Proxy respectively):

Directory	**Description**
env.sh	Source this script to set the appropriate shell environment (bash and sh) for DBC commands.
env.csh	Source this script to set the appropriate shell environment (csh and tcsh) for DBC commands.
bin/	Contains the binaries.
bin/checkproxy.sh	Script to check the status of the DBC Proxy.
bin/dbcmon	Component for high availability demands which monitors the availability of the DBC Proxy.
bin/der2pem.sh	Shell script to convert key and certificate files from DER to PEM encoding.
bin/getdictvalue	Helper program for the setup process.
bin/iproxy	The I-DBC Proxy binary which is started by the `iproxymanager`.
bin/iproxymanager	The I-DBC “Master process” binary started by `runiproxy.sh`.
bin/killprog.sh	Helper to kill a process.
bin/logger	Tool to administer and archive log files.
bin/openssl	Tool to create keys and certificates
bin/printcert.sh	Tool to print certificates in a human readable form.
bin/printior	Tool for printing an IOR in a readable way (I-DBC).
bin/proxyconfig.sh	Script to provide the DBC Proxy with initial configuration data.
bin/relocate.sh	Relocate an SPS installation to a different directory.
bin/runiproxy.sh bin/runwsproxy.sh	Script which sets the appropriate environment variables prior to executing the proxy manager (called by the script `xdn_idbc` or `xdn_wsdbc` respectively).
bin/setdictvalue	Helper program for the setup process.
bin/showciphers.sh	List possible SSL cipher suites.

bin/wsproxy	The WS-DBC Proxy binary which is started by the `wsproxymanager`.
bin/wsproxymanager	The WS-DBC "Master process" binary started by `runwsproxy.sh`.
bin/xdn_idbc	Starts the I-DBC Proxy (included only in the I-DBC installation).
bin/xdn_wsdbc	Starts the WS-DBC Proxy (included only in the WS-DBC installation).
bin/xtradyne.sh	Collection of common things for Xtradyne scripts. This is sourced by all other scripts.
lib/	Dynamic libraries for the proxy manager, proxy.
adm/	Contains configuration information, logfiles and keys.
adm/proxymanager.config	Configuration file, initial configuration information for the proxy manager (host, port).
adm/XtradyneAttachmentFilter.war	WAR archive containing a servlet reference implementation that can be used to filter SOAP attachments.
keys and certificates	Keys and certificates of the DBC installation are explained in chapter "Installing Keys and Certificates" on page 201.

2.2.14 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 on Linux 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 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 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 lists which changes in the configuration require a restart of the DBC.

Determining the Status of the DBC Proxy or SPS

To get more detailed status information about the I-DBC Proxy, the WS-DBC Proxy, and the Security Policy Server you can use the scripts:

```
<INSTALLDIR>/idbc/bin/checkproxy.sh
<INSTALLDIR>/wsdbc/bin/checkproxy.sh
<INSTALLDIR>/sps/bin/checksps.sh
```

The Security Policy Server's management port can be given as the additional option `-p <port>` (by default 15000).

Example

```
~:/usr/xtradyne/idbc/bin/checkproxy.sh
proxy…        2 processes/threads
proxymanager…       12 processes/threads
Listening on
tcp  0  0 192.168.1.90:8884 0.0.0.0:*   LISTEN    31038/proxymanager
tcp  0  0 192.168.1.90:8885 0.0.0.0:*   LISTEN    31038/proxymanager
tcp  0  0 0.0.0.0:15000     0.0.0.0:*   LISTEN    31038/proxymanager
Current connections:
tcp 0  0 192.168.1.90:2316 192.168.1.90:15000 ESTABLISHED 31038/
proxymanager
tcp 0  0 192.168.1.90:15000 192.168.1.90:2315  ESTABLISHED 31038/
proxymanager
```

In this example the proxy manager is listening on the ports 8884 (external plain IIOP acceptor), 8885 (external IIOP/SSL acceptor), and 15000 (listener for the connection to the Security Policy Server). The proxy manager has currently two connections: one from the Security Policy Server and one to the SPS).

If you operate a cluster the `checksps.sh` script will not yield information about the other Security Policy Servers in the cluster. You will have to call `checksps.sh` on every host where a Security Policy Server runs.

Finding out the Status with Linux/Unix Commands

Alternatively, the status of the Security Policy Server, the I-DBC Proxy, or the WS-DBC Proxy can be determined with the commands `ps` and `netstat`. The following examples grep for the SecurityServer. To determine the status of the DBC Proxy replace the string `sps` with `proxy` or `proxymanager`.

Linux: The ps and netstat Commands

```
~:/usr/xtradyne/sps > ps aux|grep sps
xtradyne  297 0.0  3.5 13788 9164 ?  S  10:17 0:00 ../bin/sps
```

With the `netstat` command you can determine the port number the Security Policy Server is listening on (execute the command `netstat` as root):

```
~:/usr/xtradyne/sps> netstat -tnlp|grep sps
tcp    0    0   0.0.0.0:15000  0.0.0.0:*      LISTEN 297/sps
```

Solaris: The ps Command

```
~:/usr/xtradyne/sps > ps -ef | grep sps
xtradyne 6763 6757 0 11:45:41 pts/15 0:02 /opt/xtradyne/sps/bin/sps
```

Unfortunately, the standard `netstat` command on Solaris does not support any options to determine listeners. To see all listeners you can use the command:

```
~:/usr/xtradyne/sps > ps -affe | netstat -a | grep LISTEN
```

2.3 Advanced Installation

The installation of DBC components can also be done manually, for example, when the installation is to be included in complex installation environments (e.g., Citrix) to facilitate the installation on multiple hosts.

Installation packages for Linux and Solaris can be found in the directory `resources`. The following sections describe manual installation of packages.

2.4 Installation Steps

Here is an overview of the steps that have to be executed to install the Security Policy Server and the DBC Proxy. This section lists only the main steps, you will find a detailed description of every step in the following sections.

1. To install the **Security Policy Server**:
   - install the Security Policy Server package (`Xtradyne_SPS-3.1.<x>.rpm` on Linux and `Xtradyne_SPS-3.1.<x>.pkg` on Solaris[1])
   - provide the Security Policy Server with initial configuration information by running the shell script `spsconfig.sh`
   - use the script `addproxy.sh` to add a DBC to the configuration
   - generate keys and certificates by running the shell script `generatekeys.sh`
   - copy the generated files containing the keys `ProxyKeys.tar` and `AdminConsoleKeys.tar` onto a moveable medium (e.g., a floppy disk)
   - copy the license file into `<INSTALLDIR>/sps/adm/license.txt`
2. To install the **DBC Proxy**:
   - install the DBC Proxy package:

     I-DBC Proxy: `Xtradyne_IDBC-3.1.<x>.rpm` on Linux and `Xtradyne_IDBC-3.1.<x>.pkg` on Solaris

     WS-DBC Proxy: `Xtradyne_WSDBC-3.1.<x>.rpm` on Linux and `Xtradyne_WSDBC-3.1.<x>.pkg` on Solaris
   - install keys and certificates by unpacking the tar file `ProxyKeys.tar` (this file is generated during the installation of the Security Policy Server)
   - provide the DBC Proxy with initial configuration information by running the shell script `proxyconfig.sh`
   - copy the license file (`license.txt`) into the `adm` directory of the proxy.

For the installation steps described in the following sections you need to have `root` privileges when installing the packages. For all other tasks, you must use the DBC installation account (by default `xtradyne`) which is created during installation. In case of any problems that arise during the installation, please refer to chapter 11, “Trouble-shooting” on page 217.

1 `<x>` denotes the release’s micro number.

2.5 *Installing the Security Policy Server (SPS)*

Typically the Security Policy Server will be installed on a host located in a protected network. In environments with less stringent security requirements or for evaluation purposes all three components can be installed on a single host.

To install the Security Policy Server you need to have root privileges. Installing the Security Policy Server will take some minutes. During the installation process the following steps are executed:

- An administrative UNIX user account (by default `xtradyne`) is created (cf. section “The DBC Installation Account” on page 92).
- All files are placed by default in the directory `/usr/xtradyne/`on Linux and `/opt/xtradyne/`on Solaris. A subdirectory `sps/` is created.
- The start/stop script `xdn_sps` is installed in `/etc/init.d/`.
- A link to this script is installed in `/sbin/rcxdn_sps` on Linux and `/usr/sbin/rcxdn_sps` on Solaris.
- The symbolic links `S95xdn_sps` for starting and `K05xdn_sps` for shutting down are created in the directories `/etc/init.d/rc3.d/` and `/etc/init.d/rc5.d/`. This applies to Solaris and RedHat Linux; with SuSE Linux these links are created automatically by the operating system and might be called slightly different.

In the following we will write `<INSTALLDIR>` to refer to the home directory of the DBC. By default this is `/usr/xtradyne/` on Linux or `/opt/xtradyne/` on Solaris. When installing a cluster of Security Policy Servers you have to execute the above steps on every SPS host.

Linux: Installation Command

With the installation the user `xtradyne` is created. If you want to use a different account name, set and export the environment variable `DBC_USER`. If you use the command shell bash you can do this by typing: `export DBC_USER=myuser`. Install the RPM package by typing (as root!):

```
rpm -ivh Xtradyne_SPS-3.1.<x>.rpm
```

If you want to install to a different directory use the `--prefix` option (not possible using RPM 4.0, e.g., RedHat 8.0):

```
rpm -ivh --prefix /different_directory ...
```

For more information about the installed package, e.g., date of installation, the version number, etc., use the command `rpm -q -i Xtradyne_SPS`.

Solaris: Installation Command

Install the package by typing (as root!):

```
pkgadd -d Xtradyne_SPS-3.1.<x>.pkg
```

The script is running with root permissions and will ask for your consent (answer “yes”). During the installation you are prompted for an installation directory and for a user and group to use as installation account. If these do not exist yet, they are created. By default the user is `xtradyne` and the group is `sys`.

Note that when choosing an installation directory with three elements (e.g., `/home/project/dbc`), the `pkgadd` command will display the installation directory incorrectly (e.g., the output will say “Using `</home/project>` as the package base directory”). However the package will be installed in the correct directory.

If the Security Policy Server has been installed previously, the script will ask you whether it shall overwrite any existing files. We strongly recommend to first deinstall any previous installation. For more information refer to section “Deinstalling the Software” on page 97. For more information about the installed package, e.g., the date of installation, the version number, etc., use `pkginfo -l -i XDNSPS`.

2.5.1 Postinstallation Steps

You are done with installing the Security Policy Server! What is left to do is:

- Install the license on the Security Policy Server host
- Run a script that provides the Security Policy Server with initial configuration data
- Create keys and certificates for the SPS, the DBC Proxy, and the Admin Console

Installing the License

You can obtain a licence from your reseller or directly from Xtradyne (mail to `info@xtradyne.com`). Copy the license file (`license.txt`) you received with your software package or via email:

- on the **SPS** host into the directory `<INSTALLDIR>/sps/adm`,
- on the **I-DBC Proxy** host into `<INSTALLDIR>/idbc/adm`,
- on the **WS-DBC Proxy** host into `<INSTALLDIR>/wsdbc/adm`.

Initial Configuration of the Management Network Interface

define the Management Network Interface

To provide the Security Policy Server with initial configuration information, i.e., to configure the Management Network Interface, execute the script `spsconfig.sh`. It is located in the directory `<INSTALLDIR>/sps/bin/` and has to be given the following arguments:

```
./spsconfig.sh <host> [<port> <name>]
```

`<host>` and `<port>` define the network address that will be used by the Security Policy Server. The DBC Proxy and the Admin Console contact this interface to communicate with the Security Policy Server. This communication is SSL protected. The `<port>` for the Security Policy Server's local interface is 15000 by default. Change this number only if absolutely necessary, for example, if the default port is not a free port. Note that if there is a firewall between the Security Policy Server and the DBC Proxy, the port must be opened on the firewall.

using the host name or IP address

When using host names instead of the IP addresses, ensure that proper name resolution is available. A DBC Proxy host must be able to resolve the Security Policy Server host name. Additionally, the Security Policy Server must always be able to resolve its given host name because that is the interface it binds to.

The argument `<name>` is the logical name of the SPS. `<port>` and `<name>` are optional arguments and can be left out.

Example

```
cd <INSTALLDIR>/sps/bin/
./spsconfig.sh 192.168.47.11
```

Adding a Proxy

To add a proxy to the configuration, use the script `addproxy.sh` located in the directory `<INSTALLDIR>/sps/bin`. It has to be given the following arguments:

```
addproxy.sh ["wsdbc"|"idbc"] <host> [<port>]
```

With the argument `idbc` an I-DBC Proxy will be added and with `wsdbc` a WS-DBC Proxy will be added.

`<host>` and `<port>` define the local interface that will be used by the Proxy. The `<port>` for the DBC Proxy's local interface is 14000 by default. Change this number only if absolutely necessary, for example, if the default port is not a free port. Note that if there is a firewall between the DBC Proxy and the Security Policy Server, the port must be opened on the firewall.

using the host name or IP address

When using host names instead of the IP addresses, ensure that proper name resolution is available. A Security Policy Server host must be able to resolve the DBC Proxy host name. Additionally, the DBC Proxy must always be able to resolve its given host name because that is the interface it binds to.

Example

```
addproxy.sh idbc 192.168.47.11
```

Generating Keys

This section describes how keys and certificates can be generated with the shell script `generatekeys.sh`. 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 script, is given in chapter "Installing Keys and Certificates" on page 201. If you would like to generate keys for a cluster scenario, please refer to section "Generating Keys for a Cluster of Security Policy Servers" on page 91.

For scenarios with only one Security Server please follow these steps:

1. You have to run the script as the DBC installation user (`xtradyne` by default). If you are `root` the script will switch to this user automatically.
2. Change to the directory `/<INSTALLDIR>/sps/bin/`.
3. Execute the script `generatekeys.sh` on the Security Policy Server host. The script has to be given two arguments (distinguished name fields for generating certificates):

   `./generatekeys.sh <country> <company>`

   Example: `./generatekeys.sh de xtradyne`

   The first parameter `<country>` must be the 2-letter shortcut (`us` for United States of America, `uk` for the United Kingdom, `de` for Germany, etc).
4. The keys and certificates used on the Security Policy Server host will be placed in the directory `/<INSTALLDIR>/sps/adm`.
5. The keys and certificates for other Security Policy Servers in the cluster will be packed into the tar file `<INSTALLDIR>/sps/adm/ClusterKeys.tar`.
6. The keys and certificates used on the DBC Proxy host will be packed into the tar file `<INSTALLDIR>/sps/adm/ProxyKeys.tar`.
7. The keys and certificates used on the Management host will be packed into the tar file `<INSTALLDIR>/sps/adm/AdminConsoleKeys.tar`.

8. Copy these tar files on a moveable medium; you will need them later when installing the DBC Proxy (next section) and the Admin Console (chapter 3). This step can be skipped if you install on a single host.

All the keys generated by this script are by default passphrased with the string `xtradyne`.

Overview of Keys and Certificates on the SPS

The following key files are located in `<INSTALLDIR>/sps/adm` after you completed the steps previously described (this applies for Linux and Solaris):

- The Security Policy Server private key: `SPSKey.der`
- The Security Policy Server certificate: `SPSCert.der`
- The certificate of the certification authority: `ControlConnectionCACert.der`

Generating Keys for a Cluster of Security Policy Servers

Note that before generating keys and certificates for a cluster of Security Policy Servers make sure that the clocks of the hosts you want to install on are synchronized. Otherwise generated keys might be invalid.

To generate keys and certificates for a cluster of Security Policy Servers, do the following on every Security Policy Server host:

- Copy the file `ClusterKeys.tar` (created by `generatekeys.sh` on the "first" Security Policy Server host, see previous sections, step 5.) to the directory `<INSTALLDIR>/sps/adm`.
- Generate keys and certificates using the trusted CA included in the tar file:

```
generatekeys.sh -c ClusterKeys.tar
```

This will install the keys listed in the previous section in the directory `<INSTALLDIR>/sps/adm`.

Checking Permissions for Key Files

You should check if the permissions for the keys are set correctly. Only the DBC installation user (`xtradyne` by default) should have read and write permissions. Determine the permissions with the command `ls -la` in the `adm` directory of the SPS and the Proxy, for example:

```
root@myhost:/usr/xtradyne/sps/adm > ls -la *.der *.pem
-rw------- 1 xtradyne users 592 Jul 9 17:07 CACert.der
-rw------- 1 xtradyne users 654 Jul 9 17:07 SPSCert.der
-rw------- 1 xtradyne users 677 Jul 9 17:07 SPSKey.der
```

The first column of the output is the important one. Read and write permissions (rw) should only be set for the user (the second to fourth character apply to the user); for group (next three) and world or others (last three) neither read nor write permissions should be allowed (this is indicated by the dashes)!

If permissions are not ok, (as root) use

```
chmod 600 *.der.*.pem
```

Keys must be owned by the DBC installation user (`xtradyne` by default). If this is not the case type:

```
chown xtradyne *.der *.pem
```

The DBC Installation Account

During the installation an account is created (`xtradyne` by default). When installing under Solaris you will be asked to choose an account name for the DBC installation account. When installing under Linux you have to set and export the environment variable `DBC_USER` if you do not want to use the default (`xtradyne`). If you use the command shell bash you can do this by typing: `export DBC_USER=myuser`.

For security reasons the DBC installation account has no password. To login as the DBC installation user you must first become `root` and then use the `su` command. For example: `su xtradyne`.

2.6 Installing the DBC Proxy

Typically the DBC Proxy is installed on a dedicated host within a screened subnet which is part of the existing firewall. To install the DBC Proxy you need to have root privileges. Installing the DBC Proxy will take some minutes. In this process the following steps are executed:

- An administrative UNIX user account (by default `xtradyne`) is created (see above).
- All files are placed in the directory `/usr/xtradyne/` on Linux and in the directory `/opt/xtradyne/` on Solaris. When installing an I-DBC Proxy a subdirectory `idbc` is created. When installing a WS-DBC Proxy a subdirectory `wsdbc` is created.
- The start/stop script is installed in `/etc/init.d/`:
  - **I-DBC Proxy**: `xdn_idbc`
  - **WS-DBC Proxy**: `xdn_wsdbc`
- A link to this script is installed in `/sbin` on Linux and in `/usr/sbin` on Solaris:
  - **I-DBC Proxy**: `rcxdn_idbc`
  - **WS-DBC Proxy**: `rcxdn_wsdbc`
- Symbolic links for starting and for shutting down are created in `/etc/init.d/rc3.d` and `/etc/init.d/rc5.d`:
  - **I-DBC Proxy**: `S95xdn_idbc` for starting and `K05xdn_idbc` for shutting down
  - **WS-DBC Proxy**: `S95xdn_wsdbc` for starting and `K05xdn_wsdbc` for shutting down

  This applies to Solaris and RedHat Linux; with SuSE Linux these links are created automatically by the operating system and might be called slightly different.

In the following we will write `<INSTALLDIR>` to refer to the home directory of the DBC. By default this is `/usr/xtradyne/` on Linux or `/opt/xtradyne/`on Solaris. When installing a cluster of DBCs you have to execute the above steps on every DBC Proxy host.

Linux: Installation Command

With the installation the user `xtradyne` is created. If you want to use a different account name, set and export the environment variable `DBC_USER`. If you use the command shell bash you can do this by typing: `export DBC_USER=myuser`. Install the RPM package by typing (as root!):

I-DBC:

```
rpm -ivh Xtradyne_IDBC-3.1.<x>.rpm
```

WS-DBC:

```
rpm -ivh Xtradyne_WSDBC-3.1.<x>.rpm
```

If you want to install into a different directory use the `--prefix` option (not possible using RPM 4.0, e.g., RedHat 8.0):

```
rpm -ivh --prefix /different_directory ...
```

For more information about the installed package, e.g., date of installation, the version number, etc., use the command:

`rpm -q -i Xtradyne_IDBC` (for **I-DBC** Proxies) or

`rpm -q -i Xtradyne_WSDBC` (for **WS-DBC** Proxies).

Solaris: Installation command

Install the package by typing:

I-DBC:

```
pkgadd -d Xtradyne_IDBC-3.1.<x>.pkg
```

WS-DBC:

```
pkgadd -d Xtradyne_WSDBC-3.1.<x>.pkg
```

During the installation you are prompted for an installation directory and for a user (by default `xtradyne`) and a group (by default `sys`) to use as installation account. If these do not exist yet, they are created. Furthermore, you need to confirm that scripts are run as root.

Note that when choosing an installation directory with three elements (for example, the directory `/home/project/dbc`, the `pkgadd` command will display the installation directory incorrectly (e.g., the output will say "Using `</home/project>` as the package base directory"). However, the package will be installed in the correct directory.

For more information about the installed package, e.g., the date of installation, the version number, etc., use the command:

`pkginfo -l -i XDNIDBC` (for **I-DBC** Proxies) or

`pkginfo -l -i XDNWSDBC` (for **WS-DBC** Proxies).

2.6.1 Postinstallation Steps

You are almost done with installing the DBC Proxy. All you have to do now is:

- install the license on the DBC Proxy host (please refer to the gray box “Installing the License” on page 88),
- install keys and certificates on the DBC Proxy host, and
- run a script that provides the DBC Proxy with initial configuration information.

Installing Keys and Certificates on the DBC Proxy Host

The keys and certificates needed on the DBC Proxy host were generated by the `generatekeys.sh` script and packed into the tar file `ProxyKeys.tar`. This was done when installing the Security Policy Server, see “Generating Keys” on page 90.

Execute the following steps as the DBC installation user (`xtradyne` by default). As root type, for example: `su - xtradyne`:

1. Copy the file `ProxyKeys.tar` into the `adm` directory of the Proxy on the Proxy host. You can use a removable medium like a floppy disk if no network connection is available for `scp`. (If all DBC components are installed on a single host, you can use `cp` of course.)
2. `cd` to this directory.
3. Unpack the tar file by typing: `tar xpf ProxyKeys.tar`.

The option `p` given in the `tar` command stands for “preserve permissions”. However you should check if permissions for the keys are set correctly (see gray box on page 92).

Overview of Keys and Certificates on the DBC Proxy

The following key files are in the destination install directory after you completed the steps previously described (this applies for Linux and Solaris):

- The Internal and External Public Key: `ProxyChain.pem`
- The Control Connection Private Key: `ControlConnectionKey.der`
- The Control Connection Certificate: `ControlConnectionCert.der`
- The certificate of the Proxy Certification Authority in PEM and in DER format: `ProxyCACert.pem`, `ProxyCACert.der`
- The certificate of the control connection Certification Authority in DER format: `ControlConnectionCACert.der`

A detailed description of the key management in the DBC is given in chapter 10, “Installing Keys and Certificates” on page 201.

Initial Configuration of the Security Policy Server Interface

Execute the following steps as the DBC installation user (`xtradyne` by default). As root type, for example: `su - xtradyne`.

To provide the DBC Proxy with initial configuration information execute the shell script `proxyconfig.sh`. The script is located in the `bin` directory of the Proxy and has to be given the following arguments:

```
./proxyconfig.sh <host> [<port>]
```

`<host>` and `<port>` define the local interface that will be used by the DBC Proxy. The Security Policy Server uses this interface to contact the DBC Proxy.

using the host name or IP address

Note that when using the host names instead of the IP addresses, ensure that proper name resolution is available. A Security Policy Server host must be able to resolve the DBC Proxy host name. Additionally, the DBC Proxy must always be able to resolve its given host name because that is the interface it binds to.

For the **I-DBC** the `<port>` for the DBC Proxy’s local interface is 14000 by default. For the **WS-DBC** the `<port>` for the DBC Proxy’s local interface is 14001 by default. Change this number only if absolutely necessary, for example, if this port is not a free port. Note that if there is a firewall between the Security Policy Server and the DBC Proxy this port must be opened on your firewall.

Example

```
cd /<INSTALLDIR>/idbc/bin
./proxyconfig.sh 192.168.47.11
```

Starting the DBC Proxy

Before starting the DBC Proxy after a manual setup you should configure the DBC Proxy Network Interface with the Admin Console (please see “DBC Proxy Configuration” on page 135). Otherwise the Security Server will not find the DBC Proxy on startup. For more configuration details, please see chapter 4, “Configuration of DBC components” on page 103.

2.7 Deinstalling the Software

To deinstall the DBC Proxy or the Security Policy Server, run the following commands (as root) on the respective hosts.

Note that before deinstalling make sure to stop the Security Policy Server, the I-DBC Proxy, or the WS-DBC. As root use the command:

- `/etc/init.d/xdn_idbc stop` to stop the I-DBC Proxy
- `/etc/init.d/xdn_wsdbc stop` to stop the WS-DBC Proxy
- `/etc/init.d/xdn_sps stop` to stop the Security Policy Server.

Deinstalling on Linux

Deinstall the SPS, the I-DBC Proxy, and the WS-DBC Proxy by typing:

- to deinstall the I-DBC Proxy: `rpm -e Xtradyne_IDBC`
- to deinstall the WS-DBC Proxy: `rpm -e Xtradyne_WSDBC`
- to deinstall the SPS: `rpm -e Xtradyne_SPS`

This will leave a tar ball with the keys. Files located in the directory `<INSTALLDIR>/sps/` ending with `*.old` are only temporary files which can safely be deleted after the installation process has finished. The installation account (`xtradyne` by default) is not removed automatically. If you do not need the installation account anymore, type for example:

```
userdel xtradyne
```

Deinstalling on Solaris

Deinstall the DBC Proxy and the Security Policy Server by typing:

- to deinstall the I-DBC Proxy: `pkgrm XDNIDBC`
- to deinstall the WS-DBC Proxy: `pkgrm XDNWSDBC`
- to deinstall the SPS: `pkgrm XDNSPS`

This will leave the `adm` directory containing the keys. The installation account (`xtradyne` by default) are not removed automatically. If you do not need the installation account anymore, type for example:

```
userdel xtradyne
```

CHAPTER

3 Installation of the Admin Console

This chapter gives a detailed description on how to install the Admin Console.

3.1 What's included in the Distribution?

The Admin Console can be installed on Windows, Linux, or Solaris. In the distribution (archive or CD-ROM) you will find the directories `linux_glibc-2.2`, `linux_glibc-2.3`, `solaris`, and `windows`. These directories contain the executables for the respective operating systems that will install the Admin Console (`AdminConsoleInstall-3.1.<x>.bin` for Linux and Solaris and `AdminConsoleInstall-3.1.<x>.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.

3.1.1 Mounting the CD ROM on Linux and Solaris

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

Mounting on Linux

Usually, you can mount the CD to make it available to the system by typing:

```
mount /cdrom
```

If this does not work, consult your operating system manuals. After mounting, the files contained on the CD will be available in the directory `/cdrom/`. Now you can execute the file `AdminConsole.bin` in the `linux` directory to start the installation.

Mounting on Solaris

Usually the CD is 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:

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

To start the installation execute the file `AdminConsole.bin` in the `solaris` directory you just mounted.

3.2 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.
4. Installation Summary

3.2.1 What is installed and where on Linux, Windows, and Solaris

After the installation is complete you will find the following directories in the destination install directory (if you choose to install all available install sets): `adminconsole`, `doc`, `jre`, and `tools`.

`adminconsole/`

The `adminconsole` directory contains the Administration Console: You will find the executables `AdminConsole` in the `bin` directory. The `lib` directory contains the archives of the Admin Console and other third party libraries.

`doc/`

In the `doc` directory you will find the following documentation files:

- `AdminGuide.pdf` containing this Administrator's Guide,
- `QuickInstallation.pdf` to help with quick evaluation of the product, and
- `DeploymentGuide.pdf` which explains different deployment scenarios and specialties e.g., high availability.

`jre/`

The directory `jre` contains the java runtime environment needed by the Admin Console and the tools.

`tools/`

The `tools` directory contains some useful tools which are described in detail in Chapter 5 "WS-DBC Tools" on page 49 of the Deployment Guide.

3.3 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).

The keys and certificates to be used on the Management host are generated by a shell script during the setup of the Security Policy Server. They are packed into a tar file (`AdminConsoleKeys.tar`) on the Security Policy Server host, located in the directory `<INSTALLDIR>/sps/adm` (please see also "Postinstallation Steps" on page 88).

On Linux or Solaris: Installing Keys for the Admin Console

To install the SSL Keys on the Management host do the following:

- Copy the file `AdminConsoleKeys.tar` onto the Management host into the directory `<INSTALLDIR>/adminconsole/bin`.
  To do this, you can use a removable medium like a floppy disc.
- Unpack the tar file by typing `tar xpf AdminConsoleKeys.tar`.

The option `p` given in the `tar` command stands for preserve permissions. However you should check if permissions for the keys are set correctly (see gray box on page 92).

Windows: Installing Keys for the Admin Console

To install the SSL Keys on the Management host do the following:

- Copy the file `AdminConsoleKeys.tar` onto the Management host into the directory `<INSTALLDIR>\adminconsole\bin`.
  To do this, you can use a removable medium like a floppy disc.
- Use an archiver tool such as WinZip (`www.winzip.com`) to unpack the files. Choose the `<INSTALLDIR>\adminconsole\bin` folder as destination.

Make sure keys and certificates can only be accessed by authorized users!

Overview of installed Keys and Certificates

The following key files can be found in `<INSTALLDIR>/adminconsole/bin` after you completed the steps described above (this applies for Linux, Windows, and Solaris):

- The Management Client Private Key: `AdminConsoleKey.der`
- The Management Client Certificate: `AdminConsoleCert.der`
- The certificate of the Certification Authority: `ControlConnectionCACert.der`

The Admin Console Private Key is protected by a passphrase. The Admin Console will prompt for this passphrase whenever they need to access one of the keyfiles. The default passphrase is `xtradyne`.

In case of any problems that arise during the installation of the Admin Console, see chapter 11, “Troubleshooting” on page 217.

CHAPTER

4 Configuration of DBC components

This chapter gives an introduction to the Admin Console and general configuration concepts. At this stage we assume that you have already installed one or more DBC Proxies and one or more Security Policy Servers, and the Admin Console as described in the previous chapters. The subsequent chapters describe the panels of the Admin Console in detail.

4.1 Admin Console – Introduction

The Admin Console provides a graphical user interface for configuring and managing different Xtradyne products, i.e., IIOP DBC and the Web Services DBC. This guide explains how to configure different DBC components, including

- single I-DBC and WS-DBC Proxies and Proxy Clusters (chapter 5 on page 123),
- single Security Policy Servers and a Cluster of Security Policy Servers (chapter 7 on page 183),
- Audit Policies (chapter 8 on page 189),
- Security Policies (part 3, chapter 1 on page 235).

Note that in a simple scenario a Cluster may contain only a single Proxy or Security Policy Server as a special case of a cluster, High Availability and Scalability are not supported in this case.

4.2 Quick Start: Typical Configuration Steps

The Admin Console offers a number of panels with various configuration options. For a quick start, most of the default settings can be left unchanged. This section describes how to configure those settings that require modifications. For a complete configuration example, please refer to the example in part 3 of this guide.

- Start DBC components, as user `root` enter the following commands:
  - Security Policy Server: `/etc/init.d/xdn_sps start`
  - I-DBC Proxy: `/etc/init.d/xdn_idbc start`
  - WS-DBC Proxy: `/etc/init.d/xdn_wsdbc start`
  - Admin Console:
    Linux/Solaris: `<INSTALLDIR>/AdminConsole/bin/AdminConsole`
    Windows: Click on the **Start** menu of the task bar and choose
    Programs ➜ Xtradyne Admin Console ➜ Admin Console
- Define Security Policies, e.g., access control rules for resources on the "Resources" panel (for details see "Security Policies" on page 235).
- Define Audit Policies by selecting events for which notifications should be written to an audit log (for see page 189).
- When operating an **I-DBC Proxy**:
  - Configure Initial IORs (on the "Initial IOR Set" panel).
  - On the same panel export proxified IORs to a file and make the proxified IOR available to the client application.
- When operating a **WS-DBC Proxy:**
  - Configure Resource Mappings with the Admin Console (on the "Resource Mappings" tabbing pane, for details see page 150).
- Save the configuration changes to the Security Policy Server by clicking on the "Write to Security Policy Server" icon in the tool bar (or via the "Server" menu).

4.3 Administration Concepts

This section describes the DBC administration concepts, for example, how policy versioning is done. On your first read you may safely skip this section.

4.3.1 Configuration Data

configuration data is maintained by the Security Policy Servers

All the configuration data that is required to operate the DBC is maintained by the Security Policy Servers in the Security Policy Server Cluster. To read and modify configuration data, the Admin Console will access one Security Policy Server (SPS) in the Cluster using a secure communication protocol. Access to the configuration data is protected by a user ID/password scheme, and the communication between the Admin Console and the Security Policy Server is SSL-protected by default.

Security Policy Servers in a cluster are fully cross-linked and are supervising each other. Any change in the configuration data is propagated to every other SPS immediately when the configuration is written to an SPS. Concurrent writing of the configuration is possible (see also section “Write Configuration – Properties” on page 112).

The Admin Console also allows you to write the configuration data to an external file. This may be useful if you wish to keep alternative configuration sets or if you want to make a backup of the active configuration.

Policy Versioning and Roll-back

DBC policy and configuration data is internally versioned. Versioning allows administrators to revert from a current configuration to a previous version of a policy or configuration when they find that the current version

- does not correctly define the desired security policy,
- is not working correctly because of misconfigurations,
- does not allow a clean start of the DBC Proxy and Policy Server because of, e.g., a corrupt file or severe configuration errors.

To roll-back to a previous version of the DBC policy and configuration data administrators need to know the version number of the desired target version. The DBC provides a command-line tool so that administrators, can perform the necessary steps.

To perform a policy roll-back, please perform the following steps on the SPS host:

- stop the SPS if it is running: `/etc/init.d/xdn_sps stop`
- determine the current version of the configuration with the tool: `configrestore.sh log`
  This tool is located in `<INSTALLDIR>/sps/bin/`. Called with the `log` option it lists the version number, the date, and the administrator who carried out the change.
- call the command-line tool to restore the configuration: `configrestore.sh restore <version>`.
- start the SPS: `/etc/init.d/xdn_sps start`.

Example

```
cd <INSTALLDIR>/sps/bin/
./configrestore.sh restore 3
```

The configuration data with the version number 3 will be restored. Restart the SPS, login with the Admin Console and check the restored configuration.

Note that policy roll-back is restricted to the last 15 versions of the policy.

Restoring Administrative Access Control Rules

In case of misconfigurations the administrator might not be allowed to login anymore. The DBC provides a tool that restores the original access control rules. When this tool is run all changes made in the administrative access control policy are lost and the default administrator rights are restored, i.e., a user `admin` with the password `admin` that belongs to the role `admin` is created. This role has the right to administer all parts of the configuration.

Example

Stop all running Security Policy Servers, change to `<INSTALLDIR>/sps/bin`, and type:

```
./revertaac.sh
```

Start the SPS and login as admin (password admin). Change the administrator password to protect the SPS against unauthorized access (cf. “Changing the Admin User’s Password” on page 245).

4.3.2 Importing Configuration Files of previous DBC versions

You can import configuration files of previous DBC versions (as of DBC v. 2.6). Selecting **File ➜ Import ➜ Configuration of Previous Version...** brings up a wizard that will lead you through the import. As a first step, select the file containing the configuration of a previous DBC version.

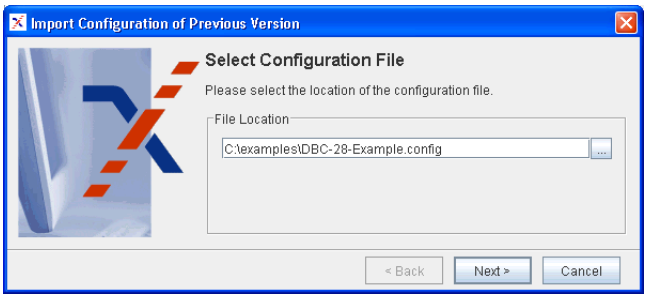


Fig. 9. Import Previous Configuration – Select Configuration File

After pressing the “Next” button, select the version of the chosen configuration file (the correct version should already be pre-selected.

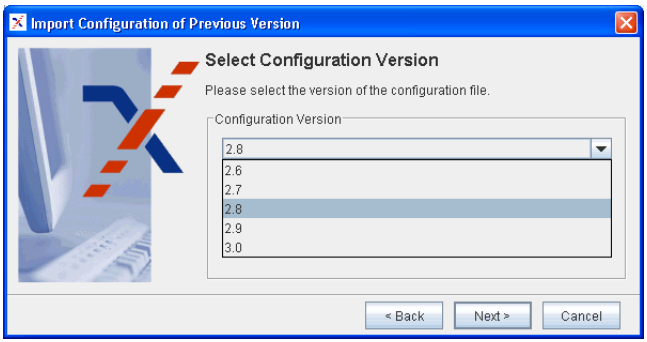


Fig. 10. Import Configuration – Select Configuration Version

After pressing the “Next” button, select the parts of the configuration that are to be converted to the current version. Selectable parts are:

- Security Policy,
- Proxy Cluster,
- Auditing,
- Security Policy Server Cluster,
- WS-Security Profiles

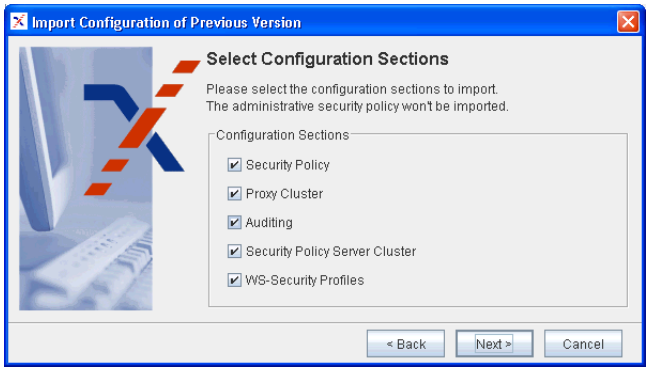


Fig. 11. Import Previous Configuration – Select Configuration Sections

Please revise the imported configuration carefully! Note that the administrative security policy (including the `admin` password) was not imported automatically. Details on how to configure this part of the configuration can be found in section “Role Properties – Administration” on page 261.

4.3.3 Log Files and Log File Backup

For convenient administration of log files the DBC uses `logrotate`. This tool performs automatic rotation, compression, and removal of log files.

For SuSE and RedHat Linux `logrotate` is included in the standard installation. For Solaris operating system `logrotate` is provided as freeware by SUN MicroSystems and can be down loaded at `http://www.sunfreeware.com/`.

If logrotate has been installed and the directory `/etc/logrotate.d` exists on the host system, the installer will create the necessary configuration files to rotate DBC log

files. The log rotated files are `proxy.log` and `sps.log` in the `adm` directory of the SPS and the Proxy. The default configuration specifies that:

- `logrotate` will be executed once a week or when the log file extends 100 MByte,
- backup logs will be compressed,
- not more than 5 backup logs will be kept,
- backup logs can be found in the `adm` directories of the SPS and the Proxy installations.

To change the default setting for log rotation edit the file (please consult the log rotate manual pages for configuration details):

- `/etc/logrotate.d/sps` (for the Security Policy Server),
- `/etc/logrotate.d/idbc` (for the I-DBC Proxy),
- `/etc/logrotate.d/wsdbc` (for the WS-DBC Proxy).

4.4 First Start

To start the Admin Console, execute `<INSTALLDIR>/AdminConsole/bin/AdminConsole` (on Linux and Solaris) or `<INSTALLDIR>/AdminConsole/AdminConsole.exe` (on Windows). The Admin Console will start up with a Login Panel (see below). Should any problems occur while starting the Admin Console, please refer to chapter “Troubleshooting” on page 217.

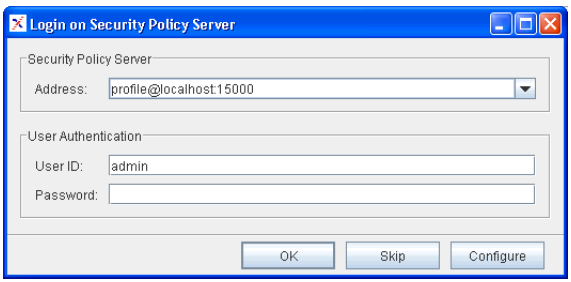


Fig. 12. Login Panel of the Admin Console

SPS Address

The SPS address consists of an optional alias name and an “@” symbol followed by the IP address or hostname, a colon and the port number. For example, you can either enter:

`alias@localhost:5000`, or

192.168.1.90:5000.

The Admin Console remembers SPS addresses from previous logins, these can be chosen from the drop-down box next to the address field. Additional login properties like SSL settings can be configured when pressing the “Configure” button, for details please refer to “Configuring the Connection to the Security Policy Server” on page 111.

DBC admin user

You can use the predefined user admin to login to the Admin Console. This user has the ID admin and the password admin and is authorized to edit the configuration (cf. “Role Properties – Administration” on page 261). You should change the admin user’s password (see “Changing the Admin User’s Password” on page 245 for details).

KEEPING THE DEFAULT PASSWORD IS A SEVERE SECURITY RISK!

Working Offline

Pressing the “Skip” button will start the Admin Console in off-line mode, i.e., not connected to the Security Policy Server. You can connect any time later by choosing the menu item **Server ➜ Login on Security Policy Server**. The Admin Console will quit if you click the ☒ in the upper right corner.

Alternatively, you can start the Admin Console with the option -skip to skip the connection dialog.

Note that when opening or importing a config file or policy data the Admin Console does some sanity checking, i.e., missing braces, unfinished strings, etc. will be recognized. The Admin Console also does some structural checking.

4.4.1 Preferences

Pressing the **Configure** button on the “Login” panel will bring up the “Preferences” panel. Three categories can be edited: “Security Policy Server”, “Write Config”, and “Event Browser”.

Configuring the Connection to the Security Policy Server

Security Policy Server preferences

When starting the Admin Console for the first time you will have to configure the properties for the connection to the SPS. Select the “Security Policy Server” category on the left side of the properties panel.

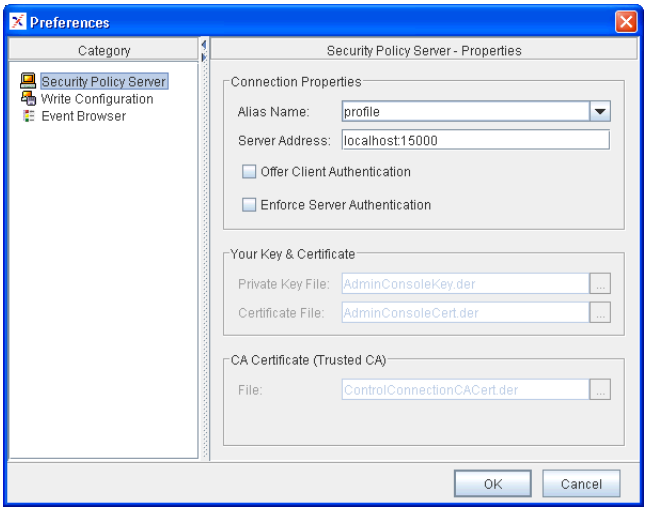


Fig. 13. Login Panel: Server Preferences

Please enter the alias name and the server address in the form `host:port`. You can select an already defined alias from the drop down box. For communicating with the Security Policy Server, SSL is used. Some SSL properties can be configured. If you enable client or server authentication you have to provide the names of the key file and certificates. During the Security Policy Server installation process a proper set of keys and a certificates for the Admin Console will be generated. The installation of these keys is described in section “Installing SSL Keys” on page 101. The default filenames and locations point to these key and certificates.

For testing purposes the “Offer Client Authentication” and the “Enforce Server Authentication” checkbox can be disabled. The Admin Console will then connect to the Security Policy Server without authenticating, but the connection stays encrypted. The user ID and password check will of course be carried out.

Configuring the Event Browser

By default the Admin Console logs audit events from the Security Policy Server as well as local events from the Admin Console. These events are displayed at the bottom of the

Admin Console window (explained in detail in “Audit Event Browser” on page 119). In the “Preferences – Event Browser” panel you can configure the following properties:

- *Audit Local Events*: If this box is checked local events generated by the Admin Console will be displayed in the event browser. Admin Console events have the Category *Local*. They are not written to a log file and only displayed locally in the event browser. A local event is for example *ConfigLoadedFromServer*.
- *Audit Remote Events*: If this box is checked remote events generated by the Proxy and the SPS will be displayed in the event browser.
- *History Buffer Size*: Configure the number of events that will be kept in the buffer (by default 10).
- *Wait msec between 2 Event Fetching Cycles*: Configure the interval between two event fetching cycles (by default 300 msec).
- *Display Table Columns*: Configure which event browser columns will be displayed.

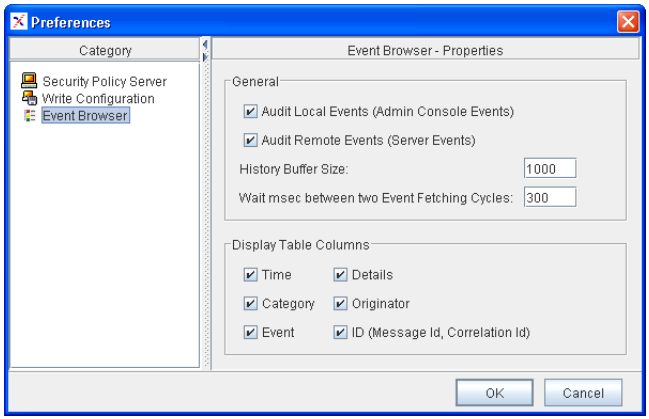


Fig. 14. Event Browser Preferences

Write Configuration – Properties

In the “General” part of this panel you can decide if the configuration and policy data should be written as a whole or incrementally, i.e., only the differences to the Security

Policy Server's current configuration will be saved. Additionally, you can specify a time-out value for write operations to the SPS (default is 300 seconds).

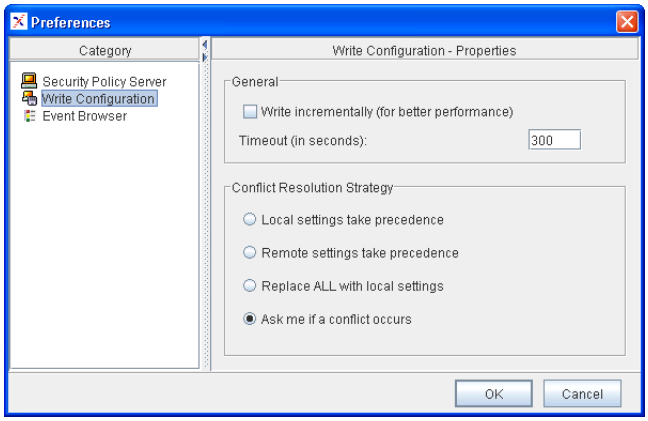


Fig. 15. Write Configuration Preferences

conflict resolution strategy

The DBC supports concurrent administrator access. Therefore the work of two administrators may potentially conflict. Before policy is written to the SPS a check for possible conflicts will be performed and the configured "Conflict Resolution Strategy" will be followed. You can choose between the following:

- "Local settings take precedence": In case of a conflict between a section of the local policy and the policy currently active in the SPS, the conflicting policy section on the SPS will be overwritten with the settings in the local policy.
- "Remote settings take precedence": In case of a conflict between a section of the local policy and the policy currently active in the SPS, the conflicting policy setting in the local policy will be overwritten with the policy section on the SPS.
- "Replace ALL with local settings": The configuration defined with the Admin Console will completely overwrite the configuration on the SPS. Unlike the previous options, this affects the complete policy, not just conflicting sections.
- "Ask me if a conflict occurs": Anytime you try to write a configuration to the SPS and a conflict occurs, a dialog panel will be displayed. You can choose between the options described above.

Passphrase Prompt

Click the OK button to proceed with the login. When using client authentication, the Admin Console prompts for the passphrase that protects the private key file (the passphrase is `xtradyne` by default).

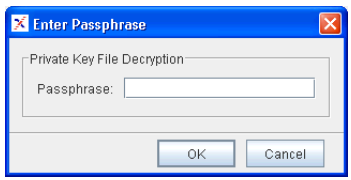


Fig. 16. Passphrase Prompt

After the login on the Security Policy Server is completed, the main window of the Admin Console pops up. The icons in the lower left corner show the connection state. If the plug is plugged into the socket the Admin Console is connected to the SPS. The lock is closed if SSL is used.

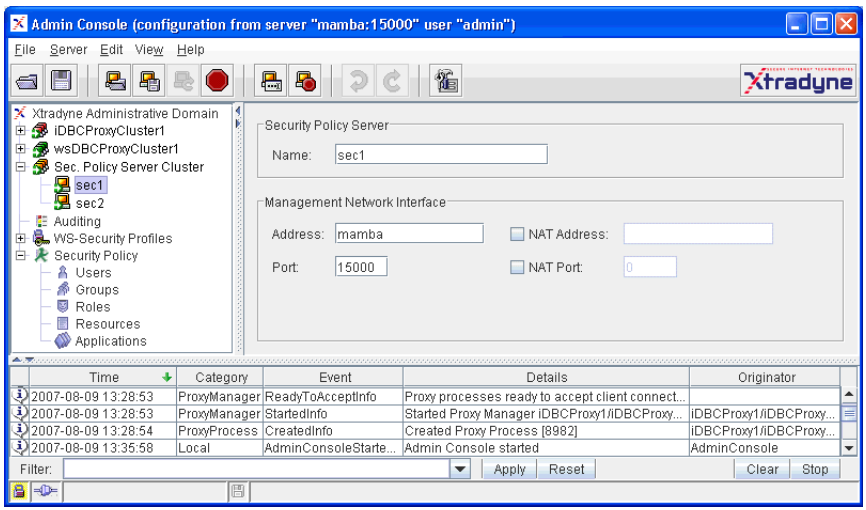


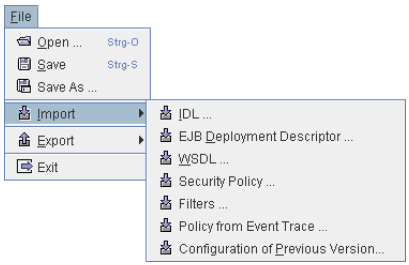
Fig. 17. Screenshot of the Admin Console

4.4.2 General Navigation

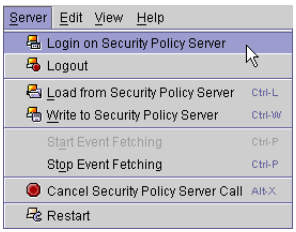
Tool Bar Icons

	Open a configuration from a local file.(**File➜Open**).
	Save the configuration data to a local file (**File➜Save**). This will automatically append the extension `.config`.
	Load the Security Policy Server's current configuration (**Server➜Load from Security Policy Server**).
	Store the configuration data on the Security Policy Server (**Server➜Write To Security Policy Server**).
	Restart the SPS (Cluster) or DBC Proxy (Cluster) with the server's current configuration. The **Restart** function is context sensitive. It will only be active when selecting an SPS (Cluster) or a DBC Proxy (Cluster) in the navigation tree.
	Cancel a call to the Security Policy Server. If the Security Policy Server does not respond, the call can be cancelled before it times out (**Server➜Cancel Security Policy Server Call**).
	Login on Security Policy Server (**Server➜Login on Security Policy Server**).
	Logout from the Security Policy Server (**Server➜Logout**).
	Undo and Redo the last local modification. These functions do not affect the configuration stored on the SPS. The undo stack is unlimited, which means you can undo as many changes as you wish (**Edit➜Undo**, **Edit➜Redo**).
	Switch to expert mode and back (**View➜Switch to Expert Mode**, **View➜Switch to Standard Mode**).

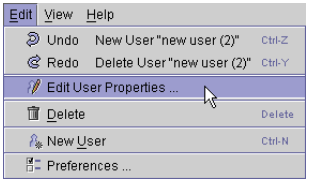
Admin Console Menus



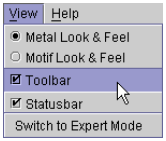
The Admin Console can store and retrieve configuration data from files (**Open ...**, **Save ...**, **Save As ...**). There are several import options for importing security policies (e.g., security policies can be imported from IDL or WSDL files). Security Policies can be exported with the Admin Console in XACML format.



The Server Menu comprises actions performed on the Security Policy Server, like login, logout, or reading and writing of configuration data. You can also cancel a call to the Security Policy Server. The **Restart** function is context sensitive. The currently selected DBC component will be restarted.



The Edit Menu is also context sensitive. Depending on the selected DBC component in the navigation tree the Edit Menu will adapt. This example shows the entries of the Edit Menu when a single user in your security policy is selected. The last item of the Edit Menu is the “Preferences Dialog” where you can set up the Security Policy Server connection, configure the event browser, and the “Write Configuration” settings.



You can change the Java Look & Feel, hide the tool bar and status bar from view and switch from standard mode to expert mode and back.

4.5 General Organization of the Administration Console

When starting the Admin Console and logging on to an SPS, the current configuration is loaded from the Security Policy Server automatically.

standard mode and expert mode

The Admin Console has two modes: the standard mode and the expert mode. By default it runs in the standard mode. In the following we will always use the standard mode of the Admin Console. The expert mode is explained in detail in chapter “Expert Mode” on page 197.

Usually, you will only work in the standard mode. Using the expert mode is not recommended as it is easily possible to render a DBC setup unusable. The expert mode should only be used when advised by technical support.

In the standard mode you see a tree view on the left side. Depending on the selected entry different panels will be shown on the right side. The table below depicts the navigation tree (left side) and explains the main nodes (right side).

I-DBC /WS-DBC Proxy Cluster
These are the properties shared by all Proxies in this cluster.

Security Policy Server Cluster
Basic Properties shared by all Security Policy Servers in the cluster.

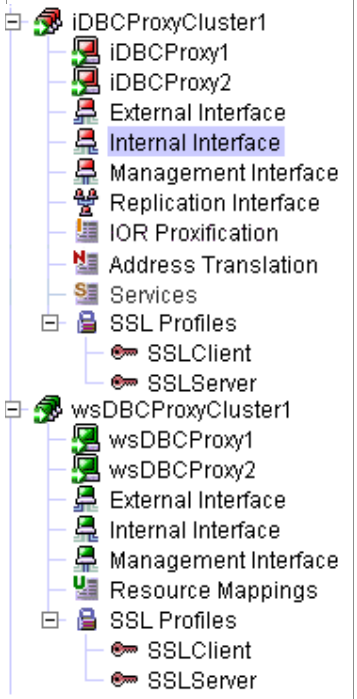
Audit Policy
Activate and deactivate audit events and specify which facility consumes them.

WS-Security Profiles
Define the keys and certificates for signing and verifying XML Digital Signatures (**WS-DBC** only).

Security Policy
Define the Security Policy, i.e., how resources are protected.

Proxies and Security Policy Servers that are up and running are marked with a green arrow in the Admin Console.

When clicking the ⊞ next to the **DBC Proxy Cluster** main node in the navigation tree several subnodes appear. Depending on the type of Proxy (I-DBC or WS-DBC) the available subnodes vary a bit (as depicted below).



DBC Proxy belonging to this cluster
Properties pertaining to **one** DBC Proxy only.

External Interface
Configures the communication endpoints for all connections from and to the public domain.

Internal Interface
Configures the communication endpoints from and to the protected domain.

Management Interface
Configures the interfaces of the Proxy to which the Security Policy Server will connect (selected at install time).

Replication Interface (I-DBC only)
Configures the replication interface (only visible when replication is enabled on the I-DBC Proxy Cluster panel).

Initial IOR Table (I-DBC only)
Configures references to CORBA objects that will be accessible through the I-DBC.

Address Translation (I-DBC only)
Defines address mappings for **outgoing connections** from the I-DBC Proxies to CORBA servers.

Services
Configure services like DBC Monitoring.

Resource Mappings (WS-DBC only)
Define target URLs of Web Services that will be accessible through the WS-DBC.

SSL profiles
SSL profiles for communication endpoints.

4.5.1 Audit Event Browser

The audit event browser is displayed at the bottom of the Admin Console (main) window. Every audit event that is enabled according to policy (see “Audit Policy” on page 189) and is recorded by the Security Policy Server will be listed here. The event browser displays the following columns: *Time* (time stamp), *Category* (denotes the component that triggered the event), *Event* (event name), *Details* (a detailed event description), *Originator*, and an *ID*. Double-click on the event to get more event details listed.

using the filter

A “Filter” text field is provided for convenience at the bottom of the event browser. When entering, for example, “Category = SecuritPolicyServer” in the “Filter” text field, all audited events in the Category *SecurityPolicyServer* will be listed. Wildcards can be used. Concatenation of filter terms is not possible. A history of filter terms is available when pressing the arrow next to the “Filter” text field.

Additional functionality can be reached via Server/Edit menu or via the context menu (right-click into the event browser window to bring up the context menu):

event browser preferences

You can configure event browser properties like the number of displayed events via the menu item **Edit ➜ Preferences**, or via the context menu (choose **Event Browser Preferences...**).

start/stop event fetching

Event fetching can be started and stopped via the menu item **Server ➜ Start Event Fetching**, or **Server ➜ Stop Event Fetching** respectively, or via the context menu.

marker event

Marker events are special events that are not implicitly recorded as part of the regular DBC operation, but triggered explicitly by administrators to add information to the audit stream. Marker events can be used, for example, to mark the start and end of a test run. To trigger a marker event choose **Send Marker Event** from the context menu. An event details text can be given in the provided text field. The *SecurityPolicyServerMarkerInfo* event with the specified event details will show up in the event log and in the event browser. If you would like to use the Marker event, please enable it in the audit policy (see also section “Audit Policy” on page 190).

4.5.2 Activate a Configuration on the DBC Proxy or SPS (Cluster)

There is an asterisk in the title bar and a red floppy disk symbol in the status bar indicating a configuration that needs to be saved. To make changes in the configuration take

effect, choose **Server➔Write To Security Policy Server** (or the corresponding icon in the tool bar).

Conflicts When Writing to the SPS

When writing a configuration to the Security Policy Server, the Admin Console checks for conflicts with concurrent modifications by other administrators. If conflicts are detected and the general conflict resolution strategy is "Ask me if a conflict occurs" (cf. "Write Configuration – Properties" on page 112), the Admin Console displays a dialog panel and you can choose between several options on how to proceed.

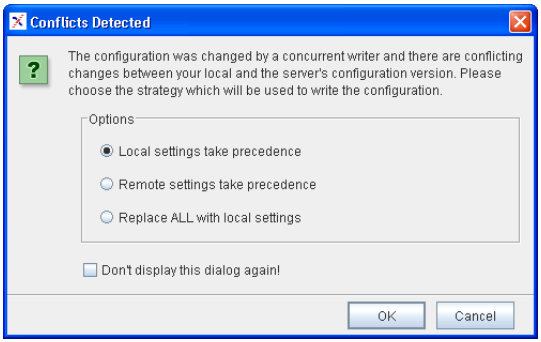


Fig. 18. Conflicts Detected

If you check the box "Don't display this dialog again!" the general conflict resolution strategy will be changed, i.e., the chosen option will always apply whenever a conflict occurs and the dialog box will not be shown again. The options in the panel depicted above can also be changed in the "Preferences – Write Configuration" panel. For a detailed description, please see "Write Configuration – Properties" on page 112.

When to Restart the DBC Proxy / Security Policy Server

A restart of the DBC Proxy or Proxy Cluster, or the Security Policy Server or SPS Cluster is required when changing the addressing properties of the DBC Proxy or the Security Policy Server, or when SSL settings change. This should happen rarely as you would change these settings only when initially configuring the DBC. Operations that are performed frequently, as adding initial IORs or making changes to the security policy, do **not** require a restart. The Admin Console will give a visual prompt when a restart of the

Security Policy Server or the DBC Proxy is required. The table below summarizes when a restart is required.

Restart required when
Changing **port numbers** (on the “External-”, “Internal-” and “Management Network Interface” panel).
Changing **host names** (on the “Network Interfaces” panel).
Changing **SSL settings** (on the “SSL Profiles” and “Security Policy Server” panels).
Adding a trusted certificate to the file `TrustedWSSECCAs.pem` (not configurable with the Admin Console, see “Making the DBC Proxy Trust External Certificates” on page 205).

To restart a DBC Proxy or a DBC Proxy Cluster select the item that shall be restarted and choose **Server➜Restart** from the menu or press the restart button in the tool bar. To restart a Security Policy Server or a Security Policy Server cluster, select the item that shall be restarted and choose **Server➜Restart** or press the restart button in the tool bar.

Note that when restarting an **I-DBC** all current access sessions are lost! This implies that a client has to relogin to access a target service via the I-DBC Proxy (this does not apply to the WS-DBC Proxy).

Note that when performing a re-login the configuration is not loaded automatically from the server. You will be prompted if you want to overwrite the configuration. Thus, you can make changes offline, then login to the Security Policy Server and write the configuration to the server.

CHAPTER

5 DBC Proxy Cluster Configuration

The DBC Architecture supports multiple clusters of DBC Proxies to ensure High Availability and Scalability (see Chapter 1 "High Availability and Scalability" on page 11 of the Deployment Guide). Each DBC Proxy in a cluster shares most of its properties with any other DBC Proxy in the same cluster. (An installation with only a single DBC Proxy can be regarded as a special case of a cluster. In this case, High Availability and Scalability are not supported.)

5.1 DBC Proxy Cluster

Configure the basic settings of the DBC Proxy Cluster, i.e settings that apply to all DBC Proxies in a DBC Proxy Cluster.

Adding and Deleting DBC Proxy Clusters

You can add and delete DBC Proxy Clusters or single DBC Proxies by clicking with the right mouse button on the DBC Proxy (Cluster) in the tree.You can also choose **Edit** from the menu bar and select the type of cluster (WS-DBC or I-DBC) that you want to add.

When there is only one DBC Proxy in a cluster, the cluster view will be hidden. The configuration panels will slightly change: The "DBC Proxy" panel in a cluster configuration corresponds to the "Network Interfaces" panel in a configuration containing only one DBC Proxy. The "External-", "Internal-", and "Management Interface" panels in a cluster configuration can be found below the "Network Interfaces" panel in a configuration containing only one DBC Proxy.

The panels are different depending on the type of proxy. I-DBC Proxy configuration panels are explained in the following sections, WS-DBC configuration panels are explained on page 132.

5.1.1 I-DBC Proxy Cluster - General

The “General” tabbing pane allows you to configure general properties of the I-DBC Proxy Cluster.

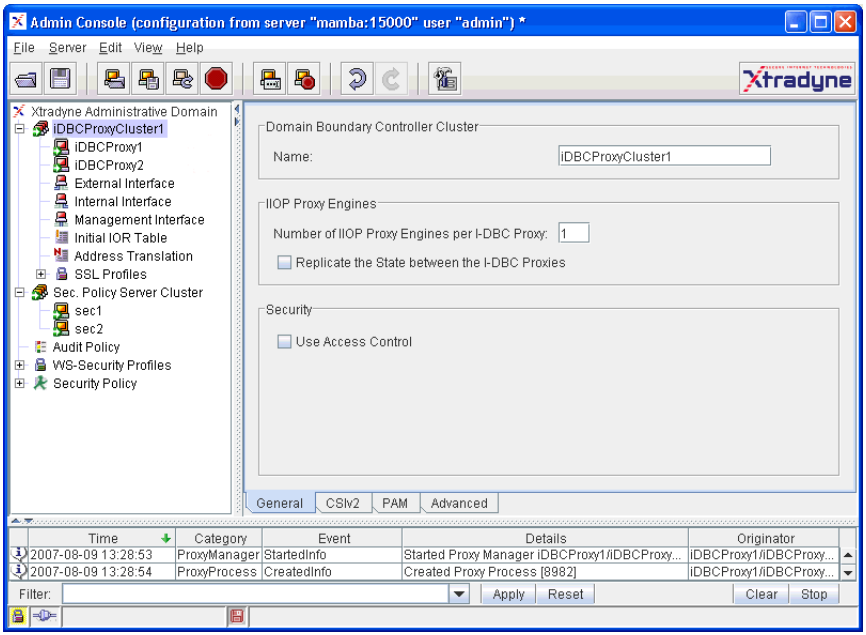


Fig. 19. I-DBC Proxy Cluster - General Properties

I-DBC Cluster Name

I-DBC Proxy Cluster name

A name for the I-DBC Proxy Cluster. These names have to be unique for the I-DBC installation.

IIOP Proxy Engines

Defines general properties pertaining to IIOP Proxy Engines:

- *Number of IIOP Proxy Engines per I-DBC Proxy*: If you don't use replication, not more than a single Proxy Process can be run per host. In this case set the value to "1". With replication, the recommended number of IIOP Proxy Engines per I-DBC Proxy is equal to the number of processors (CPUs) of the host the I-DBC Proxy is running on.
- *Replicate the state between I-DBC Proxies*: If this box is checked, the state of every single I-DBC Proxy in the cluster will be replicated between the other I-DBC Proxies in the cluster. In case of a malfunction of an I-DBC Proxy a stateful failover will be performed, i.e., another I-DBC Proxy will take over. The failover is performed transparently, i.e., clients will not notice anything but a small delay. When this feature is activated, the details of the state replication can be configured on the "Replication Interface" panel. For a detailed discussion on the concepts of replication, please refer to Appendix 2, "Replication" of the Deployment Guide.

Security

use access control

If the "Use Access Control" box is checked access control will be enforced on the I-DBC Proxy according to the Security Policy (cf. part 3, chapter 1, "Security Policies" on page 235).

5.1.2 I-DBC Proxy Cluster – CSIv2

CSIv2 (Common Secure Interoperability Protocol Version 2) is a protocol implementing security features for inter-ORB communication. CSIv2 enables interoperable authentication, delegation, and privileges.

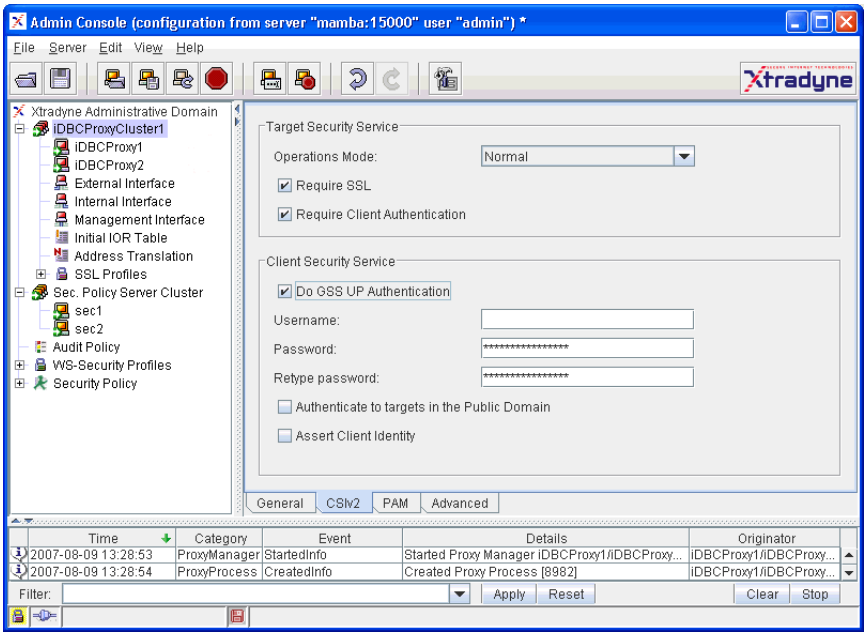


Fig. 20. I-DBC Proxy Cluster - CSIv2

Target Security Service

The Target Security Service settings apply when the DBC is in the server role. The following settings can be configured:

- *Operations Mode*: Defines the CSIv2 operation mode. One of the following modes can be chosen:
  - *Normal*: In the “Normal” mode the DBC acts as an endpoint, i.e., the DBC authenticates the user, verifies the identity token, and grants or denies access according to the configured security policy.
  - *Transparent*: In the “Transparent” mode the DBC authenticates the user, verifies the identity token, performs access control according to the configured

security policy and then passes on the service context to the “real” server transparently.

- *Off (Filter Service Context)*: The DBC will not process CSIv2 service contexts. CSIv2 service contexts will be dropped unless a pass through option for CSIv2 service contexts is defined on the “IOR Proxification” page, “Proxification Options” tab.

- *Require SSL*: When checking this box the client has to use SSL.
- *Require Client Authentication*: When this box is checked, the client has to authenticate to the I-DBC.

Note that when configuring users in the DBC’s security policy, users can be allowed to assert other identities. For details on how to configure this, please refer to “CSIv2 (I-DBC Proxy only)” on page 243.

Client Security Service

The Client Security Service settings apply when the DBC is in the client role.The following settings can be configured:

- *Do GSS UP Authentication*: GSS UP stands for “General Security Service Username Password”. When checking this box Username/Password Authentication will be done. Enter the user name and the password in the corresponding fields.
  - *Authenticate to Targets in the Public Domain*: When GSS UP Authentication is selected, you can additionally configure whether the I-DBC Proxy shall be allowed to send GSS UP credentials to targets in the public domain.
- *Assert Client Identity*: When checking this box client identities will be asserted, i.e., an identity token stating the client’s identity is added to the service context.

5.1.3 I-DBC Proxy Cluster - PAM

On the “PAM” tabbing pane you can configure Pluggable Authentication Modules. Activate this feature when you want to use an external authentication mechanism to authenticate users connecting to the DBC Proxy. All users authenticated via PAM will be mapped to one DBC user. A security policy for this user can then be defined (please refer to chapter “Security Policies” on page 235 on how to do that). An advantage of activating PAM is that you can use your existing user database instead of having to redefine all potential DBC users in the DBC’s security policy.

If the DBC shall use PAM check the “Use PAM for Authentication” box. The following values are can be configured:

- **Service Name**: the PAM service name, i.e., the name of the PAM configuration. Under Linux this is the name of the PAM configuration file which is stored under `/etc/pam.d`, under Solaris this is the name of the PAM configuration entry in the file `/etc/pam.config`. For a more detailed description, please refer to your local PAM documentation.
- **DBC User ID**: the DBC user any user authenticated via PAM is mapped to. The user ID has to correspond to one of the user IDs defined in the Security Policy (cf. “Users” on page 241).
- **Result Cache Lifetime**: the number of seconds a positive authentication result will be cached by the DBC.
- **Perform multiple authentications per PAM transaction**: Check this box if you want to perform multiple authentications per PAM transaction.

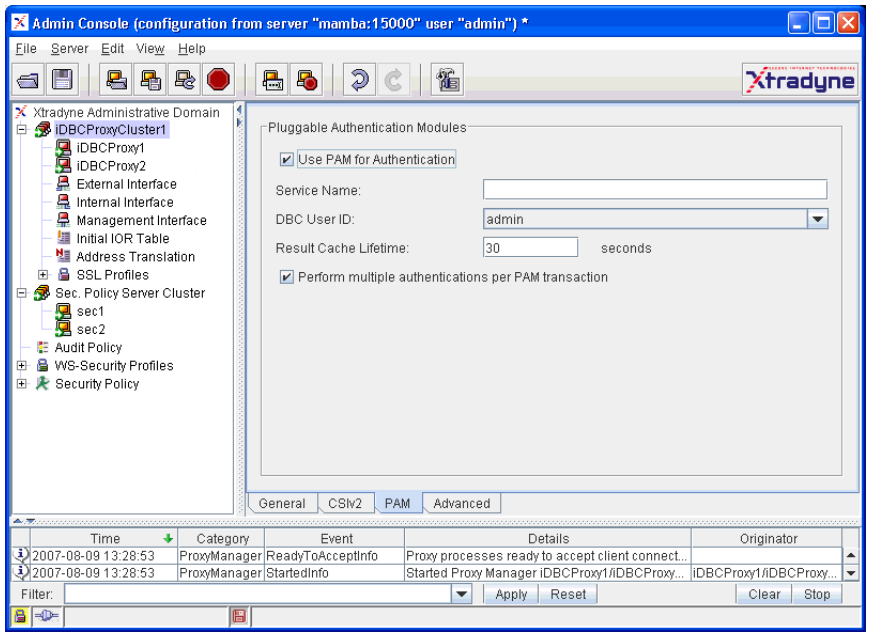


Fig. 21. I-DBC Proxy Cluster - Pluggable Authentication Modules

5.1.4 I-DBC Proxy Cluster - Advanced

The “Advanced” tabbing pane configures advanced properties of the I-DBC Proxy Cluster, like the Access Session Management or GIOP Connection timeouts. (On your first read you may safely skip this section).

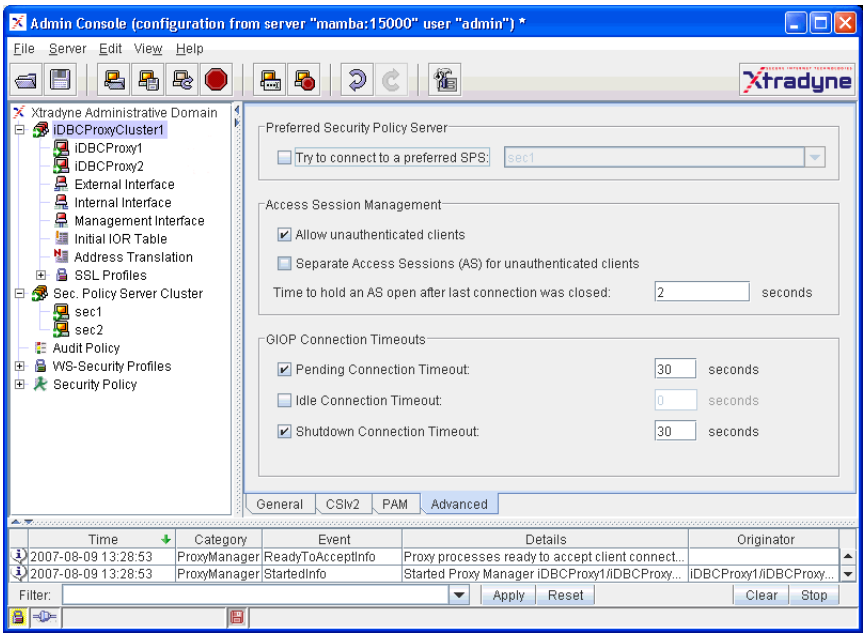


Fig. 22. I-DBC Proxy Cluster - Advanced Properties

Preferred Security Policy Server

Select a Security Policy Server from the drop-down menu. The Proxy will try to connect to this SPS. If the selected SPS cannot be contacted the Proxy will try to reach one of the other Security Policy Servers in the Cluster.

Access Session Management

An access session is established when a client first connects to the I-DBC. It includes all objects (IORs) accessed by the client during that session. Configurable properties of access sessions are:

- *Allow unauthenticated clients*: allows access session creation for unknown clients. This might be useful when testing the I-DBC, for example, with callbacks. For more details on configuring callbacks, please see “Callback Support” on page 281.
- *Separate Access Sessions (AS) for unknown clients*: only available if the “Allow unknown Clients” box is checked. If this box is checked every unknown client connecting from a different host will be put into a separate access session. For more details on the I-DBC’s access session concept, please refer to “Number of Access Sessions and Access Session Hierarchy” on page 247.
- *Time to hold an AS open after last connection was closed*: You can give the delay (in seconds) until an access session is really closed. Configuring such a delay makes sense if you have a client that closes the connection, subsequently connects to the I-DBC Proxy again and should use the same access session as before (this might be the case when clients use a naming service for bootstrapping).

Setting GIOP Connection Timeouts

The GIOP connection timeouts described in this section can be modified to adapt the I-DBC’s connection management to specific environments. They should only be modified if required and can be left untouched otherwise. On your first reading you may safely skip this section.

setting GIOP connection timeouts

There are three optional GIOP Connection timeout values which can be set to modify the connection management policy of the I-DBC Proxy Cluster. Each timeout value is enabled by activating the check box left to the timout label. The timeout value is given in seconds and entered into the text field right to the timeout label. If the check box is deactivated this timeout value disabled. The following timeout values may be set:

pending connection timeout

- *Pending Connection Timeout:* A pending connection is an accepted TCP connection on which no CORBA message has been received yet. The timeout constrains the time until an I-DBC Proxy in the cluster closes the connection if no CORBA message has been received. The recommended setting is 30 seconds.

idle connection timeout

- *Idle Connection Timeout:* A connection is idle if no GIOP messages are currently being received or sent on this connection. The timeout constrains the period after which an I-DBC Proxy will close an idle connection. It is recommended to disable

the timeout. For a more detailed description, see also chapter “Troubleshooting”, section “Firewall Configuration – TCP Connection Timeouts” on page 227.

- *Shutdown Connection Timeout:* Before closing a connection an I-DBC Proxy indicates its intent to close a connection by sending a GIOP CloseConnection message to the client or server, which indicates that the connection is no longer used. The timeout value constrains the time period after which an I-DBC Proxy will shutdown the connection regardless of the client’s or server’s behavior and the recommended setting is 30 seconds.

shutdown connection timeout

5.2 WS-DBC Proxy Cluster

The “WS-DBC Proxy Cluster” pane allows you to configure the basic settings of the WS-DBC Proxy Cluster, i.e., settings that apply to all WS-DBC Proxies in a WS-DBC

Proxy Cluster. Note that you first have to add a WS-DBC Proxy to switch to the “Cluster View” (see below).

5.2.1 WS-DBC Proxy Cluster – General

The “General” tabbing pane allows you to configure general properties of the SOAP Proxy Engines.

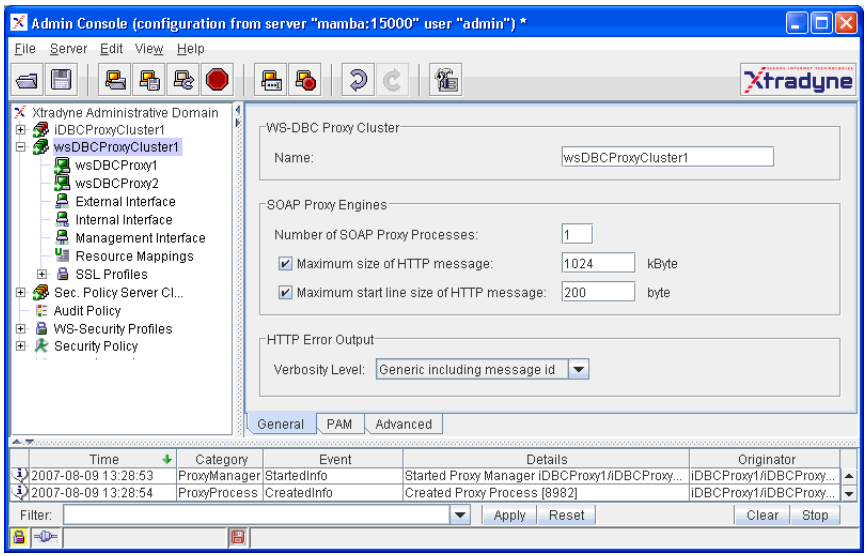


Fig. 23. WS-DBC Proxy Cluster Properties

WS-DBC Proxy Cluster Name

Name the WS-DBC Cluster. Names have to be unique for the WS-DBC installation.

Number of SOAP Proxy Engines

Each SOAP Proxy Engine (or Proxy Process) is single threaded and can only take advantage of a single CPU at a time; to achieve the highest performance on a multiple processor machine, the number of SOAP Proxy Processes should match the number of available CPUs on the WS-DBC Proxy host.

Maximum Size of an HTTP Message

To prevent Denial of Service attacks the maximum size of an HTTP message can be defined. The recommended setting is to leave the default value (1024 kByte).

Maximum Start line of an HTTP Message

To prevent Denial of Service attacks the maximum size of the “HTTP First/Start Line” can be limited. The recommended setting is to leave the default value (200 Bytes).

Verbosity Levels on HTTP/SOAP Errors

Define the verbosity level of error messages that are returned to the client by the WS-DBC Proxy. Three levels are available:

- *No output at all*: An HTTP/SOAP error containing no information about the error that occured.
- *Generic including message id*: The WS-DBC Proxy generates a generic error message. For a list of generic HTTP/SOAP error messages, please refer to Appendix B, “Error Messages and System Exceptions”.
- *Detailed*: An HTTP/SOAP error containing a detailed description of the error reason (resembling the messages in the log file).

Verbosity Levels can be defined for a resource or for a WS-DBC Proxy. Verbosity levels for a resource (on the “Resources – Outgoing Policy” panel) take precedence over verbosity levels defined for the WS-DBC Proxy (on this panel).

5.2.2 WS-DBC Proxy Cluster – PAM

On the “PAM” tabbing pane you can configure Pluggable Authentication Modules. For details, please refer to the description of the I-DBC Proxy Cluster’s PAM panel on page 127.

5.2.3 WS-DBC Proxy Cluster – Advanced

SSL accelerator hardware

On the “Advanced” tabbing pane of the “WS-DBC Proxy Cluster” pane you can configure SSL accelerator hardware that may be used on the cluster hosts to increase performance with respect to SSL encryption. If accelerator hardware is employed, choose the

appropriate accelerator type from the drop down menu. Select “none” if no accelerator hardware is used.

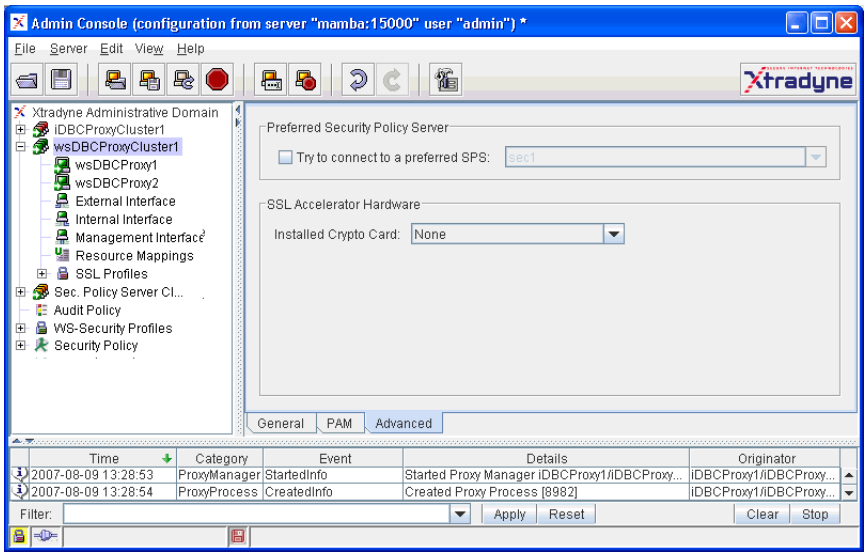


Fig. 24. WS-DBC Proxy Cluster panel: Advanced tab

Preferred Security Policy Server

Select a Security Policy Server from the drop-down menu. The Proxy will try to connect to this SPS. If the selected SPS cannot be contacted the Proxy will try to reach one of the other Security Policy Servers in the Cluster.

5.3 DBC Proxy Configuration

The “DBC Proxy Properties” pane (see screenshot below) allows you to configure the basic settings of a single DBC Proxy. These setting are not shared with other DBC Proxies in a cluster.

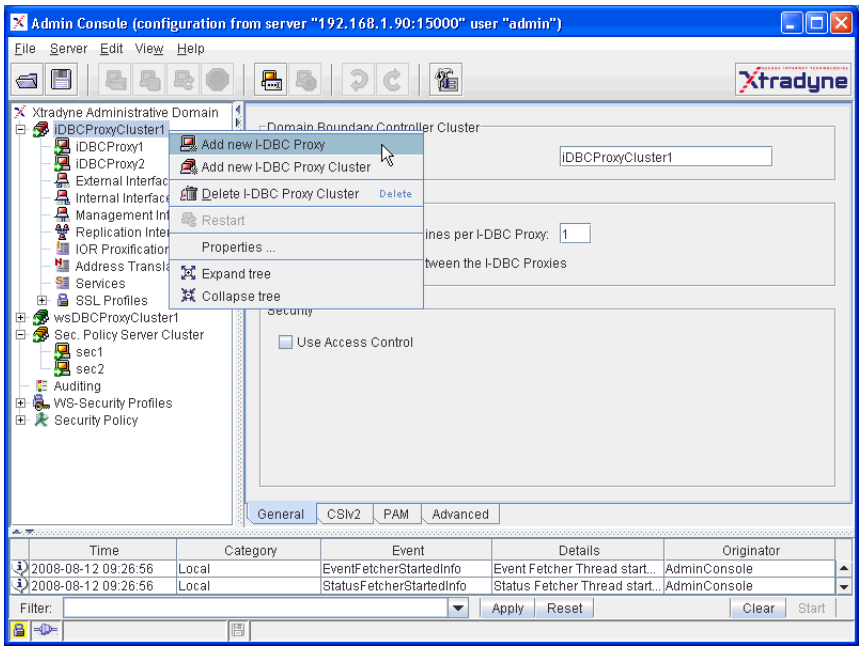


Fig. 25. DBC Proxy Properties

Adding and Deleting DBC Proxies in the Configuration

You can add and delete single DBC Proxies by clicking with the right mouse button on the DBC Proxy (Cluster) icon on the left side of the panel and selecting from the context menu. You can also choose **Edit➜Add New DBC Proxy** and **Edit➜Delete DBC Proxy** from the menu bar.

In a configuration containing only one DBC Proxy the cluster view will be hidden. All the properties that can be configured in the “DBC Proxy” pane in a cluster configuration are available on the “Network Interfaces” pane in a configuration containing only one DBC Proxy.

5.3.1 DBC Proxy

In the upper part of the “DBC Proxy” panel you define the name of the DBC Proxy. Names have to be unique for each DBC installation.

5.3.2 DBC Proxy Network Interfaces

The DBC Proxy can manage between one and four physical network interfaces for its four logical interfaces. Two network interfaces are employed to separate the protected network from the external network. A third network interface can be dedicated to establish the control connections to the Security Policy Server Cluster. The fourth interface – the replication interface – applies to the I-DBC only. This interface is used to replicate the access session state between I-DBC Proxies to enable stateful failover.

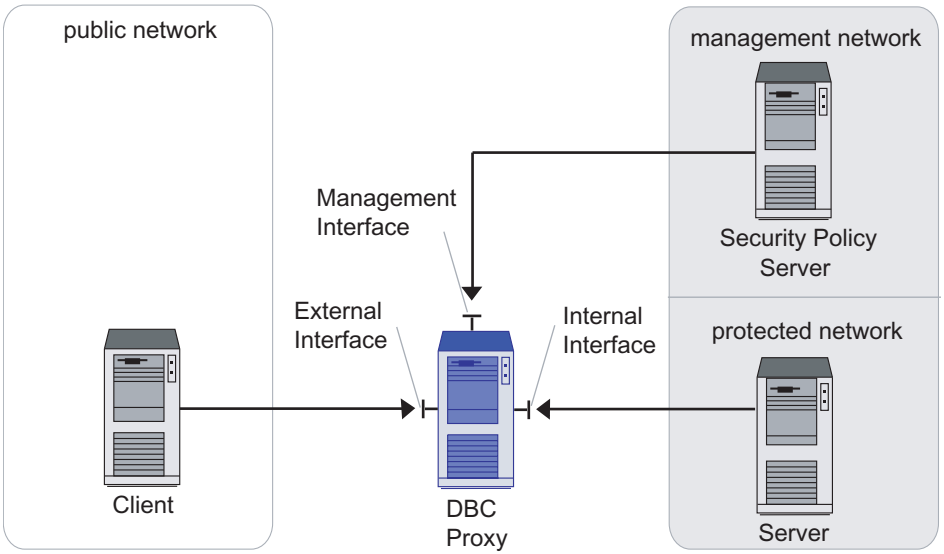


Fig. 26. DBC Proxy Network Interfaces

If you intend to set up more than one network interface you need to specify the logical interfaces each network interface serves:

- *External Interface:* The interface to the external network (also referred to as public network).
- *Internal Interface:* The interface to the protected network.
- *Management Interface:* The interface to the network where the Security Policy Server Cluster is located.
- *Replication Interface* (I-DBC only): The interface used by the I-DBC Proxies in a cluster to exchange their state. This interface can only be assigned when checking the box “Replicate the state between I-DBC Proxies” on the “I-DBC Proxy Cluster” panel. Note that a NAT address cannot be configured for this interface.

The following table presents reasonable assignments of logical interfaces (External, Internal, Management, and Replication Interface) to either one, two, three or four different physical interfaces.

Table 3. Assignment of network interfaces to logical interfaces

		Logical Interfaces			
		External Interface	**Internal Interface**	**Mgmt. Interface**	**Replication Interface**
Physical Interfaces	Interface A	✔	✔	✔	✔
	Interface A	✔			
	Interface B		✔	✔	✔
	Interface A	✔	✔		
	Interface B			✔	✔
	Interface A	✔			
	Interface B		✔		
	Interface C			✔	✔
	Interface A	✔			
	Interface B		✔		
	Interface C			✔	
	Interface D				✔

The table “DBC Proxy Network Interfaces” on the “DBC Proxy Properties” pane contains one row for each interface (please see also screenshot “DBC Proxy Properties” on

page 135). Each row contains one text field for the IP-Address or host name of the network interface (as seen from the potential communication peers). Another text field is provided for the NAT Address (Network Address Translation). NAT addresses for DBC network interfaces are explained in detail in the next section.

When using host names instead of IP addresses, be sure that proper name resolution is available.

assigning logical interfaces to network interfaces

The right-hand side of the panel contains three radio button groups. These buttons are used to assign the logical interfaces to the network interface. To assign a network interface, first choose the logical interface you wish to assign. Then enter the appropriate IP address or host name into the text field.

The properties of the External, Internal, and Management Interface can be configured on the corresponding panes. Settings for these interfaces apply to the whole DBC Proxy Cluster and are explained in the following sections.

5.3.3 NAT Addresses for I-DBC Proxy Interfaces

If NAT or some other address translating device is employed at the network boundary, then the address of the DBC Proxy host is not visible from other networks. A different, external address must be mapped to the DBC Proxy host address, in this case and the DBC Proxy can only be reached via this translated address. In a DBC setup a NAT device could be employed:

- at the boundary to the public network between clients and the DBC Proxy host,
- at the boundary to the protected network between servers and the DBC Proxy host,
- at the boundary to the management network between DBC Proxy and SPS host.

Figure 27 illustrates these options. Note that NAT address can not be configured for the replication interface.

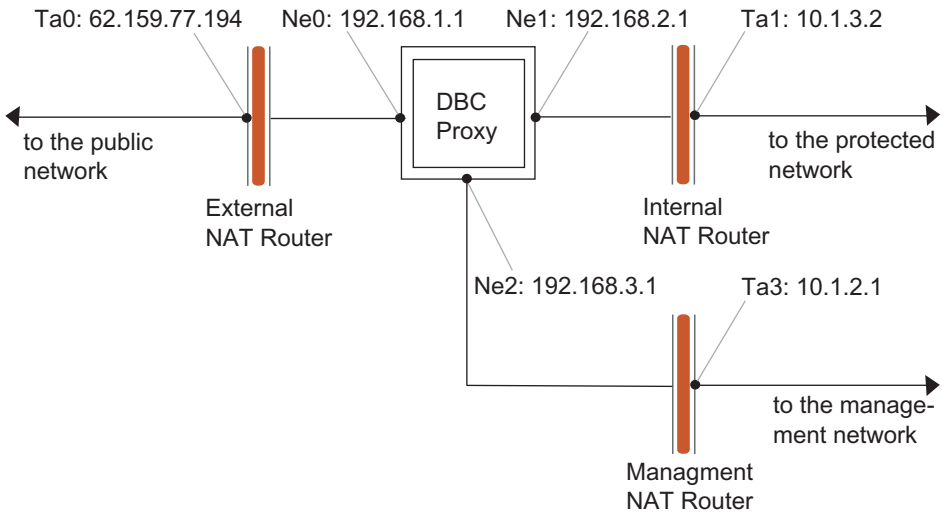


Fig. 27. DBC Proxy Network Interfaces

The NAT mappings in the scenario depicted in figure 27 imply the following:

- clients in the public network see the external address Ne0:192.168.1.1 as the translated address ta0:62.159.77.194.
- servers in the protected network see the internal address Ne1:192.168.3.1 as the translated address ta1:10.1.3.1 (internal NAT only makes sense for I-DBC installations).
- The Security Policy Server sees the management address Ne2:192.168.2.1 as the translated address ta2:10.1.2.1.

Check the NAT box next to the interface for which you want to configure a NAT address. Provide the translated address in the text field in the NAT address column of the

panel. The screenshot below shows a configuration for the scenario depicted in figure 28 with the Admin Console.

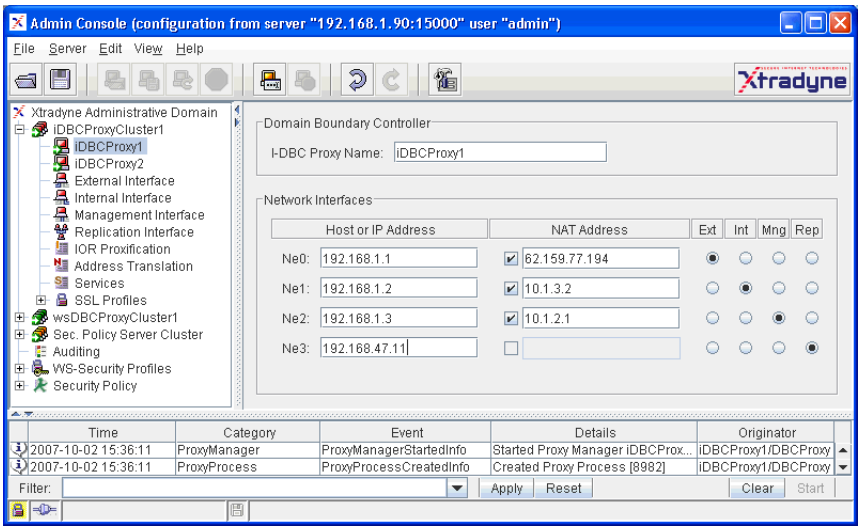


Fig. 28. Example for DBC Proxy network interfaces

Note that the NAT Address for every DBC Proxy is the same if a traffic redirector is used. In this case, you can also configure the Virtual Address on the External/Internal pane which applies to the whole DBC Proxy Cluster (for details see section “Virtual Address” on page 142).

5.4 External and Internal Interface Overview

This section gives a conceptual overview of communication endpoints used by servers and clients in the public and protected domain to contact a DBC Proxy. These communication endpoints can be configured on the “External Interface” and “Internal Interface” panes. When operating only one DBC Proxy these panels can be found below the “Network Interfaces” node.

communication endpoints

The DBC Proxy sets up communication endpoints to accept incoming connections and to establish outgoing connections. Details affecting the communication with the **public**

domain can be configured in the “External Interface” pane. There are two types of external communication endpoints:

- External Acceptors: Clients in the public domain connect to these acceptors when contacting the DBC Proxies.
- External Connectors: External Connectors are used by the DBC Proxies to establish connections to the public domain.

The same two types of internal communication endpoints exist for the communication with the **protected domain**. Details concerning these communication endpoints can be configured in the “Internal Interface” pane:

- Internal Acceptors: Clients in the protected domain connect to these acceptors when contacting the DBC Proxies to communicate with external servers.
- Internal Connectors: Internal Connectors are used by the DBC Proxies to establish connections to the protected domain.

Typically, in the **External Interface** pane both **Acceptors** are enabled while in the **Internal Interface** pane both **Connectors** are enabled.

Note that you must set up at least one Acceptor on one interface panel so that a DBC Proxy can be contacted, and one Connector on the other interface panel so that a DBC Proxy can initiate connections to the server.

Internal and External Listeners have the same properties, the panes are analogous. Therefore the following section only explains the settings for External Listeners in detail.

5.5 External Interface

Inbound and outbound connections **from and to the public domain** are configured on the “External Interface” panel. The screenshot below depicts the External Interface

panel of the I-DBC. The WS-DBC External Interface panel is almost identical (the protocol is HTTP instead of IIOP).

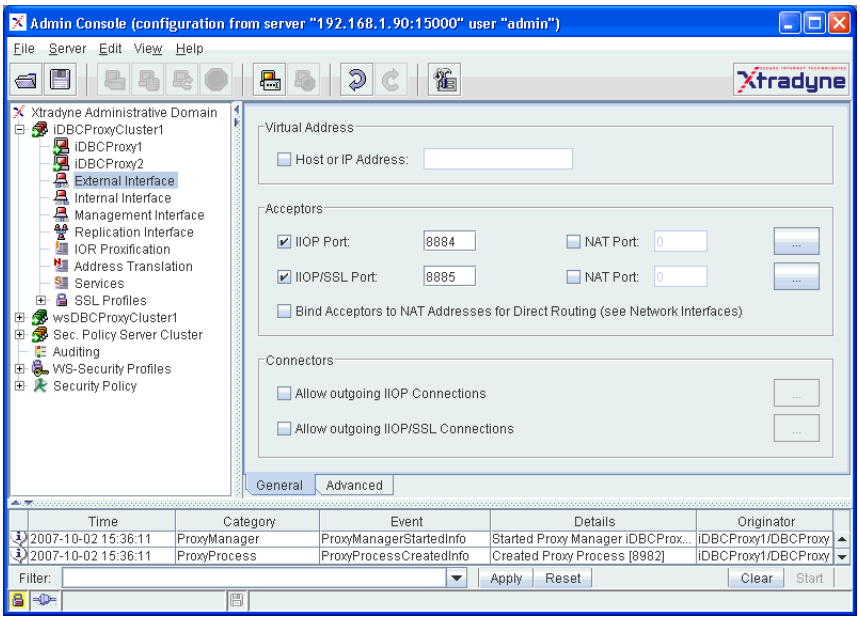


Fig. 29. External Interface - General

The following can be configured:

- The Virtual Address of the DBC Proxy in the cluster, cf. “Virtual Address” on page 142 (not available when configuring a single DBC Proxy).
- Acceptors for incoming connections from the public domain to the DBC Proxies.
- Connectors for outgoing connections from the DBC Proxies to the public domain.

For **I-DBC** Acceptors you can define a NAT port mapping on the right side of the panel. We will explain the panel without NAT from top to bottom. For a detailed description on NAT ports see section “NAT Port Mappings for the I-DBC Proxy” on page 145. The WS-DBC doesn’t need to know about NAT on that level.

5.5.1 Virtual Address

Note that if only one DBC Proxy is configured the virtual address is not configurable.

If you operate several DBC Proxies in a cluster, a traffic redirector will typically be employed to distribute the traffic amongst the DBC Proxies. Clients from the public network that wish to contact the DBC have to use the address of the traffic redirector - the virtual address - instead of the DBC Proxy's External Interface address (see figure 30, "External Interfaces and Virtual Address of DBC Proxies in a cluster"). The virtual address applies to every DBC Proxy in a cluster. Every DBC Proxy Cluster can of course have a different virtual address. Also the virtual address for the external and internal interface can be different.

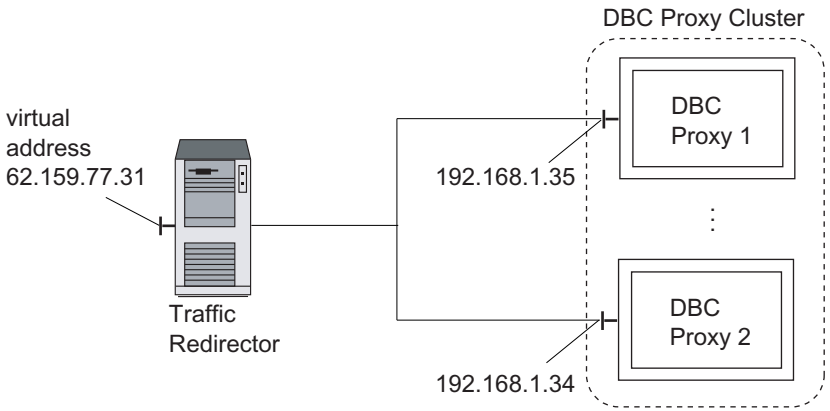


Fig. 30. External Interfaces and Virtual Address of DBC Proxies in a cluster

Note that the virtual address can also be emulated by configuring NAT addresses for the respective interface of each DBC Proxy in a cluster. NAT addresses can be configured in the "DBC Proxy Properties" pane. E.g., to emulate the external virtual address the same NAT address for the External Interface would have to be entered for every DBC Proxy.

5.5.2 Acceptors

There are two types of Acceptors for incoming connections:

- **I-DBC**:
  - *IIOP Acceptor*: Accepts incoming connections for IIOP over plain TCP.
  - *IIOP/SSL Acceptor*: Accepts incoming connections for IIOP over SSL.
- **WS-DBC**:
  - *HTTP Acceptor*: Accepts incoming connections for HTTP over plain TCP.
  - *HTTPS Acceptor*: Accepts incoming connections for HTTP over SSL.

For each acceptor a port number can be configured at which the DBC Proxy will accept incoming connections. As the DBC Proxy does not run with root privileges, no privileged ports can be assigned, i.e., the configured port number must be greater than 1023.

Acceptor Details

The “IIOP Acceptor Details” or “HTTP Acceptor Details” dialog provides additional configuration settings for the TCP transport. To open the dialog click the “...” button next to the Acceptor Port.

Fig. 31. Acceptor Details: TCP Options

The following configuration options are provided:

TCP Read Buffers

- *Read Buffer*: TCP Read Buffers are used to reduce the number of network read operations, thus increasing performance. The default buffer size is 1024 Byte (this value is also assumed when deactivating the check box). The TCP read buffer size should only be changed by experienced system administrators to adapt the DBC Proxies to specific networking environments. Normally, there is no need to change the default value.

send “KeepAlive” messages

- *Send “Keep Alive” messages:* Activates the TCP keep alive mechanism for connections. This mechanism is used to keep intermediate firewalls from disconnecting idle transport connections. Some firewalls have a timeout for idle connections and close them when the timer expires. Activating the “Keep Alive” option prevents this, provided the firewall timeout is longer than the TCP keep alive timeout of your operating system. Depending on the operating system, a keep alive message is sent on a connection that has been idle for some amount of time, e.g., two or three hours. This interval is a global setting in your system and cannot be configured with the Admin Console. To change the interval, please consult your operating system manual or refer to chapter “Troubleshooting”, section “Firewall Configuration – TCP Connection Timeouts” on page 227.

SSL Acceptor Details

The “IIOP/SSL Acceptor Details” or “HTTPS Acceptor Details” dialog provides additional configuration settings. You can specify TCP transport options and configure SSL settings. To open the dialog click the “...” button next to the SSL Acceptor port. The TCP options are the same as in the “Acceptor Details” pane (see previous section).

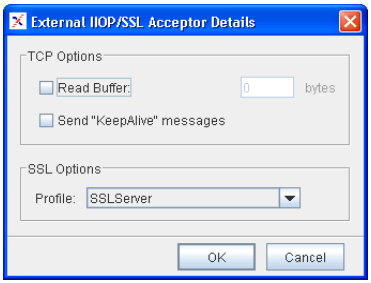


Fig. 32. SSL Acceptor Details

In the “SSL options” part an SSL profile can be chosen. By default External and Internal Interface share the same SSL Profile. The predefined profile “SSLServer” is used for Acceptors and the predefined profile “SSLClient” is used for Connectors. You can define your own SSL profiles on the “SSL Profiles” panel. For further information please refer to “SSL and WS-Security Profiles” on page 165.

5.5.3 NAT Port Mappings for the I-DBC Proxy

As discussed in section “NAT Addresses for I-DBC Proxy Interfaces” on page 138 the I-DBC can be configured to work with address translating devices employed at the network boundary. If these devices do port mappings, the translated port has to be configured in the I-DBC Proxy. This can be done by checking the NAT box and entering the translated port into the text field on the right side of the “Acceptors” part on the “External Interface” panel.

Defining Port Mappings for the I-DBC Proxy Cluster

Defining port mappings for a cluster of I-DBC Proxies can be useful if you operate more than one cluster together with a traffic redirector and want to distinguish the clusters.

In this case, port mappings can be configured via the “Virtual Address” field on the “Interfaces” panel. Note that the virtual address applies to all Proxies in a cluster, i.e., all I-DBC Proxies in a cluster share the same external NAT Address.

Bind Acceptors to NAT Addresses for Direct Routing.

Special configuration settings are required if you operate the I-DBC Proxy in a cluster using a traffic redirector like LVS (Linux Virtual Server) or Cisco Local Director in direct routing mode and your application uses callbacks. The setting “Bind Acceptors to NAT Addresses for Direct Routing” must be checked in either “External Interface”, “Internal Interface” or both, depending on which side of the I-DBC Proxy the traffic redirector is located. Additionally you must supply the virtual IP address (vip) of the cluster on the “I-DBC Proxy” panel(s) in the “NAT Address” field of the appropriate interface(s), otherwise the I-DBC Proxies will not start.

If “Bind Acceptors to NAT Addresses for Direct Routing” is checked, the I-DBC Proxy’s listeners will be bound to the address given in the NAT part of the “I-DBC Proxy Network Interfaces” panel instead of the address in the field “Host or IP Address”. Outgoing connections, as usual, will be initiated from the address given in the field “Host or IP Address”. This separation is necessary as outgoing connections in a direct routing cluster must always come from the real IP of the respective I-DBC Proxy, whereas the listeners must be bound to the virtual IP address of the cluster.

5.5.4 Connectors

In this section of the “External Interface” panel you can enable outgoing connections. There are two types of Connectors for outgoing connections:

- **I-DBC**:
  - *IIOP Connector*: Sets up outgoing connections for IIOP over plain TCP.
  - *IIOP/SSL Connector*: Sets up outgoing connections for IIOP over SSL.
- **WS-DBC**:
  - *HTTP Port*: Sets up outgoing connections for HTTP over plain TCP.
  - *HTTPS Port*: Sets up outgoing connections for HTTP over SSL.

Additionally it is possible to specify the port number at which the DBC Proxy sets up an outgoing connection for every activated Connector. As the DBC Proxy does not run with root privileges, it is not possible to assign a privileged port, i.e., the assigned port number must be greater than 1023.

Each type of Connector can be enabled or disabled. If both Connectors are disabled the DBC Proxy will not be able to establish outgoing connections. If a single Connector is enabled the DBC Proxy will only create a connection of the respective Connector type. For example, if only the SSL Connector is activated the DBC Proxy will not be able to create connections using plain TCP.

Connectors have the same detail properties as Acceptors. For the configuration of Connector Details, please refer to the previous section.

5.5.5 External I-DBC Interface - Advanced

The **I-DBC** External Interface panel additionally offers an “Advanced” tabbing pane where properties for callback support can be configured,

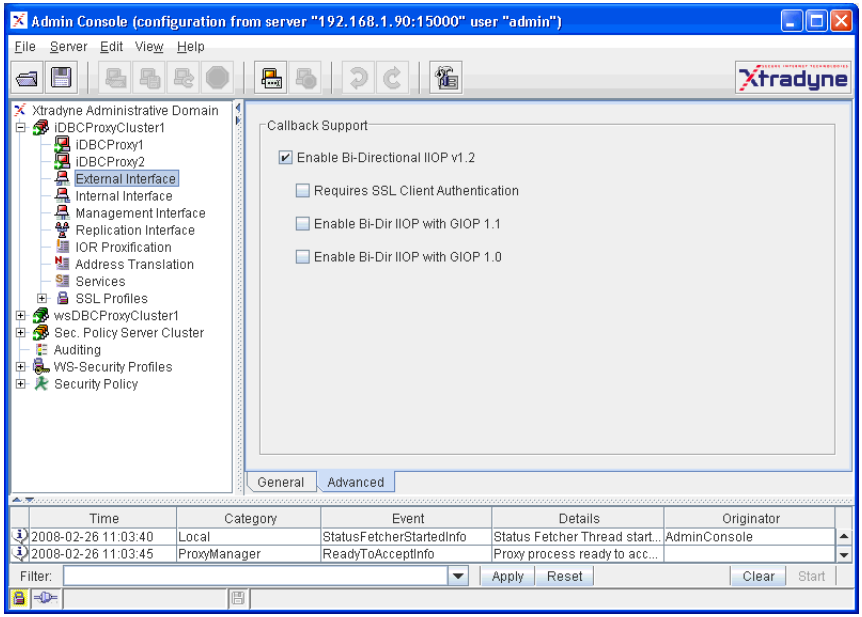


Fig. 33. External Interface - Advanced

callback support

If the first box is checked bidirectional IIOP will be used for connections between I-DBC Proxies (i.e., when operating a cascaded I-DBC Proxy). Additionally, you can configure if SSL client authentication shall be required and if GIOP version 1.1 and/or version 1.0 shall be supported.

For more details on configuring permissions for callbacks, refer to “Configuring Permissions for Callbacks” on page 282.

5.6 Internal Interface

Internal acceptors and connectors apply to communication between the DBC Proxy and the protected domain. On the “Internal Interface” pane you can specify the endpoints for inbound and outbound connections initiated by clients or servers **in the protected domain**. Internal and external acceptors have the same properties. The panes are analogous. Please refer to the previous section on configuring the internal interface.

5.7 Management Interface

The Management Interface is used by the Security Policy Server to connect to the DBC Proxy. On the “Management Interface” pane you define the following:

- *Port*: Choose a port number (default is 14000). If you don’t want to use the default port 14000, please read the gray box below.
- *NAT Port*: If NAT with port mapping is used between the Security Policy Server host and the DBC Proxy host, check the “NAT Port” box and enter the translated port. See also “NAT between the DBC Proxy and the SPS” on page 187.

port and NAT port for the Management Interface

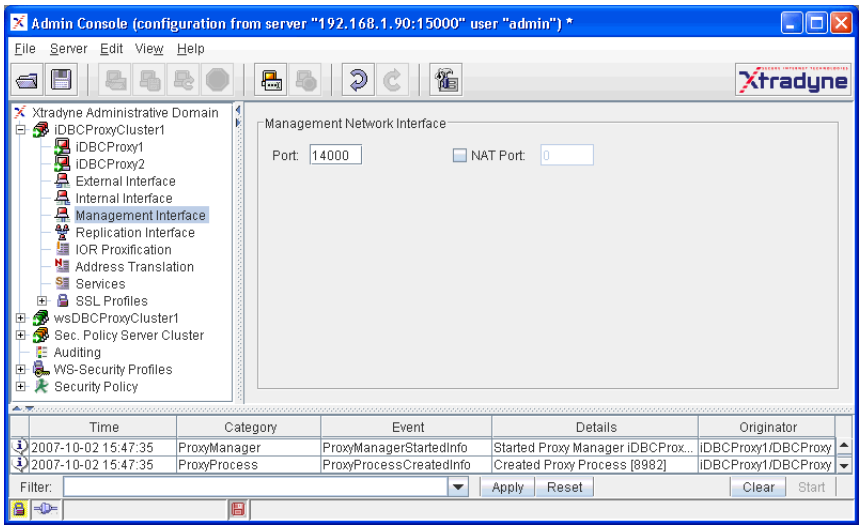


Fig. 34. Management Network Interface

5.8 Replication Interface

Replication is a feature of the **I-DBC** enterprise edition. The replication interface is only available when activating state replication on the “I-DBC Proxy Cluster” panel. For a detailed explanation, please refer to Appendix 2, “Replication”, page 27 of the Deployment Guide.

5.9 Resource Mappings

This panel applies to the **WS-DBC** only. When configuring an **I-DBC** please proceed to section “IOR Proxification” on page 154.

The resource mappings table defines the virtual names that clients use to contact Web Services through the WS-DBC. More precisely, the names given here are the URL paths that are appended to the WS-DBC Proxy address to yield a service URL.

Clients are provided with a URL of the form:

`http(s)://proxyhost.proxyport/path`.

When a request arrives at the WS-DBC, it uses this table and the target information contained in the HTTP header to map the request target to the actual service.

Note that when a WSDL file is available for the Web Services that shall be made accessible through the DBC, the exposure wizard can be used. This wizard also offers a panel to define a resource mapping. It can be reached via the menu: **File ➜ Import resource from WSDL** and is explained in detail in section “WSDL Exposure Wizards” on page 264.

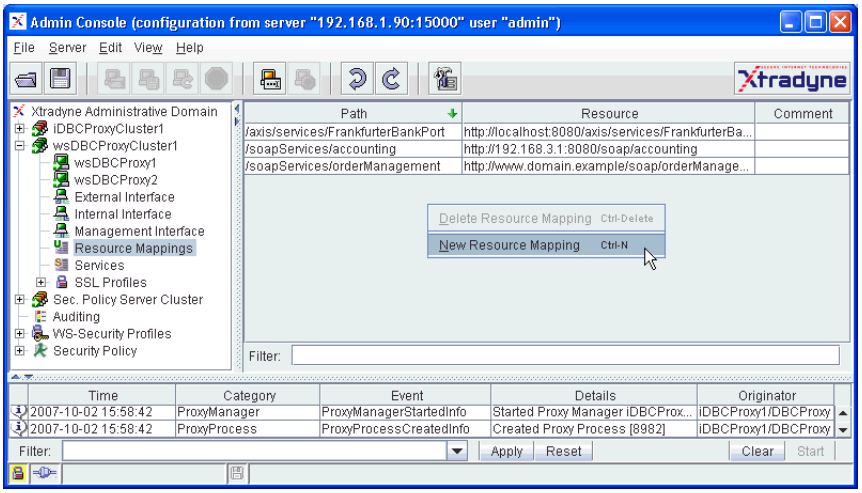


Fig. 35. Resource Mappings

When clicking with the right mouse button into the “Resource Mappings” table a context menu pops up. You can choose between:

- defining a new resource mapping (**New Resource Mapping**) and
- deleting resource mappings (**Delete Resource Mapping**). This selection is disabled when no resource is selected.

add a resource mapping

To define a new mapping choose **New Resource Mapping** from the context menu. A new table row will appear with the Web Service resource on the right side and the mapped path on the left side. The Web Service resource is the URL of the application server where your Web Service is deployed. This URL is defined in the resource properties panel and can be chosen from the drop down menu that appear when clicking on the table cell. The mapping (Path) on the left side is the path to be used by the client to contact the WS-DBC Proxy to reach the Web Service.

delete a resource mapping

You can delete a resource mapping by selecting it and choosing **Delete Resource Mapping** from the context menu.

5.10 Services

Configure services like DBC Monitoring. For the I-DBC you can additionally configure the DBC CORBA Naming Service, Flow Control, and Connection Limiting.

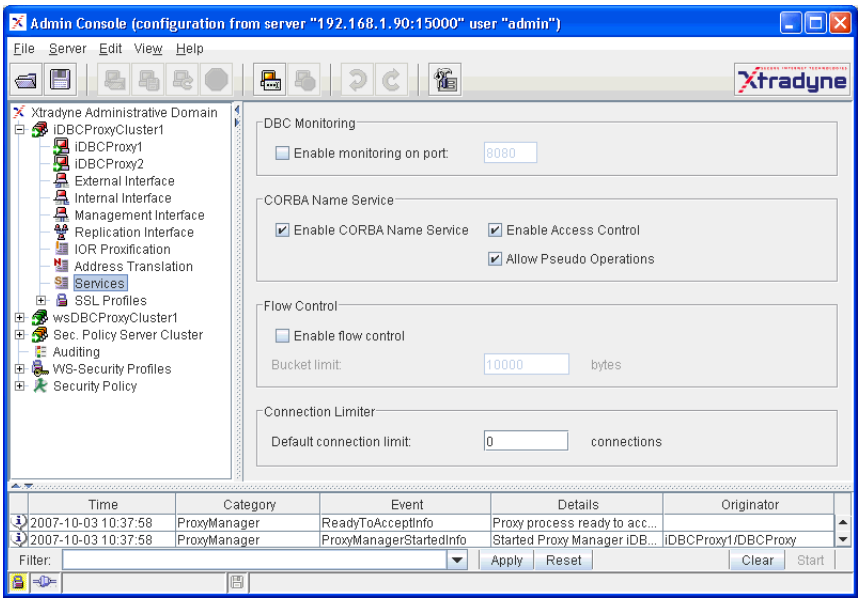


Fig. 36. Services

DBC Monitoring

If you want to monitor the Proxy, e.g. when you do load-balancing, you can enable monitoring here. Configure the port via which an external monitoring tool will be able to monitor the Proxy. Examples for such tools are:

- RedHat Piranha,
- Cisco CSS,
- dispatcher from other vendors.

I-DBC: CORBA Name Service

Enable CORBA Name Service. This service is a DBC name service functioning basically like a standard Name Service. You can specify if you want to “Enable Access Control”: Calls to the Name Service will be subject to access control. Note that if you enable

this check box you have to configure a resource for the naming service and grant the appropriate access rights. Additionally, you may “Allow Pseudo Operations” to pass.

The remainder of this chapter explains **I-DBC** configuration panels. When configuring a **WS-DBC** please proceed to section “SSL Profiles” on page 165.

I-DBC: Flow Control

To use the DBC’s flow control service check the “Enable Flow Control” box. When flow control is enabled the DBC will buffer incoming traffic that cannot be passed on. Traffic will be buffered up to the number of bytes configured in the “Bucket Limit” field. The default for this setting is 10000. If the bucket limit has been reached and further traffic is received, these messages will be discarded by the DBC.

Connection Limiter

The I-DBC allows for limiting the number of incoming GIOP connections per peer. Peer, in this case, is determined by the IP address as seen by the DBC (e.g. client IP address, NAT router address, etc.).

The default connection limit can be configured here. Additional specific limits per client IP address can be configured in expert mode. A value of 0 means unlimited. Note that if specific limits are configured, these will overwrite the default connection limit.

5.11 IOR Proxification

Define references to initial CORBA objects that can be accessed through the **I-DBC**.

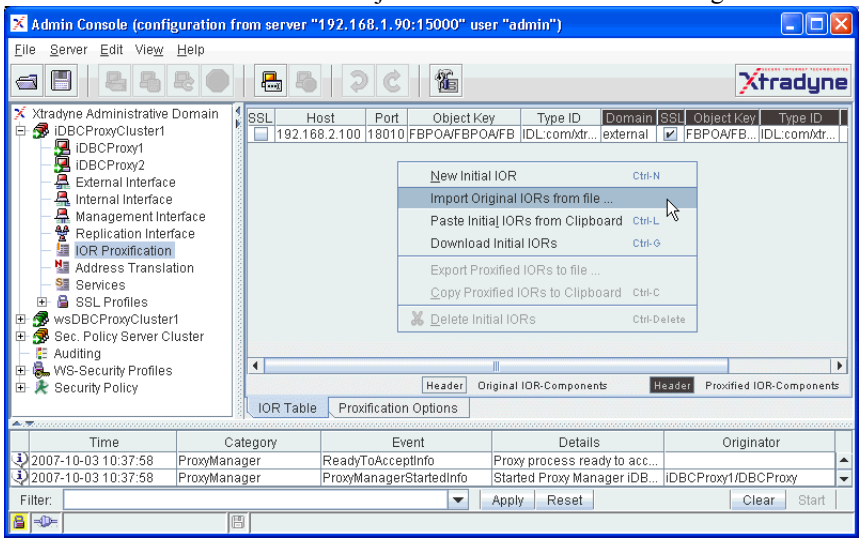


Fig. 37. Initial IOR Table

5.11.1 Initial IORs

configuring initial IORs

There are two purposes of configuring initial IORs. On the one hand the I-DBC Proxy has to know which initial CORBA services it is supposed to make accessible (through the I-DBC Proxy). This will be realized by configuring an original IOR that identifies a CORBA object. On the other hand the client needs to know where the proxy server is, i.e.,where to find the I-DBC Proxy. The client must be provided with a proxified IOR that contains the addressing information of the I-DBC Proxy instead of the original server. During request processing the I-DBC Proxy maps a proxified IOR used by the client to an original IOR which is used to pass the request to the server.

bootstrapping over the naming service

Note that when using a Naming Service as bootstrapping mechanism, you have to provide the IOR of the Naming Service in the IOR Table and export this IOR to the client.

This panel lists initial (or static) object references. Only requests to these references are allowed to pass the I-DBC. If replies contain object references the I-DBC will process them on the fly but will not display them in the table.

On the left side of the table the original IOR is displayed (gray headings). On the right side some selected parts of the automatically generated proxified IOR are displayed (dark gray headings). These can differ from the original values. The parts of the proxified IOR that are not shown are the address and port of the Proxy which are the same for all proxified IORs.

Note that usually only the original IOR part is edited. The values for the proxified IOR will be taken automatically from the original IOR. For details on when to edit the fields of the proxified IOR, please refer to section “Advanced Features” on page 157.

There are two ways to enter an original IOR into the table of initial IORs: you can define it from scratch or import it from a file. By default imported IORs are not editable with the AdminConsole.

When clicking with the right mouse button into the table of initial IORs a context menu pops up. You can choose between

- specifying a new Initial IOR (**New Initial IOR**),
- importing an existing IOR from a file (**Import Original IORs from File**),
- exporting Proxified IORs to a file (**Export Proxified IORs to File**),
- deleting Initial IORs (**Delete Initial IORs**).

The last two menu items are disabled when no IOR is selected.

create an initial IOR

With **New Initial IOR** you can create an original IOR from scratch. You will get a new table row with all fields editable in place:

- **SSL**: With the “SSL” flag of the original IOR you specify whether the server expects an SSL connection. The “SSL” flag for the proxified IOR determines whether to include a standard SSL tagged component in a generated and exported IOR. The corresponding port number contained in the proxified IOR is taken from the IIOP/SSL Acceptor Port settings.
- **Host**: The host address of the server, either name or IP address.
- **Port**: The port number of the server.
- **Object Key**: The original object key to be used in a request the I-DBC Proxy sends to the server. The proxified object key is used by the client when contacting the I-DBC Proxy. The object key is given in URL style notation as used in a `corbaloc`. The string is not NULL terminated. For characters that are not allowed as part of a URL, use the escape conventions described in RFC 2396 (the RFC can be found at `http://www.ietf.org/rfc/rfc2396.txt`). US-

ASCII alphanumeric characters are not escaped, except for the following: “ ; / : @ & = + $, - _ ! ~ * ’ ( ) % ”.

- **Type ID**: The type ID will be used by the I-DBC Proxy as part of the original IOR. The proxified counterpart will be exported to the client within an IOR.
- **Domain**: Specify the domain for which the IOR will be proxified (external or internal domain). Your choice determines which host address and port will be written into the proxified IOR, i.e., the interface definition configured in the corresponding Interfaces pane.
- **Comment**: An optional comment that describes the CORBA service.

The proxified object key will be used by the I-DBC Proxy to retrieve the original IOR. Therefore the Security Policy Server will not allow saving a configuration that contains IORs with identical object keys.

Example: Original IORs (Frankfurter Bank)

The table below shows two different original IORs of the Frankfurter Bank example.

SSL	Host	Port	ObjectKey	Type ID	Domain
yes	192.168.7.44	18011	FBPOA/FBPOA/FB	IDL:Account:1.0	external
no	samplehost.com	18010	FBPOA/FBPOA/FB	IDL:Account:1.0	external

import an original IOR

With **Import Original IORs from File** IORs can be imported from files that contain IOR strings. The IORs contained must begin with the characters “IOR:”, one per line. Unrecognized lines will be ignored (e.g., comments beginning with #). The imported IORs will be displayed in the table. The main advantage of importing an original IOR is that additional parts of an IOR, e.g., vendor specific tagged components, remain unchanged. This is especially important as clients may depend on this information.

The original part of an IOR imported from a file cannot be changed. With this restriction we ensure that the IOR will fit the service requirements. If you want to change the original parts of the IOR anyway, you have to re-import it.

export a proxified IOR

With **Export Proxified IORs to File** proxified IOR can be exported into a file. All selected IORs will be written to the file, each in a new line. If a comment is available it will be placed in a line above it prefixed with a #. It is up to you to distribute the proxified IORs to the clients for connecting with the I-DBC Proxy.

Note that the export function is only available if you are connected to the SPS! When starting the Admin Console offline or losing the connection to the Security Policy Server this menu item is grayed out. You have to (re)login first to use this function.

5.11.2 Advanced Features

In some cases it makes sense that the object key or the type ID in the proxified IOR differs from the values given in the original IOR. This section discusses these cases.

Editing the Proxified Object Key

In some bootstrapping situations the object key contains information agreed upon by the server and the client, for example, the server's IP address. (It is not conforming to a standard but some ORBs do this.) When deploying the I-DBC Proxy between client and server, this information has to be proxified too. You have to edit the **proxified** object key so that it matches requests sent by the client to the I-DBC Proxy instead of the server.

When editing the object keys, note that proxified object keys have to be unique because they will be used by the I-DBC Proxy to retrieve the corresponding original IOR.

Editing the Proxified Type ID

Usually, there is no need to edit the type ID of the proxified IOR. The only imaginable benefit of this feature is the use in access control or in interworking scenarios. In the case of access control, you can modify the original type ID (see also part 3, especially section "Resources" on page 259). You can safely change it because it is not contained in an IIOP request and thus will never be used for request processing. The type-based access control decision will be made based on the original type ID associated with the requested IOR. All IORs exported to clients contain the proxified type ID.

5.11.3 Proxification Options

Configure advanced proxification options on this tabbing pane.

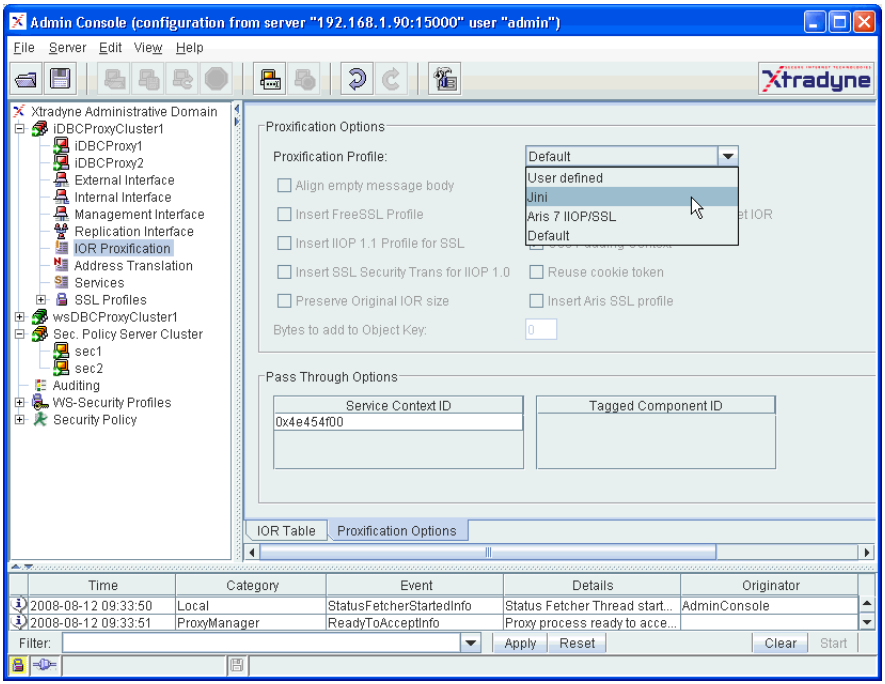


Fig. 38. Proxification Options

Proxification Options

On the “Proxification Options” panel you can choose a proxification profile from the drop-down menu. The proxification profile defines proxification options. You can choose between “Default”, “Jini”, “Aris 7 IIOP/SSL”, and “User Defined”.

In the “Default” profile the all proxification options are disabled. Only the “Use Padding Context” option is enabled. Usually, there’s no need to configure special proxification options and the “Default” profile will work fine.

If you use Jini, choose the “Jini” profile and the required settings for supporting Jini will be configured (the “Preserve original IOR size” and “Reuse cookie token” options will be enabled).

If you use Aris 7.0 (a proprietary IIOP/SSL transport plugin), choose the “Aris 7 IIOP/SSL” profile and the vendor-specific profile `0x2a` will be supported. When choosing “User Defined” the checkboxes and text field below the profile are editable and you can adjust proxification options to your needs. The following options can be configured:

- Align Empty Message Body
- Insert FreeSSL Profile
- Insert IIOP 1.1 Profile for SSL
- Insert SSL Security Trans for IIOP 1.0
- Preserve Original IOR Size
- Use Original Key
- Use Association Options of Target IOR
- Use Padding Context
- Reuse Cookie Token
- Insert Aris SSL Profile
- Bytes to add to Object Key (padding bytes)

Pass Through Options

Define Service Context IDs and Tagged Component IDs that will be passed through.

5.12 Address Translation

address mappings for outgoing connections

The virtual address on the External and Internal Interface panel defines an address mapping for incoming connections, i.e., connections from the public and protected domain to the I-DBC Proxies. Here you define address mappings for **outgoing connections**

from the I-DBC Proxies to CORBA servers both located in the public and protected domain (see figure 39, “Address Mappings”).

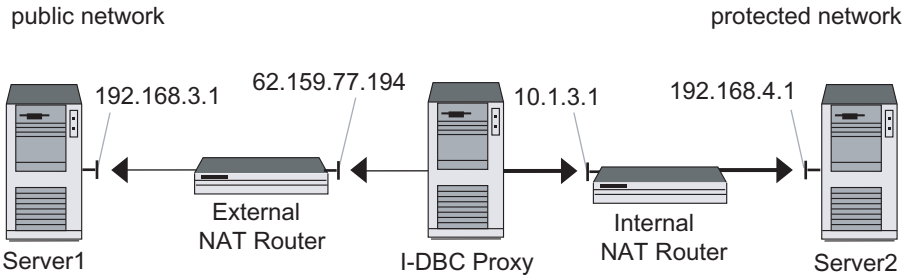


Fig. 39. Address Mappings

5.12.1 Outgoing Connections to Servers

The I-DBC Proxy connects to a CORBA server on behalf of a CORBA client. The CORBA server is commonly located in the protected network (indicated with bold arrows in figure 39, “Address Mappings”). If a NAT router is located between the I-DBC Proxy and a server, the I-DBC Proxy cannot reach the server because the address contained in the IOR is the one of the CORBA server. It must be substituted with the address of the NAT router.

Configuring Address Mappings for Outgoing Connections

Address substitutions are defined in the “Address Translation” panel. Addresses are mapped from the original host (CORBA server) to the proxy host (NAT Router).

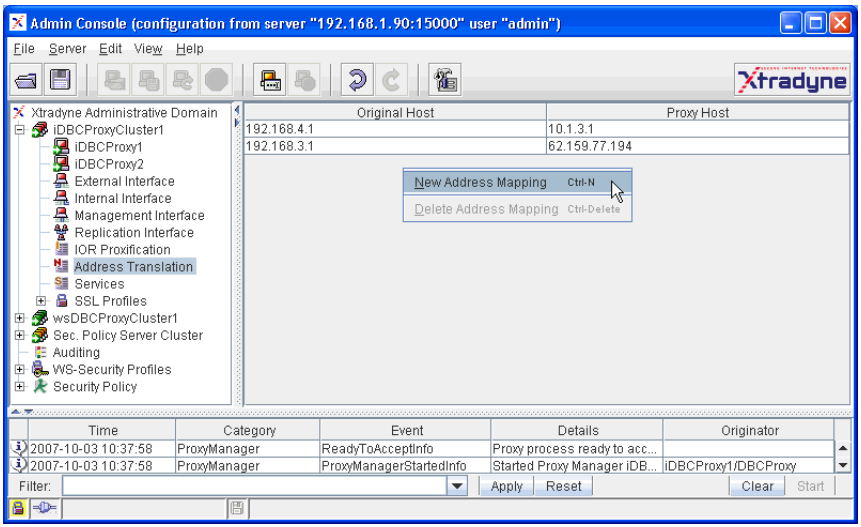


Fig. 40. Address Translation: Create a new Address Mapping

Defining and Deleting Address Mappings

Note that if you don’t use NAT the table can be left empty. Generally, if there is no address mapping for a host contained in the initial IOR table, a one-to-one mapping is assumed.

defining and deleting address mappings

By right clicking into the address translation table, you can choose between defining a new address mapping or deleting one. If you delete an address mapping needed for the specified IORs, the Admin Console prompts you to delete the affected IORs. Correspondingly, if you delete an IOR from the Initial IOR table the Admin Console will prompt you to delete all unused address mappings.

To configure the address mapping table for the scenario depicted in figure 39, “Address Mappings” you would:

- Enter the address of CORBA Server 1 (192.168.3.1) into the “Original Host” field. In the “Proxy Host” field, fill in the address of the NAT Router (62.159.77.194).
- Enter the address of CORBA Server 2 (192.168.1.2) into the “Original Host” field. In the “Proxy Host” field, fill in the address of the NAT Router (10.1.3.1).

The first mapping applies to connections *from* the I-DBC Proxy host *to* CORBA Server 1 located in the public network, the second mapping applies to connections *from* the I-DBC Proxy host *to* CORBA Server 2 located in the protected network. The original host in the address translation table corresponds to the host named in the Initial IOR table. Currently we only support host name or IP address mapping, no port mappings.

Don’t forget to specify mappings for dynamically generated IORs, i.e., for every CORBA server behind the NAT router.

5.13 Services

Configure services like DBC Monitoring. For the I-DBC you can additionally configure the DBC CORBA Naming Service, Flow Control, and Connection Limiting. Services.

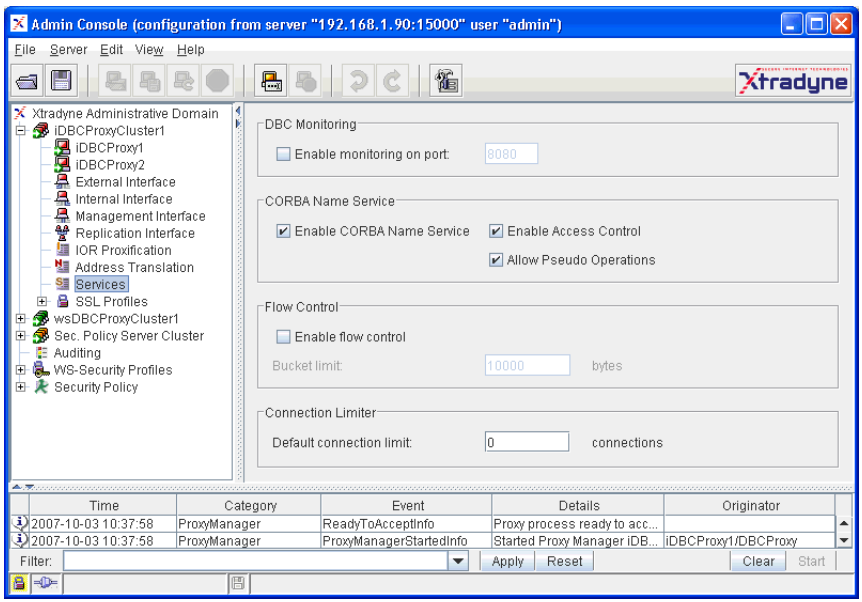


Fig. 41. Services

DBC Monitoring

If you want to monitor the Proxy, e.g. when you do load-balancing, you can enable monitoring here. Configure the port via which an external monitoring tool will be able to monitor the Proxy. Examples for such tools are:

- RedHat Piranha,
- Cisco CSS,
- dispatcher from other vendors.

I-DBC: CORBA Name Service

Enable CORBA Name Service. This service is a DBC name service functioning basically like a standard Name Service. You can specify if you want to “Enable Access Control”: Calls to the Name Service will be subject to access control. Note that if you enable

this check box you have to configure a resource for the naming service and grant the appropriate access rights. Additionally, you may “Allow Pseudo Operations” to pass.

I-DBC: Flow Control

To use the DBC’s flow control service check the “Enable Flow Control” box. When flow control is enabled the DBC will buffer incoming traffic that cannot be passed on. Traffic will be buffered up to the number of bytes configured in the “Bucket Limit” field. The default for this setting is 10000. If the bucket limit has been reached and further traffic is received, these messages will be discarded by the DBC.

I-DBC: Connection Limiter

The I-DBC allows for limiting the number of incoming GIOP connections per peer. Peer, in this case, is determined by the IP address as seen by the DBC (e.g. client IP address, NAT router address, etc.).

The default connection limit can be configured here. Additional specific limits per client IP address can be configured in expert mode. A value of 0 means unlimited. Note that if specific limits are configured, these will overwrite the default connection limit.

CHAPTER

6 SSL and WS-Security Profiles

This chapter describes SSL and WS-Security Profiles. SSL Profiles define the keys and certificates for an SSL connection. WS-Security Profiles are only of interest when configuring a WS-DBC - for details, please see page 175. On your first read you may safely skip this chapter. For a more detailed discussion on the use of SSL in DBC communication and instructions on how to replace the default keys and certificates generated while installing the SPS, please refer to "Installing Keys and Certificates" on page 201.

6.1 SSL Profiles

On the "SSL Profiles" pane you can define different SSL Profiles. These profiles are then assigned to External or Internal SSL Acceptors and Connectors. It is possible to have different SSL Profiles for External/Internal and Acceptors/Connectors, for a total of four different profiles.

There are two predefined profiles which control SSL between the DBC Proxy and the clients and targets: *SSLServer* and *SSLClient*. The *SSLServer* profile is used for incoming connections (when the DBC Proxy is in the server role). The *SSLClient* profile is used for outgoing connections (when the DBC Proxy is in the client role). The main difference between the two profiles is that the *SSLServer* profiles offers SSL v2/3 so that a client that connects to a server with an SSLv2 compliant message indicating that it would prefer to speak v3/TLS will not be rejected (see also section below).

The SSL Profiles panel offers four tabbing panes: the “Protocol, “Key & Certificate”, “Trusted CAs”, and “OCSP” pane. These tabbing panes will be explained in detail in the following sections.

For convenience, when adding a new cluster to the configuration, the Admin Console will prompt you whether the key settings shall be copied from an existing cluster in the configuration.

import from Java Keystore

Keys and certificates may as well be imported from a Java-Keystore. This is explained in detail in “Importing Keys and Certificates from JAVA-Keystore” on page 173.

6.1.1 SSL Profiles – Protocol

On this pane you can configure properties concerning the SSL protocol like the SSL version and the cipher suite.

Profile Name

Define a name (unique identifier) for this profile. This name will be used on the External/Internal Network Interfaces pane to assign the profile to communication endpoints.

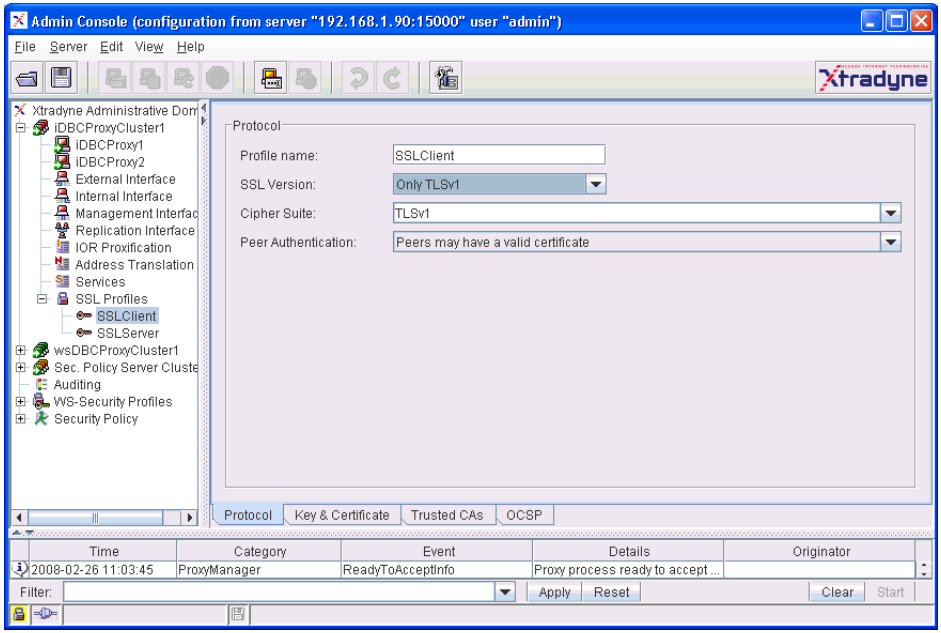


Fig. 42. SSL Profile – Protocol

SSL Version

Choose the SSL Version. The DBC supports the following SSL versions:

- Only SSLv2: We discourage you from allowing SSL v2 as it contains some security vulnerabilities!
- Only SSLv3: Support only SSLv3.
- Only TLSv1: Support only TLSv1. For the SSLClient Profile (which applies when the DBC is in the client role) it is best practice to use only TLSv1.
- V23/TLS: Support for SSL v2 and v3 is the recommended value for SSL Server Profiles (when the DBC is in the server role). The reason is, that a client may connects to a server with a v2 compliant message indicating that it would prefer to speak v3/TLS. Unfortunately, if you configure the server to only support v3/TLS, it does not recognize this “upgrade request” and the communication fails. On the

Use SSL v23 and restrict cipher suites to v3 ciphers!

other hand, if you do not want v2 at all, you may use the ciphersuite field to restrict the available ciphers to v3/TLS ciphers (see below). Thus, the “upgrade request” will be understood but only SSL v3 will be used on that connection.

Ciphersuite

choose a ciphersuite

Choose the set of ciphers to be used. The cryptographic cipher suite defines which ciphers are supported and which are not allowed. It is a string description in OpenSSL style. You can either enter the string label or choose typical settings from the drop-down menu next to the input field:

- TLSv1: The recommended setting for the SSLClient Profile.
- SSLv3: Disallows SSLv2 ciphers.
- DEFAULT: This corresponds to ALL:!aNULL:!eNULL, i.e. all available ciphers except ciphers offering no authentication and cipher offering no encryption.
- TLSv1:!EXPORT: Like TLSv1 but disallows 40 bit ciphers.
- TLSv1:!EXPORT:!aNULL:!eNULL: Additionally disallows ciphers offering no authentication and no encryption.
- SSLv3:!EXPORT: Disallows SSLv2 and the 40 bit ciphers.
- SSLv3:!EXPORT:!aNULL:!eNULL: Additionally disallows aNULL and eNULL ciphers.
- DEFAULT:!EXPORT: Like DEFAULT, but disallows 40bit ciphers.
- HIGH: Strong encryption cipher suites with key lengths over 128 bit only.
- HIGH:MEDIUM: Like HIGH, but also allow “medium” encryption ciphers (128 bit key length).

Appendix C, “SSL Ciphers” on page 355 gives a detailed description of the string format and lists the implied ciphers of the different cipher suites that can be configured.

Peer Authentication

requirements for peer authentication

This drop-down menu lets you select the requirements for peer authentication. The following options are available:

- Peers must have a valid certificate: The DBC will reject connections from peers that do not present a valid certificate.
- Peers may have a valid certificate: If the peer provides a certificate, it must be valid and trustworthy, otherwise the connection is rejected. If the peer does not provide a certificate at all, the connection attempt is successful.

- Peer certificates are ignored: The DBC will not validate the peer’s certificate.

6.1.2 SSL Profiles – Key & Certificate

This pane defines the private key and certificates that are to be used for the external and the internal connections of the DBC, respectively. The key and certificates must be provided in PEM encoding. If PEM encoding is not available, please refer to “Changing the Certificate Encoding Format” on page 216.

The private key and certificates respectively can either be stored

- on the DBC Proxy host and the filename can be provided, or
- directly in the configuration file.

Storing the private key on the DBC Proxy host is a potential security risk if an intruder can get access to the firewall host system! Therefore storing the key directly in the configuration is the preferred choice. In this case, the key is safely stored on the Security Policy Server, which is located in the protected network. It will be transmitted to the DBC Proxy host on startup for use during connection setup, but it will never be stored in the DBC Proxy host’s file system.

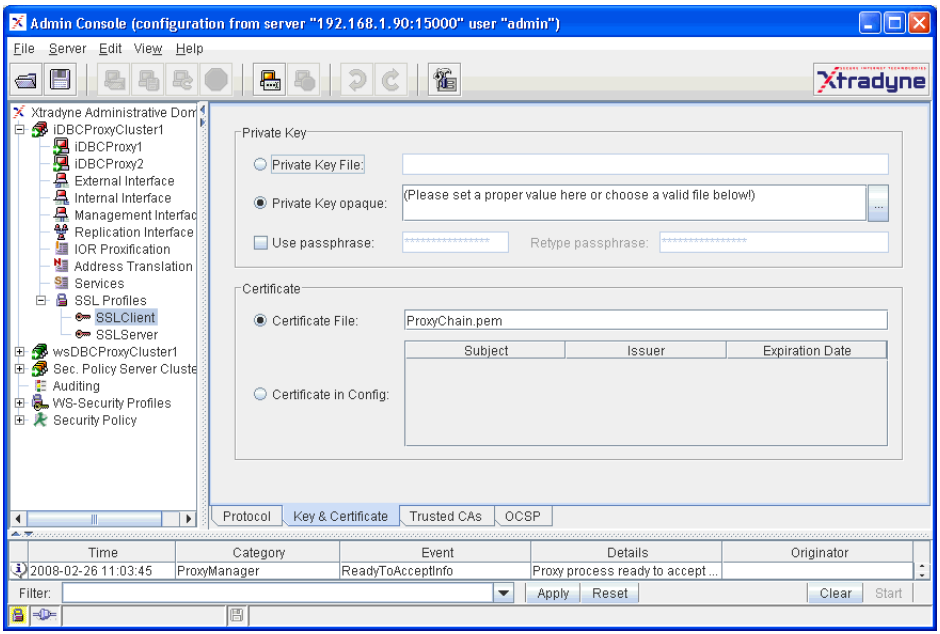


Fig. 43. SSL Profile – Key & Certificate

Private Key

In the “Private Key” part of the “SSL Profiles” pane you define the private key that is to be used for this profile:

- *Private Key File*: The name of the file on the DBC Proxy host containing the private key for this profile (in PEM format) can be configured here. If you want to use the default key which was generated during the installation of the SPS, copy the file `<INSTALLDIR>/sps/adm/ProxyKey.pem` from the SPS host to the Proxy host and place it in to the Proxy’s `adm` directory. Enter `ProxyKey.pem` into the file selection field. Note that if no absolute path is given here, the value will be taken relative to the Proxy’s `adm` directory.
- *Private Key Opaque*: The private key is stored directly in the configuration file (recommended choice). You can paste the private key into the text field (use **Ctrl C/Ctrl V** on windows or other platform-specific copy instructions). Alternatively, the private key can be loaded into the configuration by clicking the “...” button on the right. During the installation of the Proxy, the default Private Key is placed in the DBC’s configuration file and will appear in this text field.

Note that selecting a file using the file selector which appears after clicking “...” will only work if the same local path is also available on the DBC Proxy host. This is generally true only for single host installations or shared file systems.

- *Use passphrase*: If the private key is protected by a passphrase, give the passphrase here.

Certificate

In the “Certificate” part of the “SSL Profiles” pane you define the certificates that are to be used for this profile. The preferred choice is to store the certificate in the configuration file (second bullet):

- *Certificate File*: The name of the file containing the certificate chain that corresponds to the private key (in PEM format). The value will be taken relative to the Proxy’s `adm` directory. The referenced file may contain a list of public key certificates and can be extended by simply appending other certificates to the file.
- *Certificate in Config*: The certificate chain is stored directly in the configuration file. You can add certificates by choosing **Add...** from the context menu and then selecting the file containing the certificate. The Subject, Issuer, and Expiration Date of the certificated will be displayed in the table. You may export certificates

from the table into a file by selecting **Export to file...** from the context menu. Certificates may be exported in PEM, DER, or CER format.

The private key and certificates may be stored in the same file. In this case, enter the same filename in both fields.

6.1.3 Trusted CAs

Certificate Authority

Trusted Certificate Authority (CA) certificates are used to evaluate the client's certificate chain. They are also transmitted to the client during SSL handshake to indicate which CAs are accepted.

Again, the CA certificates can be read from a given file or can be stored directly in the configuration (the preferred way, described in the second bullet):

- *From File*: The location of the file containing the trusted CA certificates can be defined here. The value will be taken relative to the Proxy's `adm` directory
- *From Configuration*: The trusted CA certificates are stored directly in the configuration file. You can add certificates by choosing **Add...** from the context menu and then selecting the file containing the certificate. The Subject, Issuer, and Expiration Date of the certificated will be displayed in the table. You may export certificates from the table into a file by selecting **Export to file...** from the context menu. Certificates may be exported in PEM, DER, or CER format.

Remember that using the file selector to select a CA certificate file will only work if the local path is available on the DBC Proxy (see gray box above).

6.1.4 SSL Profiles – OCSP

OCSP – the Online Certificate Status Protocol (RFC 2560) – is used to retrieve information about the validity of a certificate in the moment of use. This is an improvement over Certification Revocation Lists (CRL) that have to be updated regularly, thus always offering windows of uncertainty about the validity.

If you'd like to use OCSP, check the "Use OCSP for Identity Validation".

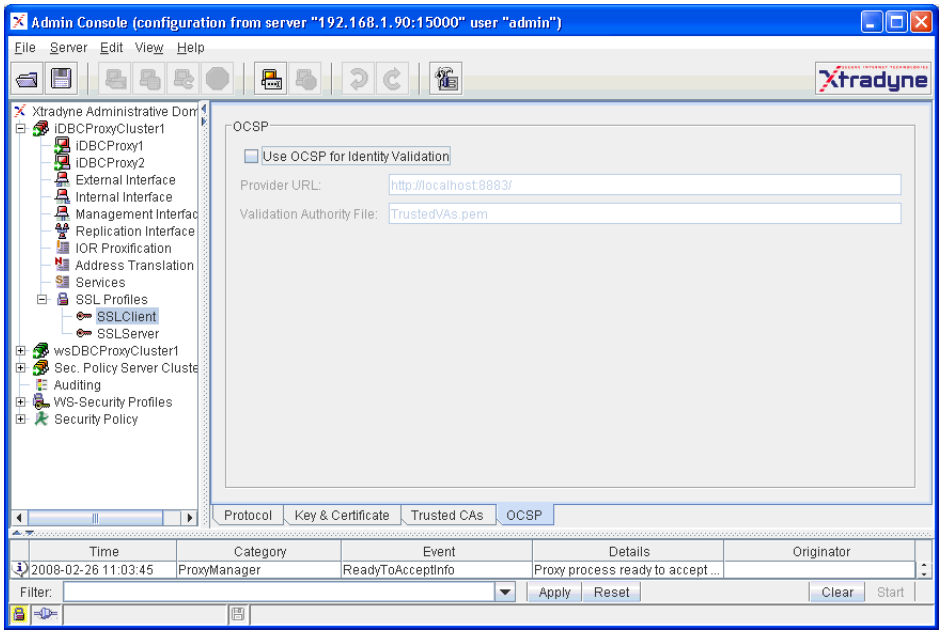


Fig. 44. SSL Profile – OCSP

The following settings need to be configured for OCSP:

- *Provider URL*: OCSP clients (e.g., the DBC) send a request to an OCSP Responder (i.e., OCSP Server) which sends a response with the validity status of the requested certificate. The request is an ASN.1 encoded message, sent via an HTTP connection. Thus, to identify an OCSP Responder, fill in the URL of that Responder in the provided text field. The URL may define HTTP or HTTPS transport. For HTTPS, no client side authentication is supported by the DBC, thus no client side key can be configured here.
- *Validation Authority File*: The DBC requires that the response is signed by the Responder. To verify the validity of the Responder's certificate, you need a list of trusted Responder certificates. The location of the file containing these certificates

is defined here (by default `TrustedVAs.pem`, this file contains the DBC Proxy CA certificate).

6.1.5 Importing Keys and Certificates from JAVA-Keystore

To import keys and certificates from a JAVA keystore, select the SSL Profile in the navigation tree for which the import shall be done. Select **Import SSL keystore** from the context menu. A Wizard will lead you through the import process.

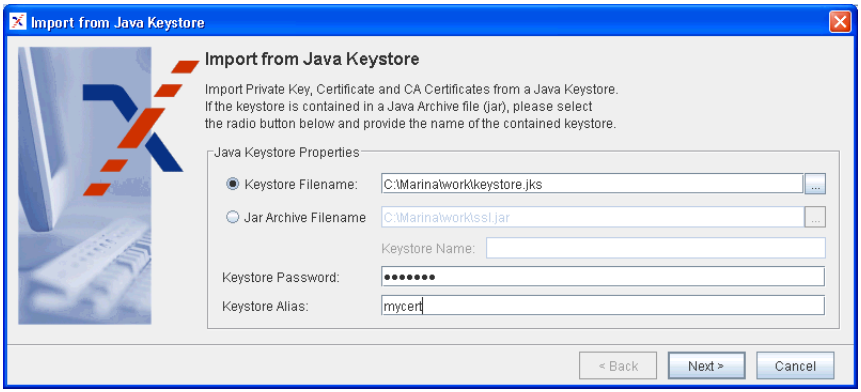


Fig. 45. Import from Keystore: Java Keystore Properties

On the first panel (cf. figure 45), provide the Keystore filename. Alternatively, a jar file which contains the keystore can be given. Select the corresponding radio button if you would like to use a jar file and provide the name of the keystore contained in the jar in the “Keystore Name” field.

Before continuing, provide the keystore password and the keystore alias in the appropriate fields.

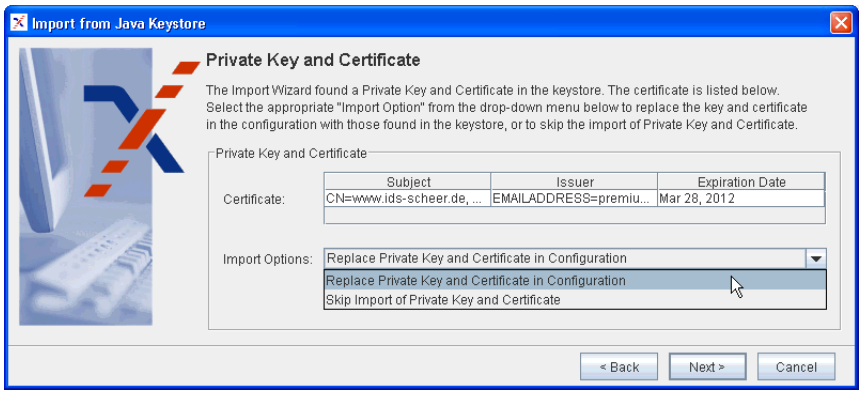


Fig. 46. Import from Keystore: Private Key and Certificate

The next panel (cf. figure 46) states whether a private key and certificate have been found in the keystore. Some of the certificate data, i.e., the subject, issuer, and expiration date is listed in a table. More certificate details can be obtained by double-clicking on the certificate in the table.

Before continuing choose an appropriate import option. Available options are:

- to replace the private key and certificate in the configuration with those found in the keystore, or
- to skip the import of the private key and certificate.

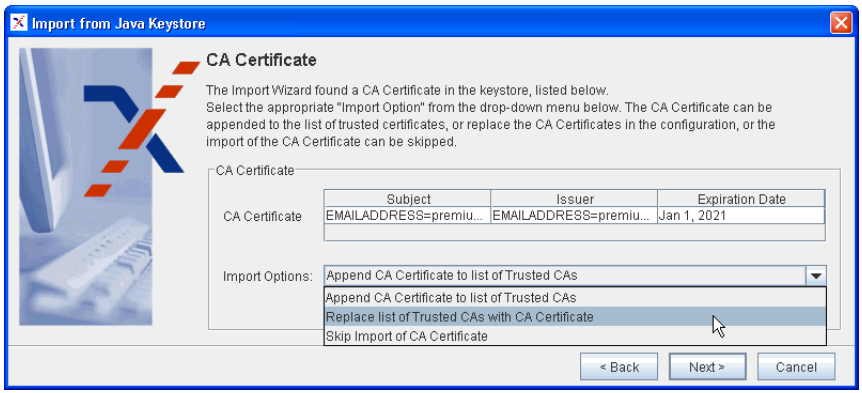


Fig. 47. Import from Keystore: CA Certificate

The next panel (cf. figure 47) states whether a CA certificate has been found in the keystore. As on the previous panel, the certificate is listed in a table.

Before continuing, please select an appropriate import option. Available options are:

- to append the CA certificate to the list of trusted CAs in the configuration,
- to replace the trusted CAs in the configuration with the one found in the keystore, or
- to skip the import of the CA certificate.

The next wizard panel gives an import summary and states whether the import was successful.

6.2 WS-Security Profiles

WS-Security Profiles apply only to the WS-DBC. They define the keys and certificates used for cryptographic operations on SOAP messages like:

- the creation of digital signatures (Signature Creation Profile),
- the verification of digital signatures (Signature Verification Profile),
- XML encryption (XML Encryption Profile), and
- XML decryption (XML Decryption Profile).

6.2.1 Signature Creation Profile

The Signature Creation Profile defines the key and certificate that will be used when creating digital signatures. The profile can be assigned on the “Resources – Outgoing Policy” panel (see “Resource Properties – Outgoing Policy” on page 277).

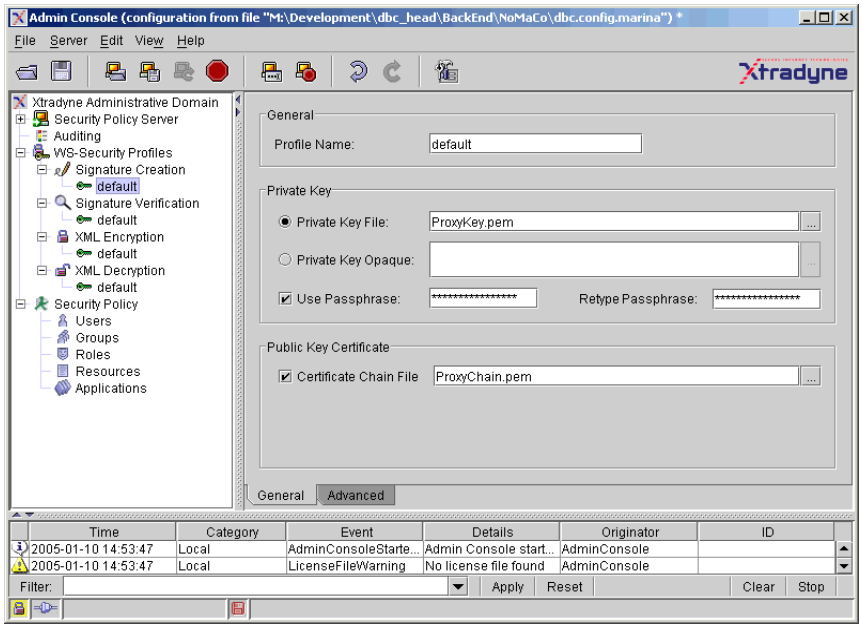


Fig. 48. Signature Creation Profile - General Tab

General Tab

Define the profile name, properties of the private key, and the trusted CAs.

Private Key

- *Private Key File*: The name of the file on the DBC Proxy host containing the private key for this profile (in PEM format).
- *Private Key Opaque*: As an alternative to providing the private key file name, the private key can be pasted into the text field.
- *Use passphrase*: If the private key is protected by a passphrase, give the passphrase here.

For details, please see section “Private Key” on page 170.

Public Key Certificate

Certificate Chain File

The file name for the public key certificate that corresponds to the private key (in PEM format). By default this is `ProxyChain.pem`.

Advanced Tab – Signature Settings

On the Advanced tab you can configure some signature settings like the signature algorithm, the canonicalization algorithm, and the key information.

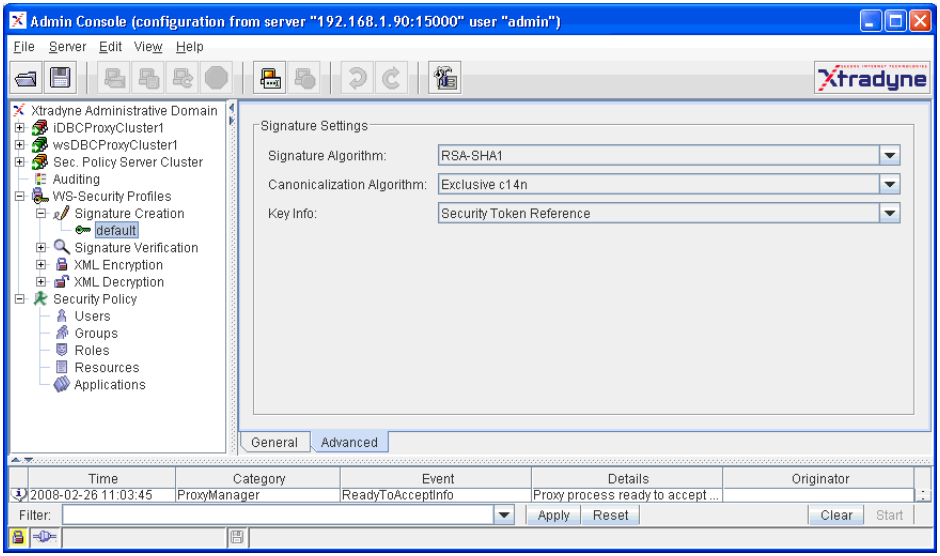


Fig. 49. Signature Creation Profile - Advanced Tab

Signature Algorithm

As signature algorithms the WS-DBC supports RSA-SHA1 and DSA-SHA1. Select one of the algorithms from the drop-down menu.

Canonicalization Algorithm

Choose one of the following canonicalization algorithms from the drop-down menu:

- Inclusive c14n
- Inclusive c14n with comments
- Exclusive c14n
- Exclusive c14n with comments

Key Info

Choose between X509 Certificate and Security Token Reference

6.2.2 Signature Verification Profile

The Signature Verification Profile defines the key and certificate that will be used when verifying digital signatures.The profile can be assigned on the “Resources – Incoming Policy” panel (see “Resource Properties – Outgoing Policy” on page 277).

General Tab

Define the profile name, trusted certificates and public key certificate on the “General” tab of the Signature Verification Profile.

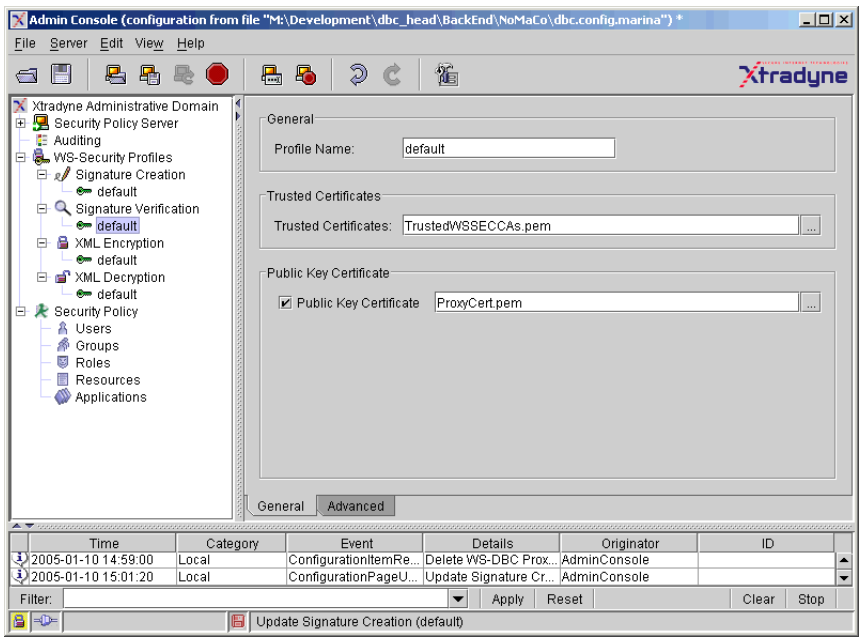


Fig. 50. Signature Verification Profile - General Tab

Advanced Tab

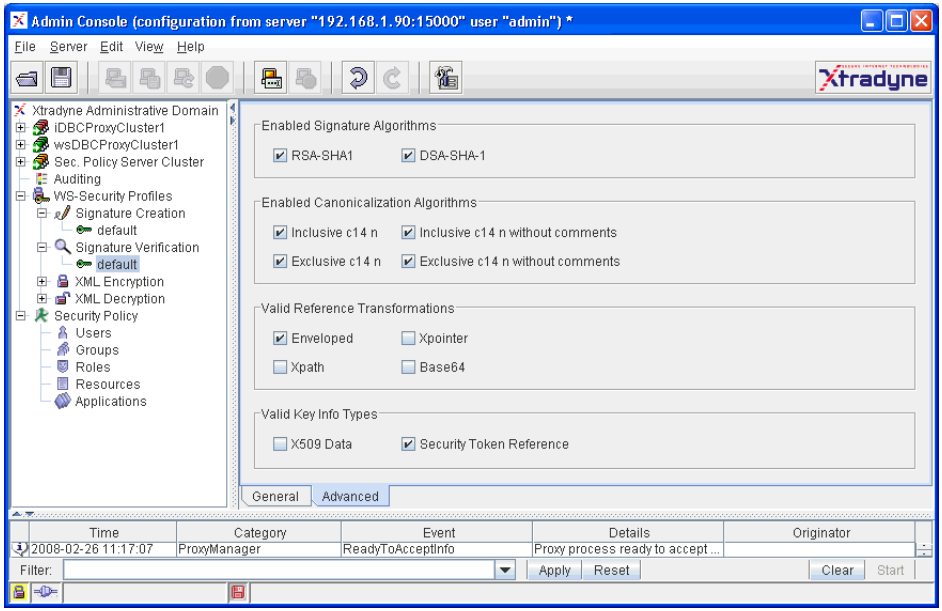


Fig. 51. Signature Verification Profile - Advanced Tab

On the Advanced tab you can select which signature algorithms, canonicalization algorithms, valid reference transformations, and valid key info types shall be supported when validating signatures.

As signature algorithms the WS-DBC supports RSA-SHA1 and DSA-SHA1. Check the box of the algorithm that shall be enabled.

As canonicalization algorithms the WS-DBC supports “Inclusive c14n”, “Inclusive c14n without comments”, “Exclusive c14 n”, and “Exclusive c14 n without comments”. Check the box of the algorithm that shall be enabled.

As valid reference transformations the WS-DBC supports “Enveloped”, “Xpointer”, “Xpath”, and “Base64”.

As valid key info types the WS-DBC supports “X509 Data” and “Security Token Reference”.

6.2.3 XML Encryption Profile

The XML Encryption Profile defines the profile name and the public key certificate that will be used for XML encryption. The profile can be assigned on the “Resources – Outgoing Policy” panel (see “Resource Properties – Outgoing Policy” on page 277).

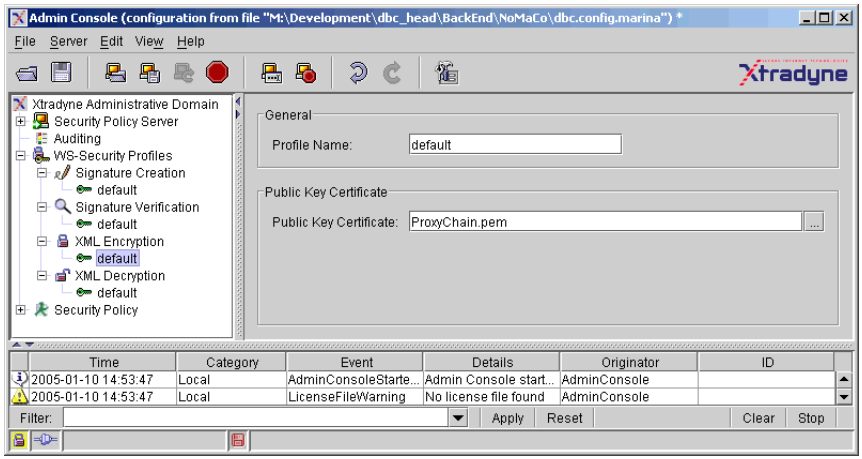


Fig. 52. XML Encryption Profile

6.2.4 XML Decryption Profile

The XML Decryption Profile defines the profile name and the private key that will be used to decrypt XML. The profile can be assigned on the “Resources – Incoming Policy” panel (see “Resource Properties – Incoming Policy” on page 269).

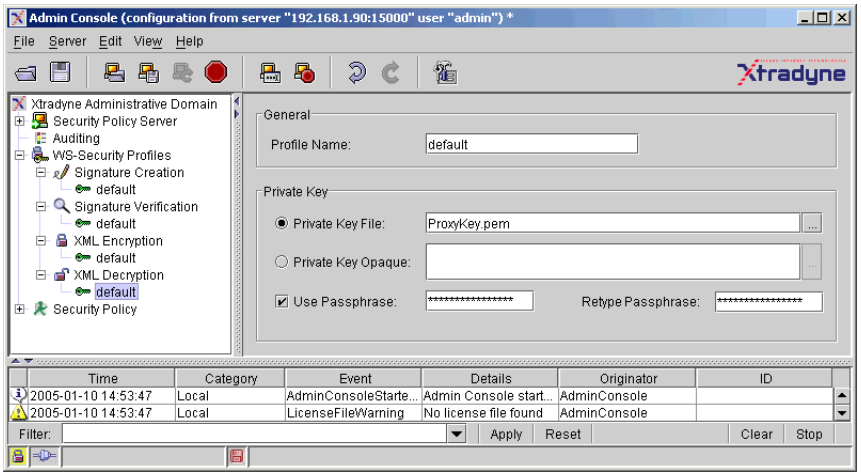


Fig. 53. XML Decryption Profile

Private Key

- *Private Key File*: The name of the file on the DBC Proxy host containing the private key for this profile (in PEM format).
- *Private Key Opaque*: As an alternative to providing the private key file name, the private key can be pasted into the text field.
- *Use passphrase*: If the private key is protected by a passphrase, give the passphrase here.

For details, please see section “Private Key” on page 170.

CHAPTER

7 Security Policy Server (Cluster)

The Security Policy Server can serve the I-DBC Proxy and the WS-DBC Proxy. The I-DBC is the IIOP Domain Boundary Controller, Xtradyne's DBC for CORBA, and the WS-DBC is the Web Services Domain Boundary Controller, Xtradyne's DBC for Web Services. The appearance of the SPS configuration panel in the Admin Console is the same for both products.

7.1 Single SPS and SPS Cluster

The DBC architecture supports High Availability and Scalability (see Chapter 1 "High Availability and Scalability" on page 11 of the Deployment Guide). Therefore, a DBC installation can consist of multiple Security Policy Servers which constitute the **Security Policy Server Cluster**. All Security Policy Servers are configured the same way so that any of those Security Policy Servers can serve requests from any DBC Proxy or Admin Console. This implies that there is only one Security Policy Server cluster belonging to a DBC installation.

In a simple scenario a Security Policy Server Cluster may contain only a single Security Policy Server as a special case of a cluster, High Availability and Scalability of Security Policy Servers are not supported in this case.

7.2 Security Policy Server Cluster Properties

This panel defines the properties of the Security Policy Server Cluster.

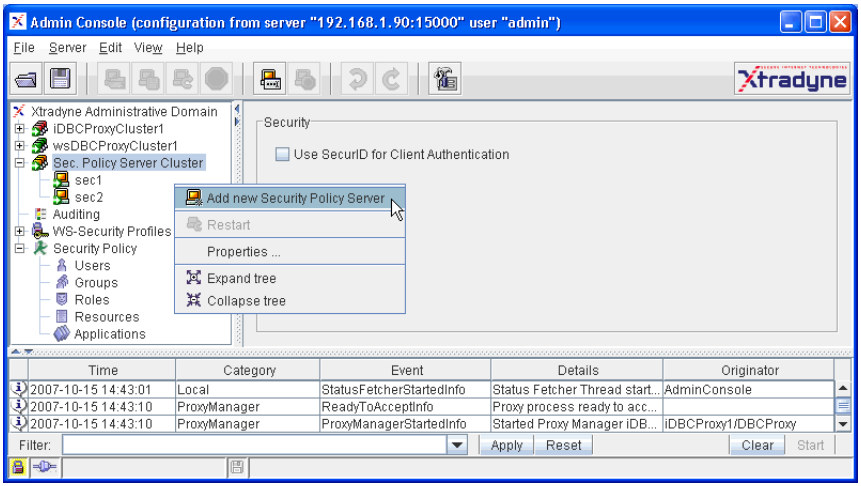


Fig. 54. Security Policy Server Cluster

Note that when working in the “Single Security Policy Server View”, all the properties you can configure in the “Security Policy Server Cluster” pane can be configured on the “Security Policy Server” pane.

SecurID for Client Authentication

If you want to use SecurID authentication check the “Use SecurID for Client Authentication” box in the general part of the Security Policy Server Cluster. This applies **only** to the **I-DBC Proxy**. If this box is checked the DBC authenticator on the SPS is activated (cf. Chapter 7 “I-DBC Authentication” on page 59 of the Deployment Guide). The DBC authenticator is needed to interact with the RSA ACE/Server which in turn performs dynamic two-factor RSA SecurID authentication. Note that the RSA ACE/Server is not part of the DBC installation and has to be installed separately. Also note that SecurID client authentication applies to clients authenticating to the DBC Proxy, not to administration users authenticating to the SPS.

7.3 Security Policy Server

The “Security Policy Server Properties” pane lets configures the management network interface of the Security Policy Server.

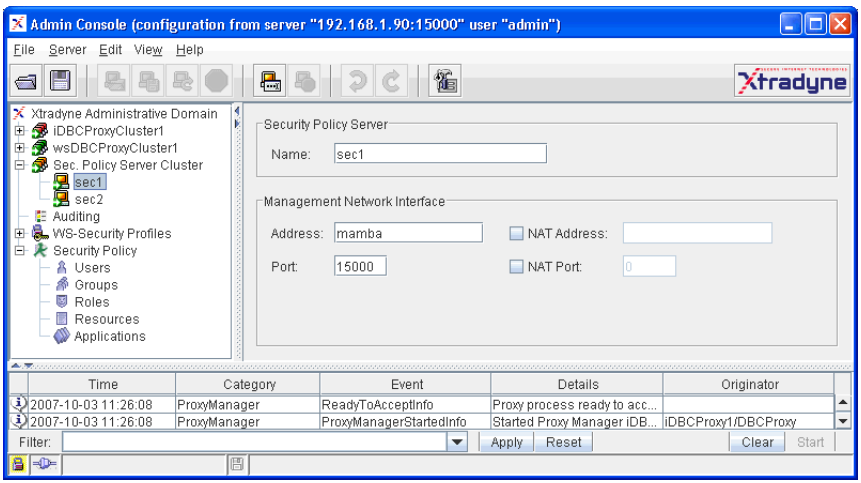


Fig. 55. Security Policy Server Properties

Adding and Deleting Security Policy Servers

You can add and delete single Security Policy Servers by clicking with the right mouse button on the Security Policy Server (Cluster) on the left side of the panel. You can also choose **Edit➜Add New SPS** and **Edit➜Delete SPS** from the menu bar.

Note that when working in the “Single SPS View”, all the properties you can configure in the “SPS” pane can be configured on the “Management Interface” pane.

7.3.1 Security Policy Server Name

In the upper part of the panel you can name the Security Policy Server. Names have to be unique for one DBC installation.

Security Policy Server Name

7.3.2 Management Network Interface

Note that in the “Single SPS” view these properties are located on the sub-panel “Management Network Interface” below the “Security Policy Server” node.

The Management Network Interface is used by all DBC Proxies to connect to the Security Policy Server. The same interface is contacted by the Admin Console. The following properties can be configured:

Address

- *Address*: The host name or IP address of the Security Policy Server host.

Port

- *Port*: The port number (default is 15000). If you change this setting, save the configuration to the SPS, and restart it. Then open the “Preferences” panel and enter the new port number in the “Server Address” field. Relogin with the Admin Console and restart the DBC Proxy.

NAT Address

- *NAT Address*: If Network Address Translation is active between a DBC Proxy Cluster and the Security Policy Server host, check the “NAT Address” box and enter the translated IP address.

NAT Port

- *NAT Port*: If Network Address Translation with port mapping is used the NAT port can be entered here.

See following section “NAT between the DBC Proxy and the SPS” for the details on Network Address Translation for the Management Interface.

When using the standard mode of the Admin Console, there is one restriction: all DBC Proxy Clusters must see the Security Policy Server under the same address. If this restriction does not fit your requirements, you can use the expert mode to configure this.

Note that if the Admin Console connects to a virtual IP mapped to the management interface of the Security Policy Servers (SPS), the traffic redirector will choose one of the SPS’s. You can also connect to one of the SPS’s directly.

7.3.3 NAT between the DBC Proxy and the SPS

If Network Address Translation (NAT) is active between the DBC Proxy host and the Security Policy Server host, check the “NAT Address” box for both directions, i.e., for the DBC Proxy contact point and the Security Policy Server contact point. Figure 56, “NAT Router between the DBC Proxy host and the SPS host” displays the situation.

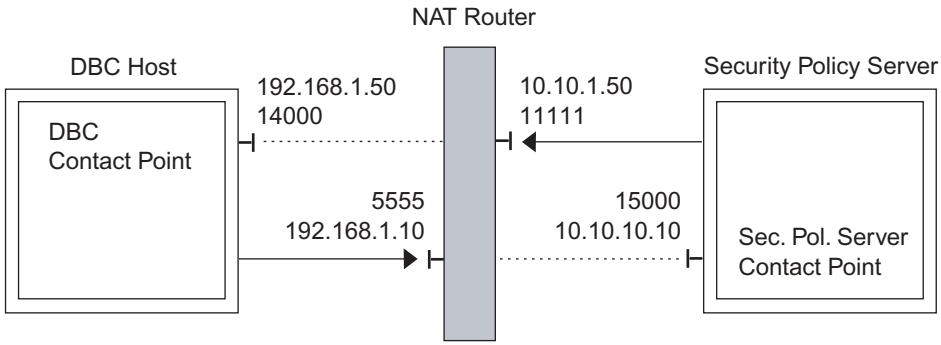


Fig. 56. NAT Router between the DBC Proxy host and the SPS host

Enter the translated host name or IP address and port into the appropriate fields. In the above example scenario, the DBC Proxy can connect to the Security Policy Server’s IP address 192.168.1.10 at port 5555. The Security Policy Server can connect to the DBC Proxy’s IP address 10.10.1.50 at port 11111.

NAT may also be only one-way, i.e., only one side is hidden. In this case, you don’t need to check the NAT field for both directions.

Using Host Names or IP Addresses

hosts must be reachable

When using host names in the fields described above, be sure that proper name resolution is available to both participating hosts. If NAT is not active, the name configured in the “DBC Proxy Properties” pane for the Management Interface must be resolvable from the Security Policy Server Host. When NAT is active for the DBC Proxy Management Interface, only the “NAT Address” part will be used by the Security Policy Server. It suffices that the address is valid on a DBC Proxy host.

The same applies for the Security Policy Server. A DBC Proxy host must be able to resolve the Security Policy Server host name, or reach the NAT Address, when NAT is active. Additionally, the Security Policy Server must always be able to resolve its given host name, because that is the interface it binds to.

CHAPTER

8 Audit Policy

An audit policy specifies which and where auditable events are logged. Event notifications are messages created by the DBC Proxy and the SPS.

8.1 Introduction

recording security relevant events

In addition to specifying security policies for the DBC, it is also necessary to monitor the system behavior. In order to determine exactly what went wrong and why a perceived breach of security was not prevented by mechanisms and policies in place, an *audit log* is required. An audit log is a record of security-relevant events that the system observed and that can be analyzed to determine the effectiveness of security services as well as the reasons and circumstances of system failures.

The DBC supports recording events in audit logs, but no additional tools are provided to analyze these logs, e.g., *Intrusion Detection Systems* (IDS) that detect correlations between security-critical events that would indicate attacks. These tools are separately available.

8.1.1 Audit Events

Event notifications include a name denoting the event, a time stamp, and the originator. Events related to IIOP messages (I-DBC) or SOAP message (WS-DBC) processing additionally carry a message identifier that facilitates the correlation of events generated during a single request. Most notifications carry additional event-specific information, e.g., the audit event for a TCP connection request from a client will include both the source and the target IP address and the TCP ports. The time stamp records the exact time when the connection was established. In this case, the originator of the event notification is the DBC Proxy.

Audit Event Types

Events are separated into the following types:

- **Error** events: indicate significant problems, e.g., a loss of functionality or data.
- **Warning** events: indicate conditions that are not immediately significant, but that may cause future problems, for example the consumption of system resources.
- **Information (Info)** events: indicate infrequent but significant successful operations, for example, when a configuration change has been recorded.
- **Success Audit** events: are security events that occur when an audited access attempt is successful, for example, a successful logon attempt.
- **Failure Audit** events: are security events that occur when an audited access attempt fails, for example, a failed logon attempt.

8.1.2 Event Flow

Events occur in different system components: in the Proxy, the Security Policy Server, and the Admin Console. The corresponding notifications are created as specified in the audit policy (see below) and consumed by the Security Policy Server. The Security Policy Server finally forwards the event notifications that it receives to a logging facility, which can be either `syslog` or an arbitrary unix command. Event processing like event correlation or generation of alarms can then be done using existing facilities or third party products.

8.2 Audit Policy

The set of events that are considered relevant are specified in an *audit policy*, which also assigns priorities to events. The DBC will only generate notifications for events that are selected from the set of auditable events. Skipping the creation of notifications that are not relevant improves the overall system performance.

enabling and disabling events

On the “Auditing” panel the audit policy can be managed and adjusted at run-time to select or deselect events as required. Administrators might, for example, want to see all events until they are reasonably confident that the system works as expected, then disable all notifications which do not indicate relevant failures. In the reverse case, they can

simply “switch on” previously disabled events if they need to diagnose specific issues. Additionally, they can specify which facility shall consume notifications.

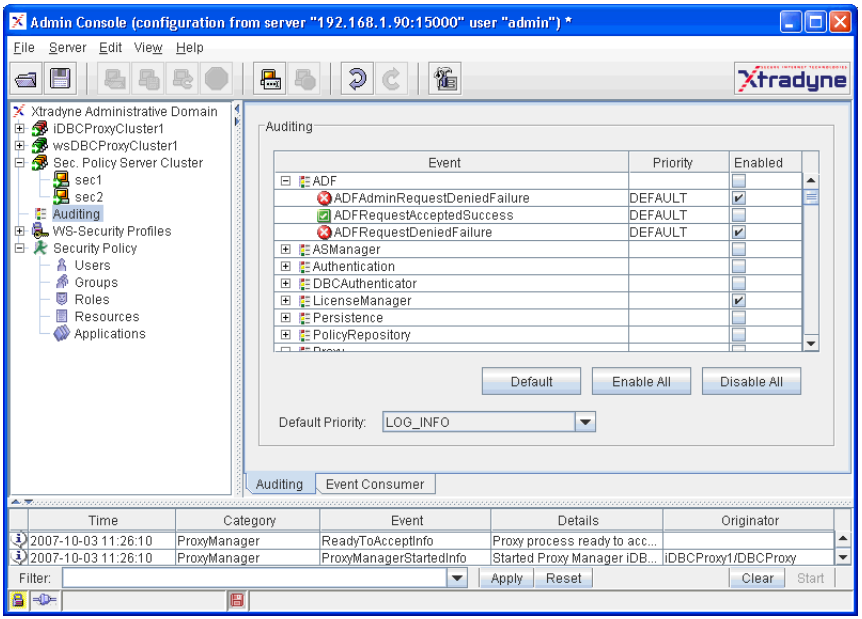


Fig. 57. Auditing – Configuring Audit Events

Events are sorted by their category and displayed hierarchically. Click on the ⊞ next to the event category to see all events of the category. To enable all events of a category check the “Enabled” box for the category. If the “Enabled” checkbox is unchecked for the whole category all events of the category will be disabled.

You can switch to a list view when clicking on the icon next to the “Event” column header.

Audit Event Categories

The set of auditable events covers a variety of event categories which are summarized here (a complete list of events is listed in appendix A, “Audit Events” on page 321):

- Proxy Events: Operational status: started, resource limits reached, etc.
- Connection-related Events:
  - GIOP connection (**I-DBC**): accepted, established, closed, diverse faults
  - HTTP connection (**WS-DBC**): accepted, established, closed, diverse faults
  - SSL connection: handshake success/failure, accepted, closed, details
- Authentication Events: mechanism, success and outcome, failure
- Authorization Events: access allowed/denied
- Policy Server Events: started, policy changed, license expired, etc.
- SOAP message processing (**WS-DBC**):
  - Schema validation: success, failure, details
  - Digital Signature Verification: success, failure, details
- IIOP message processing (**I-DBC**):
  - IOR processing: new original IOR proxified/deproxified
  - Parameter Checking: success, failure

8.2.1 Event Consumer

On this tabbing pane you can define Event Consumer details.

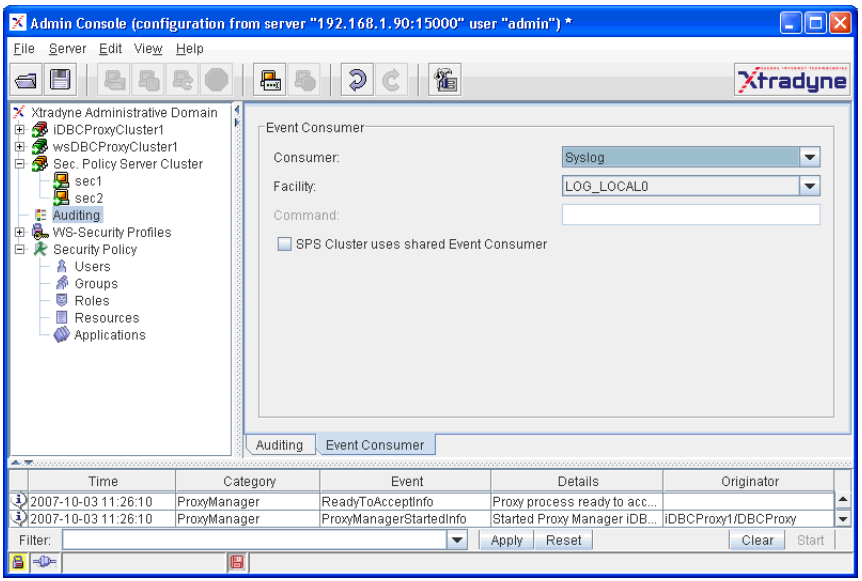


Fig. 58. Auditing – Configuring the Event Consumer

Currently audit events can be logged via `syslog` or an external command. The external command can be any UNIX command, for example:

external command

- `cat >>/tmp/event.log`
- `/usr/bin/tee /tmp/events.log 1>&2`

By default the event consumer is `syslog`. `syslog` writes logging messages to the directory `/var/log/messages`. To observe the DBC's behavior, (as root) type:

`syslog`

```
tail -f /var/log/messages
```

This is especially recommended when changing anything in a running configuration. When using syslog you can assign a priority to each event. This priority is passed on to syslog (see `syslog` manual page for details).

By default all events are replicated for the SPS cluster, so that each event log on each SPS host logs all occurring events. If your SPS cluster uses a shared event consumer, for example a syslog server to which all SPS hosts forward their syslog messages, this event replication will cause that one event will be logged several times.

shared event consumer

To prevent this you can check the box “SPS uses shared event consumer”. If activated, no event replication will be done, i.e., one SPS will log only those events generated by itself or the proxies it administers. Note that the event browser of the AdminConsole will also show only those events of the SPS it is connected to.

8.2.2 Event Priorities

In the lower part of the panel a list of available events is displayed. You can assign a priority to each event and enable or disable the delivery of the corresponding notification.

default events

Per default all events that indicate failure are enabled and their notification priority is INFO. To diagnose problems with a DBC configuration, it might be useful to enable more notifications. Below the event list there are some buttons for your convenience to:

- reset to the default setting,
- enable all notifications,
- disable all notifications.

8.3 *SNMP Support*

The Simple Network Management Protocol (SNMP) is a vendor-independent protocol standardized by the Internet Engineering Task Force (IETF). The DBC is able to generate SNMP traps from Xtradyne Event Messages. SNMP traps provide a mechanism for applications to send asynchronous notifications to a management station to signal relevant state changes like failure conditions, alarms, status information, and so on.

8.3.1 Mapping between events and trap messages

The mapping between Xtradyne audit events and SNMP traps defines three SNMP notification types which correspond to Xtradyne “Failure”, “Success”, and “Info” event types. This mapping has been chosen to distinguish between failure conditions and status information events easily. The notification types have been defined using the Structure of Management Information (SMI). The SMI file is included in the Security Policy Server package located in `<INSTALLDIR>/sps/adm/XtradyneEventMIB.txt`.

The notification types share a common structure. Two vendor specific parameters have been defined to propagate the event information:

- *eventName*: Name of the Xtradyne event.

- *eventInfo*: Complete event information propagated with Xtradyne event messages.

8.3.2 Activating SNMP trap generation

The following steps have to be carried out to activate SNMP trap generation:

- Go to the “Audit Policy” panel. In the “Event Consumer” part of the panel set the “Consumer” option to “External Command” (cf. “Event Consumer” on page 193).
- Now, set the External Command” field below to `../bin/event2trap.sh` and write the configuration to the SPS.

By default `event2trap.sh` will send traps to the trap daemon on the host where the SPS is installed. If you want to send traps to a remote management station you will need to change the setting of variable `TARGET_ADDRESS` contained in `event2trap.sh`:

- Edit the file `<INSTALLDIR>/sps/bin/events2trap.sh`.
- Replace the “localhost” setting of variable `TARGET_ADDRESS` with the IP address or hostname of the remote management station and save the file.

8.3.3 Customizing HP Openview NNM Alarm Browser

To provide improved display of Xtradyne SNMP traps with the Alarm Browser of HP OpenView NNM an event configuration file is provided. To customize your HP OpenView installation you will need to perform the following steps:

- Copy the files `XtradyneEventMIB.txt` and `ovalarm.conf` (located in `<INSTALLDIR>/sps/adm/`) to the host where HP OpenView NNM is installed.
- Start HP OpenView NNM
- From the “Options” Menu choose “Load/Unload MIBs: SNMP”.

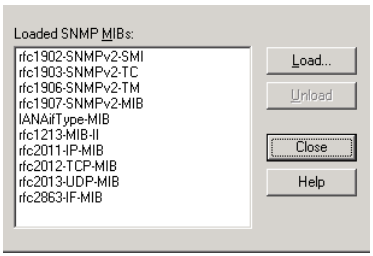
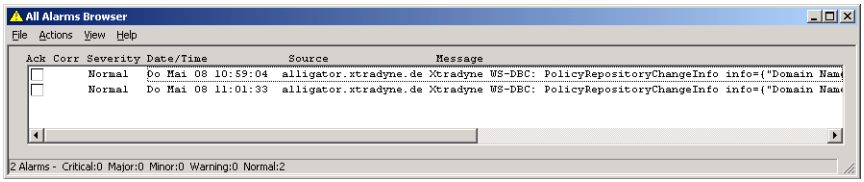


Fig. 59. HP OpenView NNM “Load/Unload MIBs: SNMP” dialog

- The “Load/Unload MIBs: SNMP” dialog will appear (see figure 59). Press the load button and choose `XtradyneEventMIB.txt` with the file selector dialog.
- A dialog appears stating that the MIB has been successfully loaded and notification type definitions have been found. Press the “OK” button to let HP OpenView enter these definitions into the event system.
- Now, press the “Close” button to exit the dialog “Load/Unload MIBs: SNMP” and terminate HP OpenView NNM.
- Now insert event configurations into the `trapd.conf` configuration file which is part of your HP OpenView NNM installation. Execute from the command line:
  `xnmevents -replace <FILEPATH>\ovalarm.conf`
- Start HP OpenView NNM again and choose “Alarms” from the “Fault” menu. The event display should now look as shown in the figure below.



CHAPTER

9 Expert Mode

Not all configuration properties are accessible in the default appearance of the Admin Console. Some features are only adjustable in the expert mode. This chapter explains how to configure settings for the DBC Proxy using the expert mode.

9.1 Dictionaries - An Introduction

The DBC's configuration data is organized in data structures called *dictionaries*. Each dictionary contains the configuration data of a certain aspect of the DBC. For example, data concerning the Security Policy Server is stored in the dictionary `securityServer`. A dictionary contains key-value pairs where the basic types are plain strings. These values can also be sub-dictionaries and vectors. A vector is an ordered list of values. These rules allow the structured storage of data. More detailed information about configuration keys used in the DBC can be ordered by emailing to `support@xtradyne.com`.

9.2 The Dictionary Explorer

The expert mode provides a facility to browse and edit configuration dictionaries: the Dictionary Explorer. You can switch to the expert mode by choosing **View➜Switch to Expert Mode** from the menu bar or, for convenience, by clicking the "Expert Mode" symbol in the tool bar. As in a file explorer you can see the hierarchical structure of the internal configuration data. This structure corresponds to the underlying configuration file

format. You can comfortably edit the raw configuration data with the Admin Console in expert mode instead of using a text editor on the configuration file.

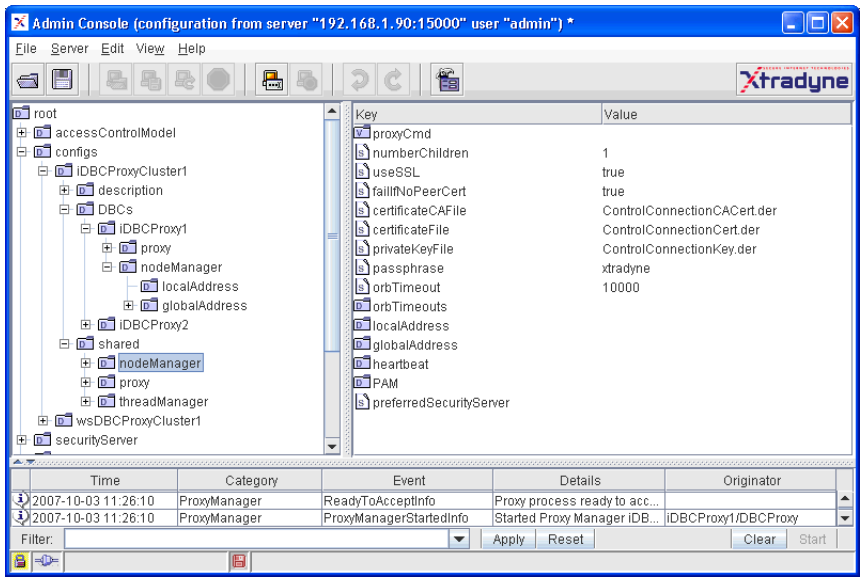


Fig. 60. Expert Mode

Note that the direct manipulation of configuration data is only recommended for experienced users. You have to know what you are doing! You can order more detailed information about the current set of mandatory and optional DBC configuration data by emailing to `support@xtradyne.com`.

The Dictionary Explorer is a generic tool that lets you build any hierarchical structures. Generally, there is no syntax checking for values. When you change them, you have to verify the correctness yourself.

When switching from the expert mode back to the standard mode the Admin Console checks the existence of some mandatory keys in the configuration. This only guarantees the working of the Admin Console, but not the correct operation of a DBC installation with such a configuration!

9.3 Editing Entries

On the left-hand side of the Dictionary Explorer you see the top level dictionaries in a tree representation. Double-click on a dictionary to view its sub dictionaries or vectors

(dictionaries are labelled with “D” while vectors are labelled with “V”). The right side shows the content of the selected item as a list of name-value pairs where only values with the basic type “string” are displayed in the right column. Such values are editable in place. The values contained in structured types will be visible only when selecting its parent in the tree. To find a particular value, you may have to descend down the hierarchy by double-clicking the name of higher elements in the path.

The name of an entry is the name by which the value is known to its parent. If the parent is a dictionary, the name corresponds to the key by which the value is registered in the parent dictionary. Such keys are editable in place.

In the case of a vector as parent the children are named like the string “element[n]”, where “n” is the index of this element in the vector. These names are not editable and will be only determined by their position in the parent vector.

9.4 Insert New Entries

Select the dictionary or vector on the left side of the panel into which you want to insert a new entry. Choose the type of the new entry in the Edit menu or alternatively in the context menu, which will pop up when clicking the right mouse button anywhere in the panel. You can insert a dictionary, a vector, or a string. Newly created values are empty and editable in place.

If the parent is a dictionary, a new name will be generated, which is the initial key by which the newly created value is known to its parent. You can edit this key by double-clicking on it. The name of a new vector entry cannot be changed, it will be determined by the order of insertion.

9.5 Copy, Cut, Paste, and Delete Entries

All entries in the dictionary structure can be copied, cut, pasted, and deleted. The corresponding operations can be reached via the **Edit** menu or the context menu.

9.6 Importing and Exporting Dictionaries

Dictionaries are the data structures that contain the configuration and policy information used by the DBC. Dictionaries can be exported to make backup copies of working configurations, or to share policy data between systems. Making backup copies is recommended practice.

Dictionaries can be exported to a file by selecting the dictionary you want to export in the tree view on the left side of the Admin Console. Click on the dictionary with the right mouse button and choose **Export**.

Dictionaries can also be imported from files by selecting the node you want to import in the tree view on the left side of the Admin Console. Click on the node with the right mouse button and choose **Import (Merge)** from the context menu. This may overwrite an existing configuration, so it is recommended to make a backup of the existing configuration by exporting first.

CHAPTER

10 *Installing Keys and Certificates*

This chapter describes the trust relations between DBC components and the required keys and certificates that are used within the DBC. This chapter is aimed at security administrators and describes where credentials are installed and how the default keys and certificates can be replaced with trusted keys and certificates of your own.

Reading this chapter requires a basic knowledge of cryptographical concepts and SSL. It is beyond the scope of this chapter to provide a thorough SSL tutorial. However, if you would like to learn more about SSL, Xtradyne provides training on this subject.

For a quick reference on how to install keys and certificates, please proceed to section "Replacing Keys and Certificates" on page 210.

Topic	**Can be found on**
Trust and other concepts	page 202
Trusting external (client) certificates	page 205
Installing keys for the communication with the external/internal network	page 210
Checking permissions for key files	page 215
Installing keys for the control and administration connection	page 215
Changing the certificate encoding format	page 216

10.1 Trust Establishment

To establish mutual trust between DBC components (the DBC Proxy, the Security Policy Server, and the Admin Console), each component of the DBC software needs

- a *public key certificate* issued by a trusted *Certification Authority* (CA) and sent to other components during the authentication,
- a *private key* that complements the public key,
- a trust database, i.e., a list of trusted public key certificates (usually CA certificates) which the peer's certificate is validated against.

The rest of this chapter explains these issues in more detail. An in-depth discussion of SSL or Public Key Cryptography is beyond the scope of this chapter, however. A general understanding of these topics is assumed.

10.2 Certificates and Certification Authorities

For security reasons, the DBC has two different kinds of communication links (explained in detail in section "Internal Communication Links and Trust Relationships" on page 63):

- The application connections between the DBC Proxy and clients and servers in the protected or public domain. Keys and certificates used on these connections are signed by a *Certification Authority (CA)* called *ProxyCA*.
- The administrative and control connections between the DBC Proxy, the SPS and the Admin Console. Keys and certificates used on these connections are signed by another Certification Authority called *ControlConnectionCA*.

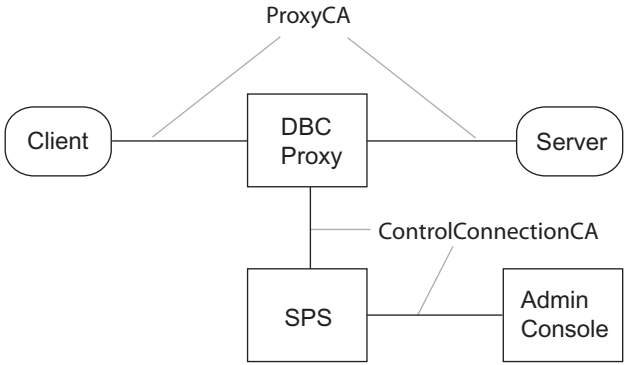


Fig. 61. Certification Authorities in the DBC installation

The distinction between application connection and control connection and the usage of different CA certificates is important because of the different levels of sensitivity and exposure of these keys: The keys on the application connections are externally visible. Also, these keys are likely to be used in a much larger number of transmissions than the keys for the control connection, which is a purely internal communication link. Thus, the DBC Proxy's keys for the application connections may require more frequent updates, e.g., in case a key is suspected to be compromised.

The CAs and the keys and certificates they sign are created during the installation. The proxy CA certificate (*ProxyCACert*) is stored in PEM format on the DBC Proxy host. The control connection CA certificate (*ControlConnectionCACert*) is stored in DER format on the SPS host, on the DBC Proxy host, and on the Admin Console host.

The corresponding private keys were used to perform the CA signature on all public key certificates in the default installation and are the ultimate source of authority in this setting. These keys are not installed with the DBC distribution so that the default CAs cannot be used to sign additional certificates, which would be a security risk.

A complete list of keys, certificates, and CA certificates is given in table 4, "Default Keys and Certificates for application connections" and table 5, "Default Keys and Certificates for Control and Administration Connections".

10.2.1 Trust Stores and Trusted CAs

The different components of the DBC software use different *trust stores* to determine if a public key certificate was signed by a trusted CA. By trust store we mean a file that contains a list of public key certificates that the DBC Proxy will accept as trustworthy CAs.

Trust Store for the Application Connections

The trust store for application connections can be kept in the configuration and CA certificates can be add and removed using the Admin Console. Alternatively, the trust store may be read from a file located on the DBC Proxy host. The file name can be configured with the Admin Console.

Trust Store for the Control and Admin Connections

For the control and admin connections (between the DBC Proxy, the SPS, and the Admin Console), the trust store is the file `ControlConnectionCACert.der`, i.e., for these connections the DBC components trust only this CA. Note that this trust store contains only a *single* CA certificate, not a list of CA certificates.

To make DBC components accept public keys signed by a different CA for the control and admin connections, the file `ControlConnectionCACert.der` has to be replaced by a new CA certificate in DER encoding.

10.2.2 Application Connections

SOAP senders that are located on hosts in the public domain communicate with the DBC Proxy. When using SSL to connect, the DBC Proxy

- *always* offers to authenticate itself to clients using a server certificate (this behavior is not configurable),
- *can* be configured to ask SOAP senders for valid client certificates signed by a trusted CA. It is also possible to configure the DBC Proxy so that it skips SSL client authentication altogether.

Internal and External Proxy Keys

The DBC Proxy uses a key pair (public key certificate and private key) called **External Proxy Keys** to establish and accept SSL connections on this link. For communication with applications located on hosts in the protected network, the DBC Proxy uses a key pair called **Internal Proxy Keys** to establish and accept SSL-secured connections (see figure 62).

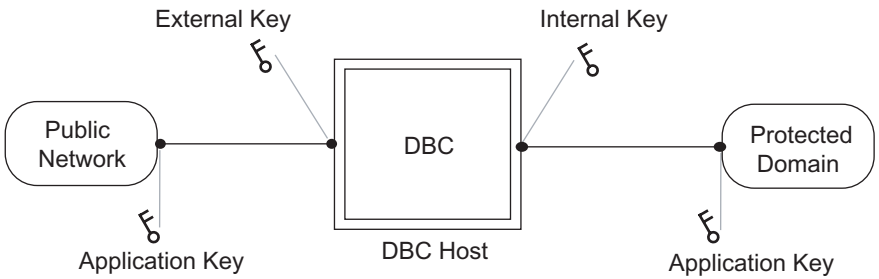


Fig. 62. Keys used on application connections

These keys are signed by the *ProxyCA* (cf. “Trust Stores and Trusted CAs” on page 203).

By default, the DBC Proxy is configured to use the *same* key pair both as External and Internal Proxy Key. For security reasons the public key certificate chain is stored in the `adm` directory on the DBC Proxy host (`ProxyChain.pem`), the corresponding private key is not stored on the DBC Proxy. It is stored on the Security Policy Server host verbatim in the configuration file `dbc.config` (located in the `adm` directory).

The SSL-related configuration options for the DBC Proxy – including the proxy keys – are set by defining and selecting *SSL Profiles* (cf. chapter "SSL and WS-Security Profiles" on page 165) for the internal or external interfaces.

Trusted CAs

The SSL profile that is selected determines not only the keys, but also the "trust store" used by the DBC Proxy when accepting SSL connections. Client certificates must be signed by one CA contained this trust store. By default this trust store is kept in the configuration file. Trusted CA certificates can be added to the trust store using the Admin Console (see "CA Certificates (Trusted CAs)" on page 214).

The table below lists all the keys, certificates and CA certificates that are used for application connections and where they can be found on the respective hosts.

Key	**File name**	**Host**
External/Internal DBC Proxy public key certificate chain	adm/ProxyChain.pem	Proxy
External/Internal DBC Proxy private key	Stored in the configuration file adm/dbc.config	SPS
List of public key certificates of trusted CAs for SSL Profiles	in configuration file or in a file (configurable with the AdminConsole)	Proxy
List of public key certificates of trusted CAs for WS-Security Profiles	adm/TrustedWSSECCAs.pem	Proxy
Self-signed certificate of the *ProxyCA*	adm/ProxyCACert.der, adm/ProxyCACert.pem	Proxy

Table 4. Default Keys and Certificates for application connections

All these keys are passphrase-protected with the passphrase `xtradyne`.

It is strongly recommended to set up your own *ProxyCA* for application connections and create new keys and certificates! This can be done with the Admin Console and is explained in detail in "Replacing External and Internal Proxy Keys" on page 211.

Making the DBC Proxy Trust External Certificates

The DBC Proxy will not accept any SSL certificates unless the certificate of the CA that signed those certificates is known to it. If you want the DBC Proxy to trust a certificate (which is usually not the peer user certificate but another CA), you have to add the certificate in PEM format to the trust store (i.e., to the file containing the trusted CA certifi-

cates). By default the DBC knows different trust stores for different kinds of SSL communication:

- The trust store for application connections is kept in the configuration and editable with the Admin Console (on how to do this, please refer to section "CA Certificates (Trusted CAs)" on page 214). Alternatively, it is located in a file in `sps/adm` (filename configurable with the Admin Console).
- `idbc/adm/TrustedWSSECCAs.pem:` Trust store for WS-Security Profiles.
- `sps/adm/TrustedLDAPCAs.pem`: Trust store for LDAP/SSL Profiles.
- `sps/adm/TrustedVAs.pem`: Trust store for validation authority files (for OCSP).

For the first bullet the following only applies when you choose to read the trust store from a file located in `sps/adm/`.

You can append a certificate to any of these trust stores with a text editor or by typing, for example:

```
cat myCACert.pem >> TrustedWSSECCAs.pem
```

If the certificate you want to add is not in PEM format, please consult section "Changing the Certificate Encoding Format" on page 216.

After adding a new CA certificate, the DBC Proxy or the SPS respectively has to be restarted in order to make the changes take effect (depending on whether the trust store is located on the Proxy or on the SPS host), see "Startup, Shutdown, and Restart" on page 82 on how to restart the DBC Proxy.

Integrating the DBC with Applications

To allow external software to authenticate the DBC Proxy when establishing secure connections using SSL, the CA certificate (`ProxyCACert.der` or `ProxyCACert.pem`) used to sign the DBC Proxy keys must be made available to external applications. In most cases you will have to configure the client application in a way that it regards the DBC Proxy's CA as trusted.

The keys of applications in the public and the protected domain are part of these applications and not of the DBC. Please refer to the documentation of these applications to for information about how to create and install application keys and certificates.

Client Keystores

If the client uses keystores, execute the following steps to establish mutual trust:

1. Export the client CA from the keystore:

Java's `keytool` command can be used to view the keys contained in the keystores:

```
keytool -list -keystore <keystore> -storepass <password>
```

You can export the client CA from the keystore with the following command:

```
keytool -keystore <keystore> -storepass <password>
        -export -alias <alias> -file <filename>
```

Each entry in the keystore has an alias, use the client CA's alias in the command above. The exported file will be in DER format.

2. Convert the exported file to PEM format:
   Use the script `der2pem.sh` located in the bin directories of the DBC Proxy and the Security Policy Server:

   ```
   der2pem.sh <DER_certificate> > <PEM_certificate>
   ```

3. Make it available to the DBC Proxy: Please refer to section "CA Certificates (Trusted CAs)" on page 214.
4. Import the DBC Proxy CA into the client keystore:

   ```
   keytool -keystore <keystore> -alias <alias> -import
           -file <certificate> -storepass <password>
   ```

   The DBC Proxy CA is by default located in the `adm` directory of the Proxy and stored in `ProxyCACert.pem`

10.2.3 Control and Administration Connections

The DBC relies on two additional, purely internal connections between the DBC components. The connection between the DBC Proxy and the Security Policy Server is called

Control Connection. The connection between the Security Policy Server and the Admin Console is called **Administration Connection** (see figure 63).

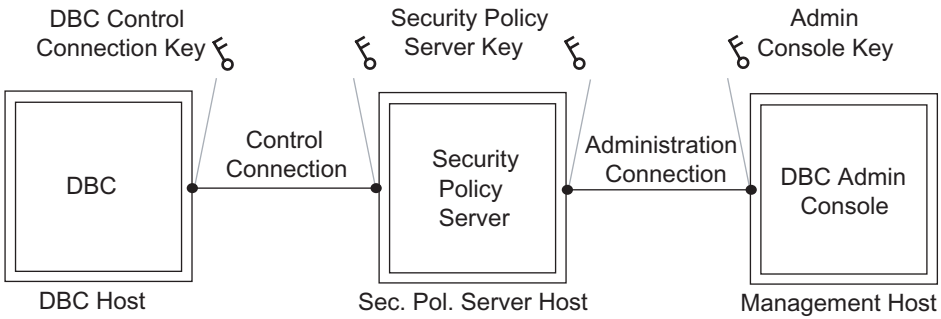


Fig. 63. Keys used on the Control and Administration Connection

On the control connection, the Security Policy Server uses the Security Policy Server keys for SSL-secured communication. These keys are also used on the administration connection between the Administration Console and the Policy Server. The Admin Console uses the Admin Console key on its side of the administration connection. The keys used on the control and admin connections are signed by the *ControlConnectionCA* (cf. “Trust Stores and Trusted CAs” on page 203).

The following table lists the different default keys and certificates that are used for the control and administration connection and where they are located on the respective hosts:

Key	File name	Host
Public key certificate of the DBC Proxy for the control connection	adm/ControlConnectionCert.der	Proxy
Private key of the DBC Proxy for the control connection	adm/ControlConnectionKey.der	Proxy
Self-signed certificate of the ControlConnectionCA	adm/ControlConnectionCACert.der, adm/ControlConnectionCACert.der, bin/ControlConnectionCACert.der	Proxy, SPS, Admin Host,

Table 5. Default Keys and Certificates for Control and Administration Connections

Key	File name	Host
Key certificate for LDAP SSL Profiles in PEM and in DER format.	adm/LDAPClientCert.der adm/LDAPClientKey.der adm/LDAPClientCert.pem adm/LDAPClientKey.pem	SPS
Public key certificate of the SPS	adm/SPSCert.der	SPS
Private key of the SPS	adm/SPSKey.der	SPS
List of public key certificates of trusted CAs for LDAP SSL Profiles	adm/TrustedLDAPCAs.pem	SPS
List of Validation Authority Files (for OCSP)	adm/TrustedVAs.pem	SPS
Public key certificate of the Admin Console	bin/AdminConsoleCert.der	Admin Host
Private key of the Admin Console	bin/AdminConsoleKey.der	Admin Host

Table 5. Default Keys and Certificates for Control and Administration Connections

All these keys are passphrase-protected with the passphrase `xtradyne`.

Note that additionally a bundle of example keys is created on the SPS host during the installation. These keys are signed by the DBC Proxy CA and may be used when testing the DBC by running a test application that does not provide keys and certificates.

Checking the Validity of Keys and Certificates

The keys and certificates generated during the installation process might not yet be valid, for example, due to a wrong time zone setting on your computer. In this case, you will get a “Server not reachable” exception when trying to login onto the SPS with the Admin Console.

To determine the validity dates of generated certificates, you can use the script `printcert.sh` located in the `bin` directory of the SPS and the Proxy.

Example

```
printcert.sh ../adm/SPSCert.der
```

The certificate will be printed in a readable form including the validity dates that look something like this:

```
Validity
   Not Before: Sep 16 09:41:56 2003 GMT
   Not After : Nov 22 09:41:56 2013 GMT
```

If the “Not Before date” lies in the future, the certificate is not yet valid (consider also the given time zone - in this example GMT!). In this case, correct the time settings on your computer and re-run the script that generates the keys and certificates (see section “Generating Keys” on page 90 for details).

10.3 Replacing Keys and Certificates

While installing the DBC a script creates all the necessary CAs, keys, and certificates according to the description in the sections before. You can test the DBC with these generated keys. Before operating the DBC in a production environment you should replace the keys for the application connections with trusted keys of your own.

10.3.1 Replacing External and Internal Proxy Keys

We assume that you have already obtained the keys you want to deploy with the DBC. The procedure of deployment is the following:

1. Define an SSL Profile that describes the properties for the new keys.
2. Select that SSL Profile for incoming or outgoing connections on the “Internal Interface” or “External Interface” panel respectively.

The rest of this section examines these steps more closely and walks you through an example.

Defining SSL Profiles

changing keys with the Admin Console

The Internal and External Proxy Keys and trusted certificates can be configured with the Admin Console. Both kinds of keys are selected using *SSL Profiles,* which were explained in detail in chapter “SSL and WS-Security Profiles” on page 165.

To define a new SSL Profile for a new key pair, go to the “SSL Profiles” panel. By default there are two SSL Profiles: The *SSLServer* profile is used for incoming connections (when the DBC Proxy is in the server role). The *SSLClient* profile is used for outgoing connections (when the DBC Proxy is in the client role).

Selecting SSL Profiles for Specific Interfaces

External Proxy Keys are chosen using the Admin Console by selecting an SSL Profile in the DBC Proxy configuration’s “Network Interfaces/External” panel. Internal Proxy Keys are selected in the “Network Interfaces/Internal” panel. On these panels, SSL Profiles are assigned to SSL Acceptors and SSL Connectors. Commonly, the same SSL Profile is used for both Acceptors and Connectors.

An Example SSL Profile

Let’s assume you want to define your own SSL Profile for **external** Acceptors. For these external communication endpoints you want to install a set of keys (private key, public key certificate, and certificate of a trusted CA). These keys correspond to the External Proxy Keys as shown in figure “Keys used on application connections” on page 204:

- The private key is called `MyCompanysPrivateKey.pem`.
- The certificate to this key is called `MyCompanysCert.pem`.
- The CA certificate is called `MyCompanysCACert.pem`.

There are two ways to make the DBC use these keys: The keys can either be stored on the DBC Proxy host and the filename can be provided, or the keys can be provided directly in the DBC's configuration file.

Storing the private key on the DBC Proxy host is a potential security risk if an intruder can get access to the firewall host system! Therefore storing the key directly in the configuration is the preferred choice.

The certificates may be stored on the DBC Proxy host. Copy these files to the directory `adm` on the DBC Proxy host. You should check if file permissions are set correctly. On how to do this, please refer to gray box "Checking Permissions for Key Files" on page 215.

To make the DBC Proxy recognize the new keys, start the Admin Console and edit the **SSLServer** Profile.

Protocol

You can leave the "Protocol" part of the SSLServer Profile as it is. When editing a new profile, you define the following:

- *Profile Name*: Define a name for your profile, for example, "SSLServer".
- *SSL Version*: Choose the SSL version; the recommended setting is "v23/TLS", for detailed explanations, please see "SSL Version" on page 167.
- *Ciphersuite*: Choose the set of ciphers to be used; the recommended value is "DEFAULT:!EXPORT:!aNULL", for details see "Ciphersuite" on page 168.
- *Peer Authentication*: Define the way client authentication will be done, for example, "Clients must have a valid certificate".

Key and Certificate

On the "Private Key and Certificate" tab of the "SSL Profiles" pane you define the properties of the private key and certificate for this profile (see also screenshot below):

- *Private Key File*: Not recommended! Click on the three dots on the right and give the location of the External Key (e.g., `MyCompanyPrivateKey.pem`). Relative path names are interpreted relative to the `adm` directory of the Proxy. We recommend to use absolute paths to avoid ambiguities.
- *Private Key Opaque*: Paste the private key (in PEM format) directly into the "Use Opaque Key" field (recommended choice). Please don't forget to include the lines: `-----PRIVATE KEY BEGIN----- ... -----PRIVATE KEY END-----`.

- *Use passphrase*: If the key is protected by a passphrase, click on “Use Passphrase” and give the passphrase.
- *Certificate File*: Click on the three dots on the right and choose the certificate corresponding to the External Key (e.g., `MyCompanyCert.pem`).
- *Certificate in Config*: Alternatively, the certificate chain can be stored directly in the configuration file. You can add certificates by choosing **Add...** from the context menu and then selecting the file containing the certificate. The Subject, Issuer, and Expiration Date of the certificated will be displayed in the table.

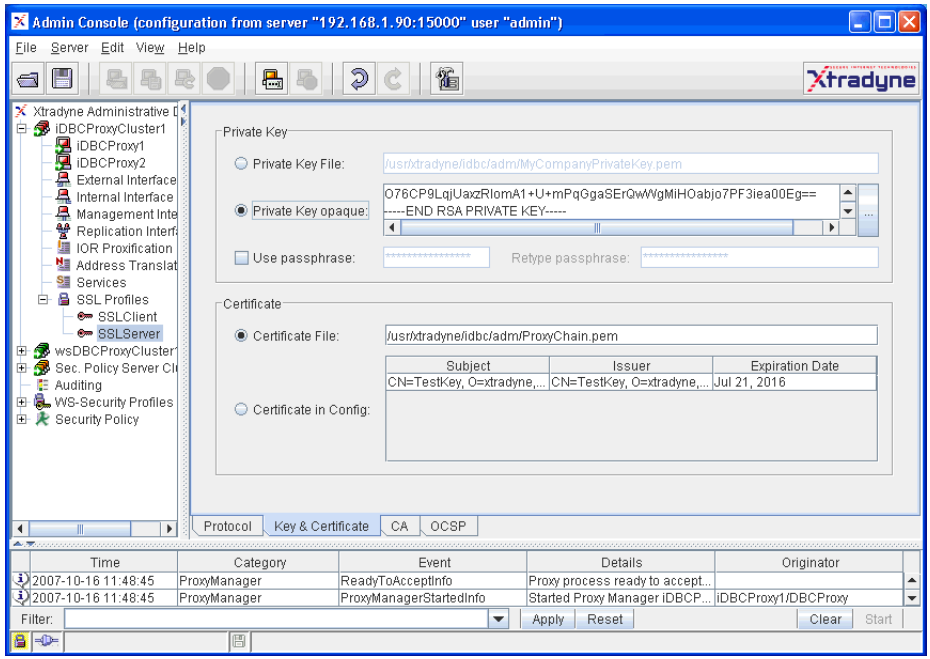


Fig. 64. Configuring an SSL Profile

Note that selecting a file using the file selector which appears after clicking “...” will only work if the local path is available on the DBC Proxy host also. This is generally true only for single host installations.

CA Certificates (Trusted CAs)

Finally, on the **DBC Proxy ➜ SSLProfiles ➜ SSLServer** page "CA" tab, you define the trust store for application connections. This trust store has to contain all required CA certificates, i.e., CA certificates that sign your trusted peer's certificates (for example the CA certificate of the client application).

With the Admin Console, you can configure whether the CA certificates shall be read from a given file or shall be stored directly in the configuration (the preferred way, described in the second bullet):

- *From File*: The location of the file containing the trusted CA certificates can be defined here. The value will be taken relative to the Proxy's `adm` directory. On how to add certificates to the file, see below.
- *From Configuration*: The trusted CA certificates are stored directly in the configuration file. You can add certificates by choosing **Add...** from the context menu and then selecting the file containing the certificate. The Subject, Issuer, and Expiration Date of the certificated will be displayed in the table.

Adding certificates to the file trust store

If you would like read the truststore from a file, go to the sps/adm directory and create a new file named for example `TrustedSSLCAs.pem` (the filename you entered in the Admin Console before). Then append the certificate in PEM format to the file. This can be done with a text editor or by typing:

```
cat <certificate file> >> <DBC Proxy Trusted CAs file>
```

For Example:

```
cat myCACert.pem >> TrustedSSLCAs.pem
```

Note that you have to restart the DBC Proxy after having added the CA certificate to the file, so that the DBC will recognize the new trusted certificate. See "Startup, Shutdown, and Restart" on page 82.

10.3.2 Defining WS-Security Profiles

WS-Security Profiles are only of interest when operating a **WS-DBC**. They define the keys and certificates for signing and verifying XML Digital Signatures. Defining a WS-Security Profile works basically the same as defining an SSL Profile (see previous section).

A difference to SSL Profiles is that the certificate file must contain a certificate chain. The WS-DBC Proxy will include the content of this file within the digital signature (when it is created by the WS-DBC). The receiver can use this information for verification. Also note that the WS-DBC will use SHA1 as digest algorithm and RSA-SHA1 as signature algorithm by default.

Checking Permissions for Key Files

You should check if permissions for keys are set correctly. Only the DBC installation user (`xtradyne` by default) should have read and write permissions. Find out permissions with the command `ls -la` in the directory `adm` of the SPS and the Proxy respectively, for example:

```
root@myhost:/usr/xtradyne/sps/adm > ls -la *.der
-rw------- 1 xtradyne users 592 Jul 9 17:07 ControlConnectionCACert.der
-rw------- 1 xtradyne users 654 Jul 9 17:07 SPSCert.der
-rw------- 1 xtradyne users 677 Jul 9 17:07 SPSKey.der
```

The first column of output is the important one. Read and write permissions (rw) should only be set for the user (the second to fourth position apply to the user), for group (next three) and all (last three) neither read nor write permissions should be allowed (this is indicated by the dashes)!

If permissions are not ok, use

```
chmod 600 *.der *.pem
```

Keys must be owned by the DBC installation user (`xtradyne` by default). If this is not the case type, for example (as `root`):

```
chown xtradyne *.der *.pem
```

10.3.3 Replacing Control and Administration Connection Keys

It is not necessary to replace the Control Connection and the Administration Client Key, which are used by the DBC Proxy and the Admin Console respectively to establish trust with the Security Policy Server. These keys are generated at install time. If you want to replace these keys anyway, please contact Xtradyne's professional services.

10.3.4 Changing the Certificate Encoding Format

If your applications only provide keys and certificates in DER format, you can convert these to PEM encoding using the shell script `der2pem.sh` included in the DBC distribution. The script prints the converted file to the console, so you will have to redirect the output to a file. To convert a DER-encoded file do the following:

Copy the file in DER format into the `adm` directory of the Proxy and call the script that converts it, for example:

```
~: cd adm
adm/: cp /usr/examples/Certificate.der .
adm/: ../bin/der2pem.sh Certificate.der > Certificate.pem
```

To make the DBC Proxy trust a given CA, you need to add its public key certificate in PEM format to the file of trusted CAs. By default, this file is `TrustedSSLCAs.pem` and located in the `adm` folder on the DBC Proxy host. The file name that points to this trust store is configured on the “SSL Profiles” panel in the “CA Certificates (Trusted CAs)” field. For details see section “CA Certificates (Trusted CAs)” on page 214.

CHAPTER

11 Troubleshooting

The following sections list frequently encountered problems and help to determine why something went wrong. Please consult this chapter before contacting customer support (see page 22 in the Preface for contact information).

11.1 Encrypting the DBC Configuration File for Support

When you are asked to send the DBC configuration file to the PrismTech support team (contact details see page 22) you may use the **File ➜ Export ➜ Encrypt for support...** facility of the Admin Console. This will encrypt confidential information contained in the config file, i.e., keys, certificates, and passwords. Encryption is done by a password of your own choice.

When receiving back the configuration file from PrismTech support it may be re-imported with the Admin Console using the **File ➜ Import ➜ Decrypt from support...** function.

11.2 How to Diagnose Problems: Logging

If anything goes wrong take a look at the log file(s) and the event output. Additionally, service clients may receive CORBA System Exceptions (when operating an **I-DBC** Proxy) or HTTP or SOAP Error Messages (when operating a **WS-DBC**). In summary: To diagnose problems take a look at:

- **Events:** are either logged by `syslog` or sent to an external (user defined) command according to the audit policy, cf. “Audit Policy” on page 189. DBC audit events are listed in appendix A, “Audit Events” on page 321. Events can be viewed with the Admin Console’s event browser, cf. “Audit Event Browser” on page 119.
- **Error Messages:**

- DBC and SPS error messages: By default DBC and SPS error messages are written to the `adm` directory into the file `sps.log` on the SPS host and on the DBC Proxy host into the file `dbc.log`. A list of DBC error messages can be found in Appendix B, section “DBC and SPS Error Messages” on page 339.
- CORBA system exceptions (**I-DBC** only): A list of CORBA system exceptions and minor codes received by service clients can be found in Appendix B, section “CORBA System Exceptions and Minor Codes” on page 344.
- HTTP Error Messages (**WS-DBC** only): Appendix B, section “HTTP Error Messages” on page 353 lists and explains HTTP Error Messages received by service clients.
- SOAP Error Messages (**WS-DBC** only): Appendix B, section “SOAP Error Messages” on page 353 lists and explains SOAP Error Messages.

11.2.1 Determining the DBC Proxy / SPS Status

One of the first steps to take when something goes wrong is to check whether the I-DBC Proxy or WS-DBC Proxy respectively and the Security Policy Server are up and running. Their status can be determined with the scripts `xdn_sps`, `xdn_idbc`, and `xdn_wsdbc` with the option `status` (you must be root to execute this):

```
/etc/init.d/xdn_sps status
/etc/init.d/xdn_idbc status
/etc/init.d/xdn_wsdbc status
```

The scripts yield “OK” if the Security Policy Server/DBC Proxy is up and running. If not, you will get “no process”. Note that the SPS and the Proxy will start up but refuse to work until a valid license file is installed (in this case, the status scripts will yield “OK” too, but you will see an error message reading “error loading license FileOpenException(0): could not open file: license.txt).” in the log file (located in `<INSTALLDIR>/sps/adm/sps.log` on the Security Policy Server host and in `dbc.log` in the `adm` directory on the DBC Proxy host).

Useful Status Scripts

For more information about the status of the SPS and the I-DBC Proxy or WS-DBC Proxy respectively like the number of Proxy Processes, the ports of the DBC Proxy and the Security Policy Server, and their current connections you can use the scripts:

```
<INSTALLDIR>/idbc/bin/checkproxy.sh
<INSTALLDIR>/wsdbc/bin/checkproxy.sh
<INSTALLDIR>/sps/bin/checksps.sh
```

Example

```
~:/usr/xtradyne/idbc/bin/checkproxy.sh
proxy ...           2 processes/threads
proxymanager ...   12 processes/threads
Listening on
tcp  0  0  192.168.1.90:7384  0.0.0.0:*  LISTEN  31038/proxy
tcp  0  0  192.168.1.90:8885  0.0.0.0:*  LISTEN  31038/proxy
tcp  0  0  0.0.0.0:15000      0.0.0.0:*  LISTEN  31038/proxy
Current connections:
tcp 0  0 192.168.1.90:2316 192.168.1.90:15000ESTABLISHED31038/proxy
tcp 0  0 192.168.1.90:15000192.168.1.90:2315 ESTABLISHED31038/proxy
```

In this example an I-DBC Proxy is listening on the ports 7384 (external plain IIOP listener), 8885 (external IIOP/SSL listener) and 15000 (listener port for the connection to the Security Policy Server). The I-DBC Proxy has currently two connections: one from the Security Policy Server and one to the Security Policy Server.

Note that `checksps.sh` yields information not only about the Security Policy Server(s) but also about the DBC Proxy in your installation. If you start, for example, a Security Policy Server but do not start the DBC Proxies and then call the script, the output will read like this:

```
SecurityServer ... 2 processes/threads
dbcCluster1:myhost1      ... not started
dbcCluster1:myhost2      ... not started
EMERGENCY: All DBCs down!
```

The script will also indicate when some of the DBC Proxies in a cluster could not be started. If everything is up and running the exit status of the script is 0.

11.3 Scripts Do Not Work

Some scripts included in the DBC installation like `proxyconfig.sh`, `checkproxy.sh` and `checksps.sh` try to switch to the user `xtradyne` before actually running. This will only work if the home directory of this user exists. At install time the user `xtradyne` is created if it doesn't exist yet. The home directory of this user is the installation directory (defaults to `/usr/xtradyne/` on Linux and `/opt/xtradyne/` on Solaris). When de-installing the user `xtradyne` will not be deleted. If you intend to upgrade to a newer DBC version and don't want to use the default directory for installation, please make sure to delete the user `xtradyne` before re-installing. Otherwise the user `xtradyne` will have the wrong (possibly non existing) home directory.

11.3.1 Checking Permissions for Key Files

After installing your own keys you should check if permissions for keys are set correctly. Only the user `xtradyne` should have read and write permissions. Determine permissions with the command `ls -la` in the `adm` directories of the SPS and the Proxy, for example:

```
root@myhost:/usr/xtradyne/sps/adm > ls -la *.der *.pem
-rw------- 1 xtradyne users 592 Jul 9 17:07 ControlConnectionCACert.der
-rw------- 1 xtradyne users 654 Jul 9 17:07 SPSCert.der
-rw------- 1 xtradyne users 677 Jul 9 17:07 SPSKey.der
```

The first column of output is the important one. Read and write permissions (rw) should only be set for the user (the second to fourth position apply to the user), for group (next three), and others (last three) neither read nor write permissions should be allowed (this is indicated by the dashes)!

If permissions are not ok, execute (as user `root` or `xtradyne`)

```
chmod 600 *.der.*.pem
```

Keys must be owned by the user `xtradyne`. If this is not the case type (as user `root`):

```
chown xtradyne *.der *.pem
```

11.4 When to Restart the DBC Proxy / SPS

If changes to the configuration you made with the Admin Console do not seem to take effect, the DBC Proxy or the Security Policy Server may have to be restarted. Note that the Admin Console will generally indicate when a restart is required. For a summary of when a restart is required, please refer also to the table on page 120.

11.5 Frequently Encountered Problems

Access Denied

If client access to a resource is denied the client will receive the CORBA System Exception “NO_PERMISSION” (I-DBC) or the SOAP Error Message: “Access Denied” (WS-DBC). This means that authorization and/or protection requirements were not met or that the client has insufficient or wrong credentials. To determine the exact cause, please consult the reason in the *ADFRequestDeniedFailure* event (see also page 332).

11.5.1 Internal Server Error

For any errors related to misconfigurations of the **WS-DBC** Proxy the client will receive the following error message: “Internal Server Error. Please contact the server administrator. Reference the message id:<id>”. In this case please consult the event log of the Proxy to determine the error reason. The message id in the client’s error message can be used to identify the event that describes the failure reason.

11.6 *SSL Connection Problems*

If the client gets the message “Error opening socket”, please make sure that all keys and certificates are valid and installed correctly. Please refer also to section “Installing Keys and Certificates” on page 201. The following sections list the most common events indicating problems with SSL and what can be done about them. Please refer also to appendix A, “Audit Events” on page 321 ff.

SSLAuthenticationCertificateFailure

The user’s certificate is trusted and valid but the DBC failed to recognize a client’s SSL certificate. This happens when the user is not known to the DBC. Start the Admin Console, go to the “Security Policy – Users” panel and check that the user is included in the list (see section “User Properties – General” on page 242 for details). Also check if the DN (Distinguished Name) in the certificate corresponds to the one given for this user (use `printcert.sh` in the `bin` directory of the Proxy to determine the DN).

SSLTransportHandshakeFailure

The DBC Proxy detected an error while in SSL handshake mode (a phase of the SSL protocol). This can have one of the following reasons:

- “unknown protocol”: The client tries to connect with plain TCP to an SSL listener. In this case, the client should use SSL, or a plain TCP acceptor should be configured using the Admin Console (on the “External Interface” panel).
- “peer did not return a certificate”: The client uses certificates that are not trusted by the DBC. During SSL handshake the DBC Proxy sends a list of DNs of trusted CA certificates to the client. The client sees that his certificate will not be trusted by the DBC and doesn’t return a certificate. Please append the client’s CA certificate

to the DBC Proxy's file of trusted CA certificates, see section "Making the DBC Proxy Trust External Certificates" on page 205 for details.

- "self signed certificate in chain": The client uses certificates that are not trusted by the DBC. During SSL handshake the DBC Proxy sends a list of DNs of trusted CA certificates to the client. Although the client has no valid certificate it returns a certificate chain. Please add the client's CA certificate to the "trusted CA Certificates" file, see section "Making the DBC Proxy Trust External Certificates" on page 205 for details.
- "sslv3 alert certificate unknown": The client does not trust the DBC Proxy CA certificate. Please consult the manual of the client application on how to make the client trust the DBC Proxy CA certificate.
- "alert bad certificate at client": The communication partners use incompatible SSL versions. If you think that this is the reason for the handshake failure, please contact customer support.

SSLTransportCertificateFailure

This event hints to an expired or not yet valid certificate. To determine the validity dates of the client's certificate you can use `printcert.sh` located in the `bin` directory of the SPS and the Proxy:

```
printcert.sh ../adm/SPSCert.der
```

Alternatively, you can use a tool like `openssl`, for example:

```
openssl x509 -in <certificate_file> -inform DER -text
```

11.7 Callback Configuration

Callbacks are only relevant in the context of an **I-DBC** installation. Callbacks can be configured in different ways. If you have problems with making callbacks work, we recommend to first configure callbacks the simplest way which is also the most insecure mode of operation and then to secure it step by step:

- Allow outgoing IIOP or IIOP/SSL connections on the "External Interface" panel.
- Enable IIOP or IIOP/SSL acceptors on the "Internal Interface" panel.
- Check the "Allow unknown clients" in the "Access Session Management" part of the "I-DBC Proxy Cluster" panel. If your application works now proceed to the next bullet, if you get a COMM_FAILURE, check the first two bullets again.
- Additionally, check the "Separate Access Sessions (AS) for unknown clients" box on the same panel, define a callback resource XDN:Callback:1.0 (on the "Security

Policy – Resource" panel) and allow "PUBLIC" access to this resource. For configuration details, see "Configuring Permissions for Callbacks" on page 282. If your application still works proceed to the next bullet. If you get a NO_PERMISSION exception, check again if you allowed "PUBLIC" access to the XDN:Callback:1.0 resource.

- Disable the "Allow unknown clients" in the "Access Session Management" part of the "I-DBC Proxy Cluster" panel. This requires the definition of a user with the correct permissions (see next bullet).
- Define proper permissions allowing only your CORBA server callback access: Add a new user with ip-based authentication ("User Properties – Authentication Methods"). Configure the server's IP address. For several servers using callbacks, configure a netmask to map the servers to your "server" identity (see also "Configuring Permissions for Callbacks" on page 282). As a last step go to the "User Properties – Privileges" panel and add the resource "XDN:Callback:1.0" to the user's privileges. Rerun your application. A NO_PERMISSION exceptions means that permissions are not yet correct. Consult the event log to diagnose the problem, e.g., an `AuthenticationIPBasedAuthenticationFailure` event indicates that the user-to-IP-address mapping failed, in this case, check if the server's IP address is correct.

11.8 License Errors

If the DBC Proxy or the Security Policy Server refuse to work, you might have forgotten to install the license:

On the host where you installed the DBC Proxy copy the license file to the directory `<INSTALLDIR>/idbc/adm/license.txt` (I-DBC), or to the directory `<INSTALLDIR>/wsdbc/adm/license.txt` (WS-DBC).

On the host where you installed the Security Policy Server copy the license file to the directory `<INSTALLDIR>/sps/adm/license.txt`.

Another reason might be that the license expired (check for the *LicenseManagerLicenseExpiredFailure* event in the event log). You can get a new evaluation license by contacting your customer support representative.

11.9 Problems with the Installers

11.9.1 Installer Exits with Exception (Linux/Solaris)

If the installer exits with an exception this might be due to the fact that no X is installed on the host where you started the installer. The installer can run without X, in a console mode. Start the installer with the parameter `-i console` to make it run in console mode. But node that the Admin Console is a graphical user interface requiring X. The DBC and SPS do not require X.

11.9.2 Installer does not start

If the installer for the Admin Console or the DBC does not start, this might be due to one of the following reasons:

Access Control on the X Display

```
Invocation of this Java Application has caused an Invoca-
tionTargetException.
```

The user who started the X server has activated access control on the X display. Typically, this applies if you use a remote X display. To grant access the owner of the X server needs to type the following on the X server host:

```
xhost + <hostname>
```

where `<hostname>` is the name of the host where you want to start the installer.

Included VM Could Not be Unarchived

```
Preparing to install...
The included VM could not be unarchived (tar). Please try
to download the installer again and make sure that you
download using 'binary' mode. Please do not attempt to
install this currently downloaded copy.
```

The problem may occur on Linux and Solaris systems. We have identified the following possible reasons for this problem:

1. The error message may occur because the path `/usr/ucb/ls` is prior to the path `/usr/bin/ls` in your path settings. To check the path settings type:

   ```
   type ls
   ```

If the output is `/usr/ucb/ls` you need to change the PATH variable setting of the login shell. To change this setting type:

```
PATH=/usr/bin:$PATH

export PATH
```

2. On some Redhat 7.1 systems the `uncompress` command is missing if a minimal system has been installed. To check for existence of the `uncompress` command type:

   ```
   uncompress
   ```

   If the output is something like “uncompress: command not found” you need to install the uncompress tool on your system. Alternatively, you can also use `gunzip`. To use `gunzip` instead, add a symbolic link as follows (as user `root`):

   ```
   cd /bin
   ln -s gunzip uncompress
   ```

3. Your system may be short on temporary file space. Make sure that there is at least 200 MB of free disk space for temporary files. To check the available disk space type:

   ```
   df -k -h /tmp
   ```

4. If the problem remains, please contact customer support.

JAVA_FONTS Variable Not Set Correctly

If the environment variable JAVA_FONTS is set to an empty directory, the installer extracts the JRE but will not start. Determine if the JAVA_FONTS variable is set with:

```
echo $JAVA_FONTS
```

Unset the variable with:

```
unset JAVA_FONTS
```

Restart the installer.

11.10 Admin Console

11.10.1 Problems With Starting the Admin Console

If the Admin Console does not start and the command does not return nor is anything displayed: If you have installed the Admin Console on Solaris and redirected the display to a Linux screen, this might happen if you are running your X with 16bit color depth (8,

24 and 32 work fine). The easiest solution is to install the Admin Console on the Linux box, too.

11.10.2 Linux, Solaris Startup

Before starting the Admin Console, make sure that the DISPLAY environment variable is set correctly. When using `bash` as shell, type: `export DISPLAY=<host>:0`. When using `sh` as shell type: `DISPLAY=<host>:0; export DISPLAY`.

Otherwise you may get the following error:

```
Exception occurred in main()
java.lang.InternalError:Can't connect to X11 window server
```

11.10.3 Problems With Logging On to the SPS

Server not reachable (`org.omg.CORBA.TRANSIENT`)

This error message can have two reasons:

- Faulty SSL configuration:
  Check the validity dates of the installed keys and certificates (cf. gray box page 210). If you installed your own Admin Console keys and certificates, please check whether you configured everything as described in chapter "Installing Keys and Certificates" on page 201. Switch off authentication and try to connect with SSL encryption only (Configure this on the Admin Console's **Edit ➜ Preferences** panel). In this mode no keys and certificates are needed.
- Addressing information is not configured correctly:
  The server cannot be reached with the used network address: Please check the host and the port number in the "Properties" dialog. Use a colon to separate the host from the port number. To guarantee that the address settings are right and the server can be reached, please try to establish a raw SSL connection without client and server authentication. If this works, but the error pertains when using authentication, it is likely to be an SSL problem (see first bullet).

Unfortunately, there is no way to distinguish a faulty SSL configuration from a pure addressing misconfiguration. For advanced diagnosis, please contact technical support.

Server not reachable (`org.omg.CORBA.COMM_FAILURE`)

This error occurs when trying to connect via SSL to the Security Policy Server but the Security Policy Server only offers a TCP listener as management interface (or vice versa). Remember that the DBC Proxy and the Admin Console share one listener to contact the Security Policy Server. There a two ways to fix this error:

- turn off SSL (not recommended!)
- reconfigure the Security Policy Server and the DBC Proxy(!) so that they use SSL for their control connection.

For details on changing the interfaces, see section “Initial Configuration of the Management Network Interface” on page 89 and section “Initial Configuration of the Security Policy Server Interface” on page 96.

User has no access (`org.omg.CORBA.NO_PERMISSION`)

The standard user ID/password authentication failed. Please check the correctness of the user ID/password combination. For details on how to administrate password settings, see section “Changing the Admin User’s Password” on page 245.

11.10.4 Problems with Adding SSL Certificates

The Admin Console cannot handle the case when a certificate (in PEM Format) additionally contains certificate info as plain text. In this case, the Admin Console will show an Error Dialog (`java.security.cert.CertificateParsingException`) when you try to add such a certificate to the configuration (e.g. in an SSL Profile or a WS-Security Profile).

11.11 *Miscellaneous*

11.11.1 Firewall Configuration – TCP Connection Timeouts

A common configuration of firewalls includes a timeout for TCP connection idle for longer than, say, an hour. This is intended to prevent dead connections from existing in the firewall state tables for extended periods. TCP idle timeouts present a problem in DBC usage scenarios where connections are unused for longer than the timeout period. This happens because most ORBs do not timeout idle connections neither does the DBC in its standard configuration.

When the connection is to be used again by the first person in the morning, for example, an application error will occur. To prevent this, there are several possibilities:

- Make the DBC Proxy timeout TCP connection itself. This will totally prevent idle TCP connections and is probably the best solution. Start the Admin Console, go to the “DBC Proxy” panel and configure the “GIOP idle connection timeout”.
- Reconfigure the firewall to not use TCP idle connection timeouts. This might not be an option due to security policy restrictions and is not recommended.
- A third option is to reconfigure the DBC Proxy (and probably the client ORB) to use TCP keep-alives. You can activate TCP keep-alives with the Admin Console. Every IIOP Listener has a details panel where the sending of TCP keep-alives can be activated.
  TCP keep-alives are usually sent every two hours on idle connections, so the TCP idle connection timeout on the firewall must be greater than two hours for this option to take effect. In addition, it is possible to change the TCP keep-alive interval on operating system level (how to do this, please see section below).

When none of the above options can be applied change the application to gracefully recover when getting a COMM_FAILURE exception on a remote call. Simply repeating the call a single time is the recommended procedure in this case.

Please make sure that the firewall, if it uses TCP idle connection timeouts, does never silently drop packets on timed out connections as this will lead to application hangs until the client side TCP times out, which can take several minutes. Meanwhile, the application will be not responding, seeming to have crashed.

Finding out and setting TCP keep-alive times

On Linux

Determine the TCP keep-alive time (in seconds) on your system, type:

```
cat /proc/sys/net/ipv4/tcp_keepalive_time
```

To set a different TCP keep-alive time (in seconds), type:

```
echo <sec> > /proc/sys/net/ipv4/tcp_keepalive_time
```

On Solaris

Determine the TCP keep-alive time (in milliseconds) on your system:

```
ndd /dev/tcp tcp_keepalive_interval
```

To change this value use ndd with the `-set` flag, e.g.:

```
ndd -set /dev/tcp tcp_keepalive_interval 1200000
```

11.12 Using Logrotate on Solaris causes sparse DBC log files

When Logrotate is used on Solaris to rotate DBC log files this may result in sparse files, i.e., the log files are not truncated properly, but filled with null bytes. The problem is that the shell (`sh`) does not implement the append file mode for output redirection but seeks for end of file. This will result in sparse files during truncate as the log files remain opened for writing by the DBC processes.

Workaround

Use `bash` instead of `sh` to execute DBC wrapper scripts. Make sure that you have installed `bash` on your system. The default installation path for bash is `/usr/bin`. If `bash` is not installed on your system and it is not provided with the Solaris distribution you can download it from `http://www.sunfreeware.com/`.

To associate the DBC wrapper scripts with bash you need to change the first line in the following files:

For the I-DBC Proxy:

- `<INSTALLDIR>/bin/xdn_idbc`
- `<INSTALLDIR>/bin/runiproxy`

For the SPS:

- `<INSTALLDIR>/bin/xdn_sps`
- `<INSTALLDIR>/bin/runsps`

For WS-DBC Proxy:

- `<INSTALLDIR>/bin/xdn_wsdbc`
- `<INSTALLDIR>/bin/runwsproxy`

Replace the first line:

```
#!/bin/sh
```

with

```
#!/usr/bin/bash --posix
```

11.13 My CORBA Application Doesn't Run With the I-DBC

This section is only relevant when troubleshooting the operation of an **I-DBC**.

System Exception 'NO_RESOURCES'

Some ORBs (for example, ORBacus) throw a "NO_RESOURCES" exception if given an IOR with both SSL and TCP profiles, but the application is configured only for TCP connections. An IOR that is exported by the Admin Console always contains information for all configured listeners. If you specify a TCP and an SSL listener, the IOR will contain a TCP and an SSL profile. If you come across such a problem with your ORB, please remove the SSL profile from the proxified IOR executing the following:

```
cat <IOR.ref> | <INSTALLDIR>/sps/bin/proxifyIOR
    -h <DBCHost> -p <DBC_TCP_Port> >
    <IORWithoutSSLProfile.ref>
```

`printIOR` can be used to view the profiles contained in an IOR (please see also example on page 231).

No Listener Available for IOR Proxification

When importing an IOR with the Admin Console you will see that SSL is enabled in the proxified IOR components. If you don't set up an IIOP/SSL Acceptor on the "External Interface" pane and try to save this configuration, the Admin Console will come up with the following message: "Configuration has not been written. No listener available for IOR proxification (wants SSL but no SSL ICP available)." Set up an IIOP/SSL Listener or run your CORBA client without SSL (not recommended!).

Communication Failures

```
org.omg.CORBA.COMM_FAILURE: minor code: 1330446336 completed: No
```

There are two possible reasons for this message:

- Wrong set up of IIOP listeners: Did you activate the SSL listener in the "Internal Interface" panel but started the application server without SSL? Activate the plain IIOP listener on the "Internal Interface" panel or start the server with SSL support.
- The application's SSL inserts empty fragments: Upgrade to newer IAIK/SSL version if possible. If the problem persists, please call customer support.

No Permission

If "Use Access Control" is enabled but the client doesn't use SSL at all, you will get "NO PERMISSION".

Check if an event called *ADFRequestDeniedFailure* occurred. If the event consumer is syslog, execute (as root): `less /var/log/messages`. If `less` is not installed try using `tail` and `grep` then. Go to the end of the messages file with **Shift-G** and look for *ADFRequestDeniedFailure*. Disable access control in the "I-DBC Proxy Cluster Properties" panel of the Admin Console and restart the I-DBC Proxy.

No Connection

Use `printIOR` to check the IOR your client uses. For example:

```
<INSTALLDIR>/idbc/bin/printIOR Bank.proxi.ref
```

Check if the host name is correct. If you see something like `host.domain.example`, you forgot to configure the I-DBC Proxy host name. Configure this on the "I-DBC Proxy Properties" pane. Go to the "Initial IOR Table" panel and export the IOR again.

The Application Doesn't React At All

If your application doesn't react at all, this might be due to wrongly configured listeners. For example if you changed the listener port and forgot to re-export the proxified IOR. In this case, check if the port number in the proxified IOR is the correct one. You can use the tool `printIOR` which gives you some information about the values coded in the

proxified IOR. For example the proxified IOR of the Frankfurter Bank example application would look like this:

```
:~/FrankfurterBank; printIOR Bank.prox.ref
Read from file Bank.prox.ref:
IOR:
Byte order: (1) Little Endian
TypeId: IDL:com/xtradyne/frankfurterbank/Account:1.0
Number of Tagged Profiles: 3
[1]   IIOP Profile
   Byte order: (1) Little Endian
   Version: 1.1
   Host: 192.168.47.11
   Port: 8885
   ObjectKey:
    15 byte(s)
    43 43 49 6D 70 6C 2F 43 43 50 4F 41 2F 43 43  FBPoa/FBPoa/FB.
   Number of Tagged Components: 1
   [1]   SSL_SEC_TRANS Component (componentId=0x14)
      Byte order: (1) Little Endian
      Target supports: 126 - Integrity & Confidentiality &
DetectReplay & DetectMi
sordering & EstablishTrustInTarget & EstablishTrustInClient
ordering
      Port: 8885
corbaloc::1.1@192.168.1.90:8885/FBPOA/FBPOA/FB
```

PART

3 *MANAGING SECURITY POLICIES AND DEPLOYING WEB SERVICES*

The WS-DBC respects three different kinds of security policies: authentication policies, message analysis and protection policies, and access control policies. The first chapter of this part explains how security policies are defined using the Admin Console, with special emphasis on access control concepts. Chapter 2 describes the rules for defining parameter filters for the WS-DBC and finally, chapter 3 "Deploying Web Services – An Example" on page 305 gives a detailed example on how to define security policies with the Admin Console and how to protect an existing Web Service with the WS-DBC.

CHAPTER

1 Security Policies

Access control as performed by the DBC follows the concept of Role-Based Access Control (RBAC). The first section of this chapter gives a general introduction to this concept. The subsequent sections explain how to use the Admin Console to define security policies. Additional details are explained along with an example in the following chapters.

1.1 Access Control

Access Control must consider both an application and its users. We divide the description of Access Control information in two parts:

- the application side, where the rights to send operations to Services are defined and combined into roles, and
- the user management side, where users can be grouped in teams, departments, etc., and assigned to roles.

This distinction is based on the Enterprise Java Beans (EJB) separation of administrative concerns between the application developing domain (Bean Provider, Application Assembler), the production environment (System Administration), and the Deployer who links the two domains.

1.1.1 Access Control Policy

The access control policy comprises four components: Users, Groups, Roles, and Resources. Each component has a different function in the model. Figure 1, “Components of the access control policy”, depicts the relationship between Users, Groups, Roles, and Resources.

Resources, Roles

Resources represent Services that are identified by an URL – in case of Web Services – or an Object Reference (IOR) – in case of CORBA Services. A Resource has accessors, i.e., one or more Roles (see association ⑤ in Figure 1). Generally, a Role can access a Resource by sending operations. Roles represent real-life tasks and receive the necessary *permissions* to invoke *operations* on Resources to fulfill these tasks. Roles can contain Subroles to model different levels of abstraction (e.g., OracleManager – DepartmentManager). If a role Y has another role X as actor (association ④ in Figure 1) role X inherits Y's permissions. Hence, Y can be viewed as "junior" to X as X has all permissions Y has, but may have additional permissions of its own.

Users, Groups

Users can be grouped (e.g., in teams, departments, etc.) by the System Administrator. *Groups* can contain other Groups as members ②, e.g., three teams, a secretary, and a management person form a task force. Vice versa, the task force group has five members: the management person, the secretary, and the three teams, which are again Groups.

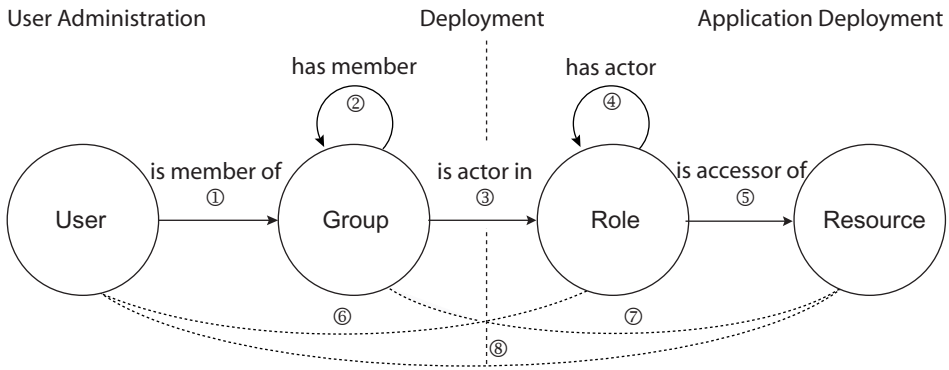


Fig. 1. Components of the access control policy

Roles are accessors of Resources ⑤, i.e., they receive permissions from Roles. Roles propagate these permissions to other Roles ④ or Groups ③. Groups pass those on to their members which can be either other Groups ② or individual Users ①.

Role Engineering

We recommend that Users are not directly granted access to a Resource ⑧, but are assigned to Groups ① that receive their permissions from Roles. This approach encourages you – prior to assigning anything with the Admin Console – to think about the abstract structure behind your policy. The full advantages of Role-Based Access Control (especially the reduced maintenance cost) can only be achieved by careful role-engineering. For flexibility, "short-cuts" are allowed (dotted lines in Figure 1), but we recommend that they are used only sparingly!

1.2 Defining Security Policies

The following sections describe how to use the Admin Console to define the following policies:

- The *access control policy* defines which user has access to which resource (either for the whole Service or, optionally, for individual RPC operations offered by the Service). Access control in the DBC uses the concept of Role-Based Access Control as described in the previous section.
- The *authentication policy* defines how the client must authenticate.
- The *message protection policy* defines how a message will be cryptographically protected to ensure message integrity and secrecy.
- The *content inspection policy* defines filter rules against which message parameters will be checked. In the **WS-DBC** this policy additionally includes whether and against which schemas SOAP messages will be validated.

setting up and maintaining access control policies

The Admin Console helps you to setup and maintain your access control policies. If a security policy has already been saved on the SPS, retrieve the authorization information from the server by clicking **File➔Load from Server**. When the loading is completed click on the ⊞ next to the security policy icon on the left side of the Admin Console to browse the policy.

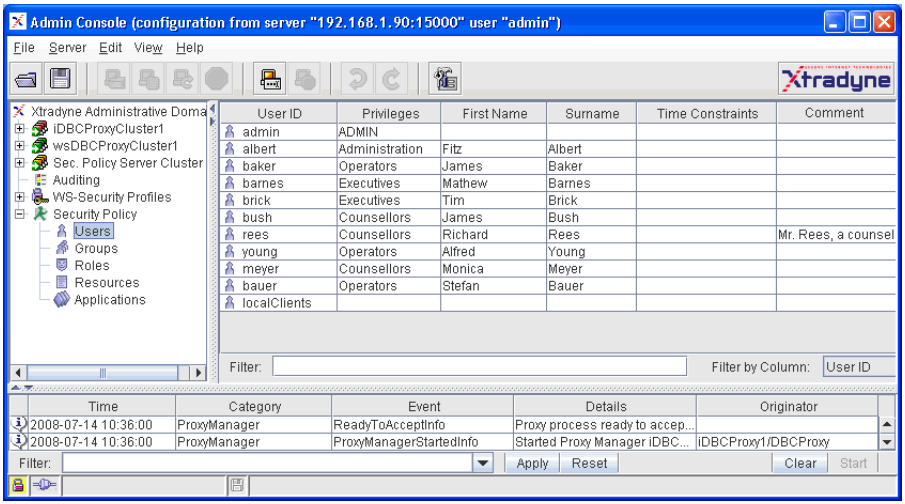


Fig. 2. Screenshot of the Admin Console – Example User List

For I-DBC security policies the Admin Console offers an Learning Mode Wizard. This wizard tries to automatically extract a security policy from event traces. For more details, please refer to “Learning Mode” on page 291.

1.3 Security Policy

The “Security Policy” panel defines where the Security Policy Server stores security policies, and the properties of the chosen storage.

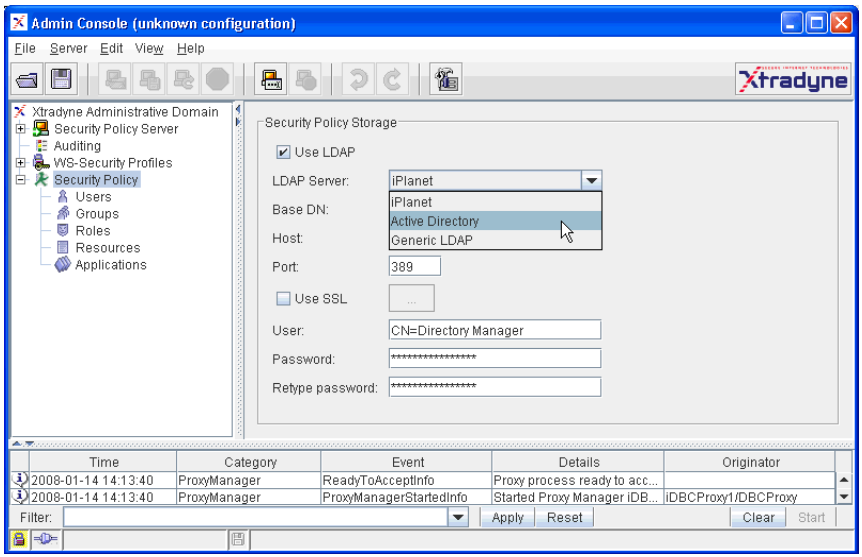


Fig. 3. Security Policy: General Properties

1.3.1 Security Policy Storage

Use LDAP

If you would like to store the security policy in an LDAP Server, check this box. If not checked, the SPS will store the security policy in the configuration file (located on the Security Policy Server host `adm/dbc.config`).

LDAP Server

You can choose an LDAP server type from the drop down menu. The properties of the LDAP server can be configured in the lower part of the panel.

1.3.2 LDAP Server: Prerequisites

Before configuring the LDAP Server's properties with the Admin Console, there are some configuration steps that need to be carried out.

DBC Base DN

Regardless of the LDAP Server type you want to use, you first have to obtain a node (`BaseDN`) from your LDAP administrator, under which the DBC will store the policy information.

obtain a node in your organization's LDAP structure

LDAP Account

You will also need an account that has read/write access to that `BaseDN` node and must be allowed to create and delete subnodes below it. Make sure that nobody else has access to that data, since it contains all your authorization information.

iPlanet: Prerequisites

To configure DBC support for iPlanet, please perform the following steps:

- Import `<INSTALLDIR>/sps/adm/templates/iplanet.ldif` into the LDAP server: With iPlanet 5.1, import the file via the Directory Server console (Tasks-Import Database, please refer to the Directory Server Admin Guide).
- Alternatively, you can use command line tools like `ldapmodify`, for example:

```
ldapmodify -h <ldapserverhost>
           -D "cn=Directory Manager"
           -w <password> -f iplanet.ldif
```

- Configure the LDAP Storage with the Admin Console (see next section).

Active Directory: Prerequisites

To configure DBC support for Microsoft Active Directory, please execute the following steps:

- Backup your Active Directory schema. Schema mistakes cannot be erased.
- Edit the file `activedirectory1.ldif` and `activedirectory2.ldif` (located in the folder `<INSTALLDIR>/sps/adm/templates`). Replace the DN (`dc=xtradyne,dc=de`) with the DN of your own company.
- Import these schema files into Active Directory. You will need appropriate privileges to import the schemas. The schemas are numbered to reflect the order in which they must be imported.

```
ldifde -i -f <INSTALLDIR>/sps/adm/templates/activedirectory1.ldif
ldifde -i -f <INSTALLDIR>/sps/adm/templates/activedirectory2.ldif
```

Generic LDAP

If you want to use a different LDAP adapter, you can choose “Generic LDAP”. Note that the schema of your LDAP Server will have to be adapted to make it work with the DBC.

1.3.3 Configuring the LDAP Server

Configuring the LDAP Server involves filling out the following fields:

- Base DN: the node under which the DBC will store the policy information on the LDAP server.
- Host: the host name or IP-Address on which the LDAP server runs.
- Port: the LDAP Server’s port.
- Use SSL: If SSL is to be used when contacting the LDAP server, enable this checkbox. The SSL properties for LDAP can be configured when pressing the “...” button next to the check box. Defining LDAP SSL profiles is similar to defining SSL and WS-Security Profiles (cf. “SSL and WS-Security Profiles” on page 165). By default the LDAP SSL Profile uses `LDAPClientKey.pem` as private key file, `LDAPClientCert.pem` as certificate file and `TrustedLDAPCAs.pem` contains the trusted CA certificates. These files are located in the `sps/adm` directory.
- User DN: the DBC LDAP user’s distinguished name.
- Password: the DBC LDAP user’s password.

1.4 General Navigation

In addition to the general “Security Policy” panel, the Admin Console offers four views, one for each kind of entity in the model: a list of users, groups, roles, and resources. You can switch between these views by clicking on the entries in the navigation tree (left side of the panel). This displays a list of the respective entities. To view or change the entities, double-click on the entity, use the menu **Edit➔Edit Properties** or the context menu (via the right mouse button). Multiple entities can be selected by pressing the CTRL-key (selecting one by one) or the SHIFT-key (selecting a range of entities).

using the filter

If you are looking for a specific entity, you can use the filter. Type the first few characters of the entity’s ID in the “Filter” text field, between the panes and the entity list. The list shortens immediately, and only matching entities are displayed. Note that the filter is case insensitive and applies only to the ID field of the “User”, “Group”, “Roles”, and “Resources” panel.

sorting entries

Entries can be sorted for each column in increasing or decreasing order. Just click on the heading of the appropriate column once or twice. The filter will then apply to the sorted column.

The following sections describe the properties that can be defined for each of the entities. Pictograms are often used in the text, for example  denotes the entity user,  denotes groups,  roles, and  resources.

Ambiguous User/Group/Role IDs

A user can have the same ID as a group and/or a role (and a group can have the same ID as a role as well). In this case, in all places where this ID is part of a list which can contain more than one entity type, the type is given in parentheses behind the ID (e.g., a user and a group with the ID foo, will be displayed: foo (group), foo (user)). In some cases, the ID is additionally prefixed with a descriptive icon.

1.5 Users

In general, each entity stores two kinds of information: mandatory data, which is essential for the functioning of the DBC, and optional descriptive information. For example, a user’s first name, surname, and a comment are not vital for the DBC, but convenient for you to remember who that user was. An entity’s name can include alphanumeric characters and additionally the following characters: “=”, “.”, “:”, “ ”.

1.5.1 User Properties – General

To create or edit a new user, double-click on the entity, or use the menu **Edit➔Edit Properties** or the context menu (via the right mouse button). A window pops up which comprises four tabbing panes (see Figure 4, “General User Properties”): The General tabbing pane defines the personal data of a user. The Authentication Methods tabbing pane defines by which means the user must authenticate to the system. The Privileges tabbing pane defines the user’s relationship with other entities, for example the user’s memberships in certain groups. The Constraints tabbing pane defines constraints for this user, for example an activation and expiration date.

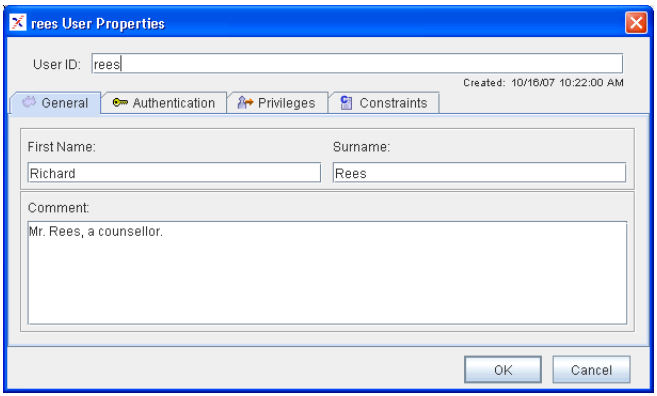


Fig. 4. General User Properties

Each user has a unique *User ID*, a *First Name*, *Surname*, and a *Comment*. First name, surname, and comment are optional information. The User ID identifies a user in the whole system.

The creation date (generated by the system) is displayed in the top right corner. This is for your information only and cannot be edited.

Discard your changes by clicking the Cancel button, accept your changes with OK (the window will disappear) or Apply (the window will stay open). Note that the configuration has to be written to the Security Policy Server for changes to take effect.

1.5.2 User Properties – Authentication Mechanisms

authentication policy

For the system to realize who a user is, he or she must authenticate. The DBC offers several ways of authenticating a user: User ID/Password authentication, IP-based authenti-

cation, SSL authentication, and – when operating a WS-DBC – authentication via SAML assertions. In the following we will explain these methods and the corresponding configuration panel.

To configure the available authentication methods for a particular user, go to the Authentication Methods panel in the "User Properties" window. Right-click into the field "Applied mechanisms" and choose an appropriate authentication mechanism. The properties of each authentication mechanism can be defined on the right side of the panel.

CSIv2 (I-DBC Proxy only)

When using CSIv2 the "Users can assert other identities (CSIv2 only)" checkbox may be enabled to allow users to assert a different identity than the one proven by authentication. For details on how to configure CSIv2 for the DBC, please refer to "I-DBC Proxy Cluster – CSIv2" on page 126.

Authentication via SAML Assertion

This authentication method is only available when operating a **WS-DBC** Proxy.

The WS-DBC can authenticate a client via the SAML assertion included in a SOAP message. For this authentication method there is nothing to configure here. The WS-DBC simply extracts and verifies the SAML assertion. The WS-DBC will check if the assertion was signed by a CA certificate it considers as trusted. The file containing the trusted CA certificates can be defined in the "WS-Security Profiles" panel (see page 165).

Note that, irrespective of the outcome of the verification, no additional authentication information is considered, which means that a failure to verify an existing SAML assertion will lead to rejection of the message.

SSL X.509 Certificate Authentication

SSL authentication and HTTPS

X.509 is a standard for digital certificates used by SSL. Certificates include among other things information about the identity of the certificate holder (i.e. the user). When choosing this authentication method this implies that the user, i.e., the client application, has to use SSL, but we recommend using SSL anyway because it provides integrity and confidentiality protection for messages in transit.

On the right side of the pane you give the distinguished name (DN) of the certificate the client uses to authenticate to the DBC (single elements of the distinguished name have to be separated by a comma).

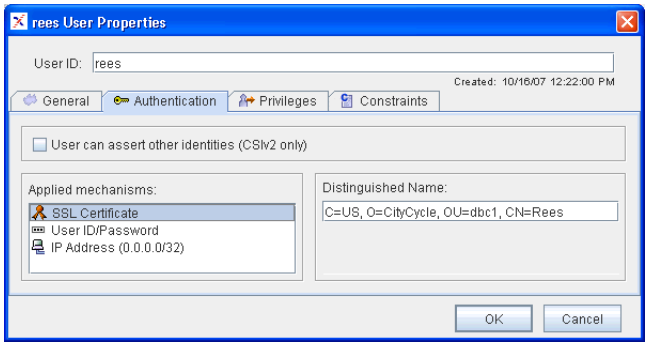


Fig. 5. Authentication Methods: X.509 Certificate authentication

Note that the client certificate DN must be unique, please ensure that multiple users do not share the same client certificate DN.

User ID/Password Authentication

user ID/password authentication

A user can authenticate via a user ID/password scheme. The user ID to be supplied during the authentication process must be the same as the user ID of the peer in the system.

When operating an **I-DBC** Proxy user ID/password authentication may be used with the DBC Authenticator. When operating a **WS-DBC** Proxy this authentication method may be used with HTTP Basic authentication or UsernameToken authentication (see also section “Resource Properties – Incoming Policy” on page 269).

Configure the user ID and password here for the respective user. The user ID is displayed in the upper part of the panel. Provide the password (at least 4 characters long) on the right hand side of the panel.

Note that when a user forgets his password, there is no way of reconstructing it because only a SHA-1 hash is stored in the policy. In such a case the administrator has to assign the user a new password with the Admin Console.

Also note that the number of asterisks displayed in the Admin Console does *not* correspond to the number of letters contained in the password so that the actual length of an assigned password cannot be observed by an onlooker.

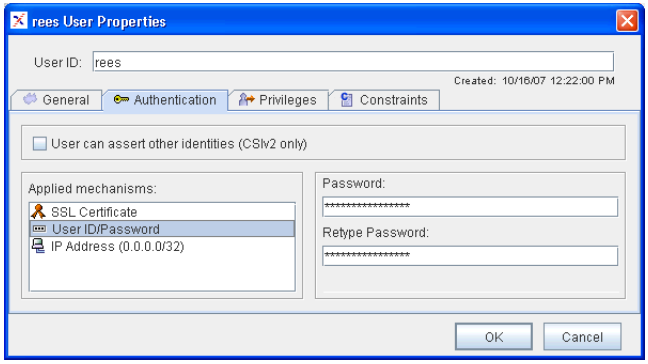


Fig. 6. Authentication Methods: User ID / Password Authentication

Changing the Admin User's Password

The security policy contains one predefined user – the user "admin". This user has by default the right to administrate all parts of the configuration (cf. "Role Properties – Administration" on page 261). To change the admin user's password double click on the admin user in the users list and go to the Authentication Methods pane (depicted above). Enter the new password and confirm it.

KEEPING THE DEFAULT PASSWORD IS A SEVERE SECURITY RISK!

IP Address

In case a particular user always connects from a single IP address or a range of IP addresses, you can map the IP address directly to the user ID. In this case, the user ID really corresponds to a name for a host or a subnet. A subnet mask can be provided which *identifies* – rather than mask out – the relevant bits of the *policy IP* address that

are compared to the *source IP* address. The policy IP address is the one defined in the panel. The source IP address is the one from where the client connects.

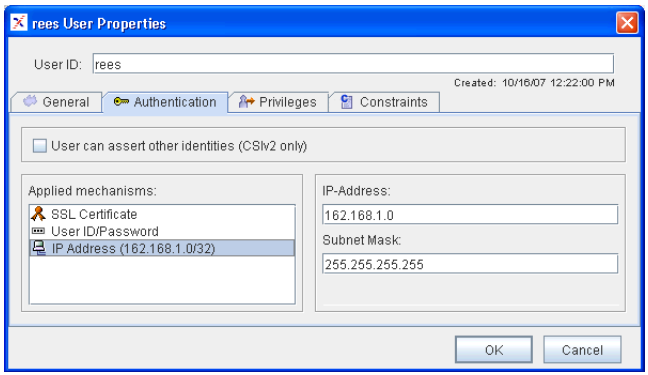


Fig. 7. Authentication Methods: IP Address

Note that in the list of applied mechanisms on the left side of the “User Properties” panel the configured IP address and netmask length is displayed to distinguish different configured IP address mappings.

Using the Subnet Mask

The comparison works as follows: First the source IP is bit-wise AND combined with the subnet mask. The result is then compared to the policy IP address. If the result is equal the IP address matches. As a formula: *(source IP & subnet mask) = (policy IP & subnet mask)*. Please note that all zero bits in the subnet mask must also be zero in the IP address, or as a formula: *(policy IP & subnet mask) = policy IP*
Otherwise the given IP address is invalid with respect to the subnet mask.

For examples, see table below.

policy IP Address	Subnet Mask	Description
144.10.210.16	255.255.255.255	match exactly that IP address (default value)
144.10.210.0	255.255.255.0	matches the class C net, i.e. all host IP addresses starting with 144.10.210
144.10.210.176	255.255.255.240	matches a 4-bit subnet 144.10.210.176 – 144.10.210.255
144.10.210.7	255.255.255.0	is invalid as zero bits in the subnet mask are not zero in the originator IP address

Table 1. policy IP address and subnet matching

You should be aware that this kind of authentication is vulnerable to IP spoofing attacks. It can be useful, for example, if a partner organization has a fixed IP address range and the whole partner organization should be mapped onto the same user ID.

Number of Access Sessions and Access Session Hierarchy

This section applies to the configuration of **I-DBC** Proxies only. If you configure a WS-DBC, please proceed to “User Properties – Privileges” on page 250.

User management and the number of access sessions that will be created in the I-DBC are closely related. When contacting the I-DBC every client is assigned an IP-based access session. This may be the same access session for multiple clients, depending on how the user management is configured. For each other authentication mechanism, a separate access session is created in the hierarchy below the IP-based access session for every unique user identity. For that reason, a client using SSL will be assigned two different access sessions, one for its IP address and another for its SSL identity. If two different clients have the same SSL identity but connect from two different IP addresses which are not mapped onto the same user identity, four access sessions will be created. That is because the access session hierarchy is a tree, meaning the SSL identities are located below the nodes of the IP-based access sessions. Access sessions can retrieve information (e.g., IORs) from other access sessions that are above them in the hierarchy.

Example: Access Session Hierarchy

Let’s assume that there are two users “client1” and “client2”. Both use authentication by IP address. “client1” connects from IP address 192.168.1.11 and “client2” from IP address 192.168.1.12.

Two access sessions will be created when the clients contact an I-DBC that uses a default configuration. The access sessions will be assigned the ids that correspond to the IP address they connect from (e.g.: “id=192.168.1.11” and “id=192.168.1.12”). In the access session hierarchy the two new access sessions are directly below the root node (as depicted on the left side in figure 8).

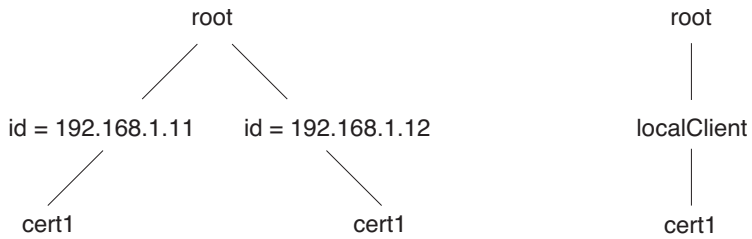


Fig. 8. Access Session Hierarchy

If “client1” additionally uses an SSL certificate “cert1” another access session with the id “cert1” will be created. This access session will be below the node “id=192.168.1.11” in the access session tree.

If another client, for example, “client2” uses the same certificate another access session with the id “cert1” will be created below the node “id=192.168.1.12”. In this case, there will be four access sessions for two clients (see figure 8).

Example: One Access Session For Multiple Clients

You can use the Admin Console to configure your user management in a way that multiple clients connecting from different IP addresses share the same access session. Go to the “User” panel and create a new user, for example, with the id “localClients”. As

authentication method choose “IP address” for this user and specify, for example, the IP address 192.168.1.0 together with the netmask 255.255.255.0 (see screenshot below).

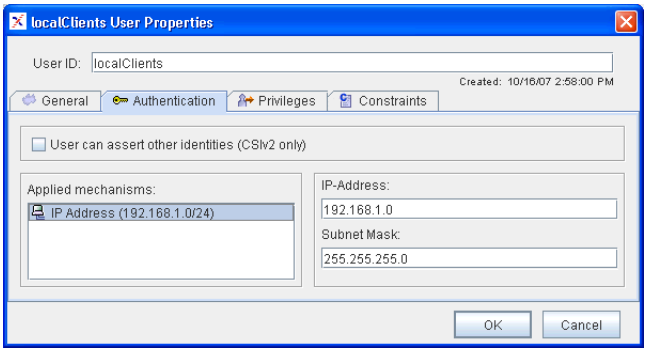


Fig. 9. Configuring One Access Session for Multiple Clients

This configuration implies that all clients with an address from the range 192.168.1.0 to 192.168.1.255 will be resolved to the user name “localClients”. With such a configuration only one access session will be created for clients connecting from this range of IP addresses. If your clients additionally use an SSL certificate “cert1” a second access session will be created below the node “localClients” (as depicted on the right side of figure 8, “Access Session Hierarchy”, on page 248).

1.5.3 User Properties – Privileges

On the “User – Privileges” panel you can choose those entities from the list of existing groups, roles, and resources (on the right side of the panel) for which the user has privileges (left side of the panel).

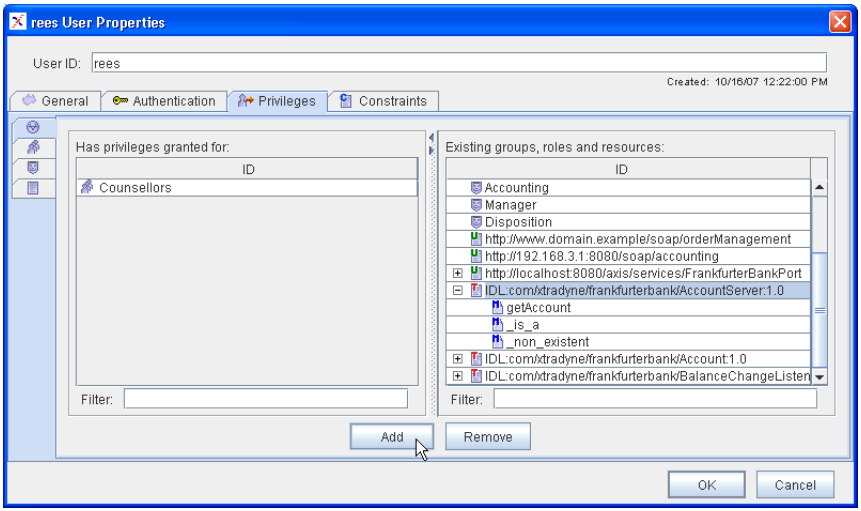


Fig. 10. User Properties: Privileges

If a resource accepts specific operations, you can view these operations by clicking on the ⊞ next to the resource. You can grant privileges to send operations to a resource or for single operations. When double clicking on an operation the “Resources – Accessors” panel for that operation will pop up.

On the left hand side, there are additional tabs which let you focus your view on different aspects of memberships. When selecting the tab all memberships of the user are displayed in the left window and all known groups, roles, and resources are listed in the right window. The tab lets you view only the group memberships in the left window and all existing groups in the right window. The tab lists only the roles the user is a member of in the left pane, and all existing roles on the right side. Finally, the tab lists only the resources the user is explicitly granted access on the left side and all resources on the right side.

Recall the basic modelling concept: A user can be member of a group (displayed when choosing the pane on the left side of the Privileges tabbing pane). A group can in turn be assigned to roles, which are granted permissions to certain resources. A user can also be assigned to a role or even a resource directly. Direct role memberships are dis-

played when choosing the panel . Direct resource memberships are displayed when selecting .

granting privileges for the selected user

In each of the panels one or more entities can be selected by pressing the CTRL-key (selecting one by one) or the SHIFT-key (selecting a range of entities). Use the filter to shorten the displayed list. While you are typing, the list is reduced to the entities with names that begin with the letters you typed. To grant a user privileges for other entities, select the entities in the window on the right side and press the “Add” button. The entities will now appear on the left side of the Privileges panel.

Groups and roles (but not resources) have similar tabbing panes because groups can be members of other groups, roles or resources. Additionally, roles can be members of other roles, and accessors of resources.

1.5.4 User Properties – Constraints

On the “Constraints” tab you can define constraints for the selected entity. The available constraints are the same for all entity types (users, groups, roles, resources). Currently, three different types of time constraints may be defined:

- Activation Date: Defines a date when this entity will be activated.
- Expiration Date: Defines a date when this entity will expire.
- Time Frame: Defines a time frame during which this entity will be activated.

Note that when defining time constraints you should make sure that the hosts on which the DBC Proxy and the Admin Console are running have correct time zone information (i.e., correct local time). Time is entered in local time and converted to universal time (UT) by the Admin Console. In case of time frame the begin and end points of the time interval are converted to seconds.

On the Constraints tab, you can define constraints for this user.

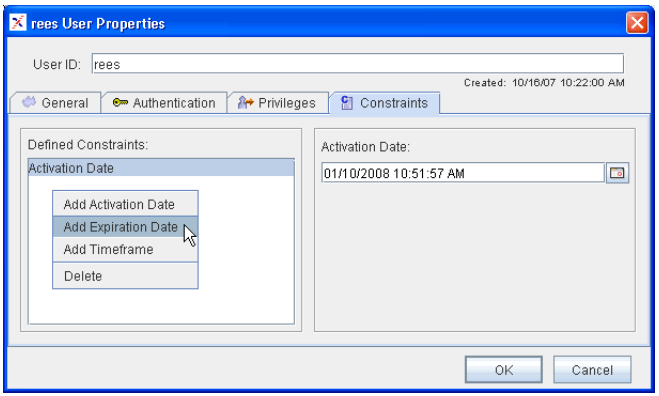


Fig. 11. Define Constraints

Activation/Expiration Date

Activation Date

You might want to create a user in advance – before he or she shall have access to the system. in this case you can add an “Activation Date” constraint and enter a date in the future. This may be, for example, the first working day of a new employee or the beginning of a freelancer’s work. Entities that are not yet activated are handled as if they do not exist.

Expiration Date

Terminating the access for a user works the same way. You can add an *Expiration Date* constraint, and the access for that user is automatically revoked from that day/time on. Again, this can be useful for freelancers or temporary cooperation with external partners. Entities that are expired are handled as if they do not exist.

To add a an Activation/Expiration Date constraint right-click into the “Defined Constraints” area and select the type of constraint you would like to add from the pop-up list (e.g., Activation Date). Then enter the date in the text field that appears on the right side of the panel (enter, for example, 01/01/10 00:00:00 am). When clicking on the calendar symbol next to the text field a comfortable date chooser (calendar) pops up (see screen-

shot below). You can choose the month and year at the top and the day which adapts to the chosen month/year below.

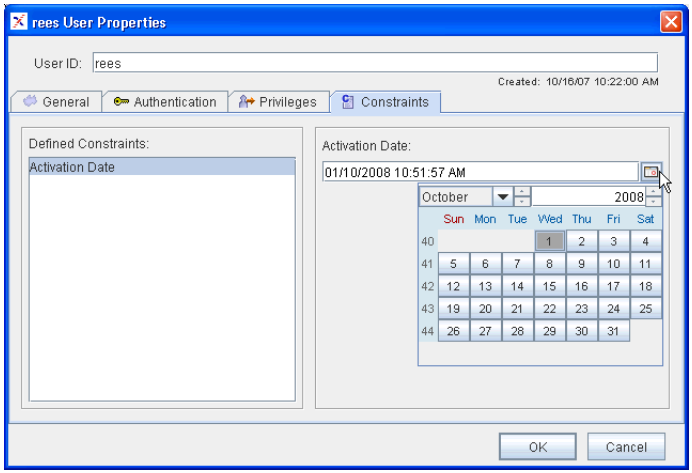


Fig. 12. Choose a Date

Time Frame Constraint

Select an interval type from the drop-down box. You can choose between **Once**, **Hourly**, **Daily**, and **Weekly**. Depending on the chosen type, the input field of the constraint changes. In any case you can define the interval begin and end values.

Once

When choosing **Once**, the entity will be activated exactly once: From the date/time chosen as interval begin to the date/time chosen as interval end. As interval begin and end a complete date including the time (hour, minute, second am/pm) has to be entered. You may use the date chooser by clicking on the calendar symbol next to the text field (cf., Figure 12).

Hourly

When choosing **Hourly**, the entity will be activated every hour. As interval begin and end minutes and seconds can be entered. You may use the spinner next to the input fields. The begin and end values are taken relative to the full hour, for example, when entering the interval 15:00 - 30:00 the entity will be activated every hour, a quarter past the full hour and will be deactivated 15 minutes later.

Daily

When choosing **Daily**, the entity will be activated every day. As interval begin and end hours, minutes, seconds and am/pm can be defined. You may use the spinner next to the input fields or the drop-down box for selecting am/pm. The begin and end values are taken relative to the start of the day, for example when entering 9:00:00 am - 5:00:00

pm, the entity will be activated in the morning at nine o'clock local time and will be deactivated in the after noon at five o' clock.

Weekly

When choosing **Weekly**, the entity will be activated every week. As interval begin and end the day, hours, minutes, seconds and am/pm can be defined. You may use the drop-down box to select the day and the am/pm value. The spinner may be used to adjust the other values. The begin and end values are taken relative to the start of the week, for example, when entering Mon 9:00:00 am - Fri 5:00:00 pm the entity will be activated Monday morning at nine o'clock and will be deactivated Friday afternoon at fie o'clock.

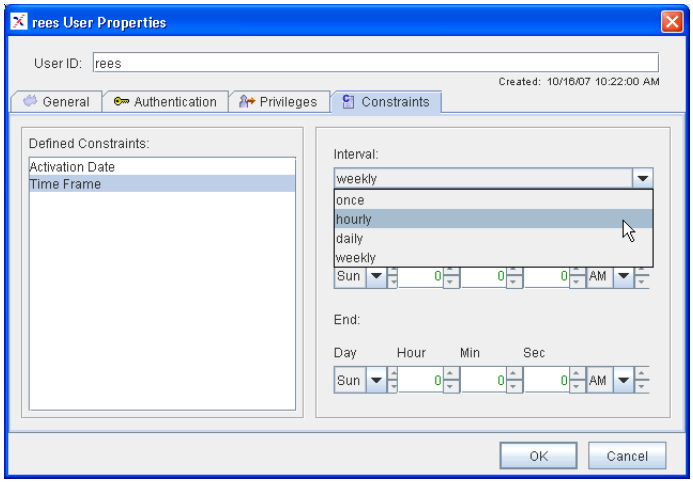


Fig. 13. Define a Time Frame Constraint

1.6 Groups

The Groups pane shows the list of groups, with the group ID, the group's members, their privileges, time constraints, and a comment. A group's members can be an arbitrary

number of users and other groups. Privileges can be group and role memberships, and permissions to access resources.

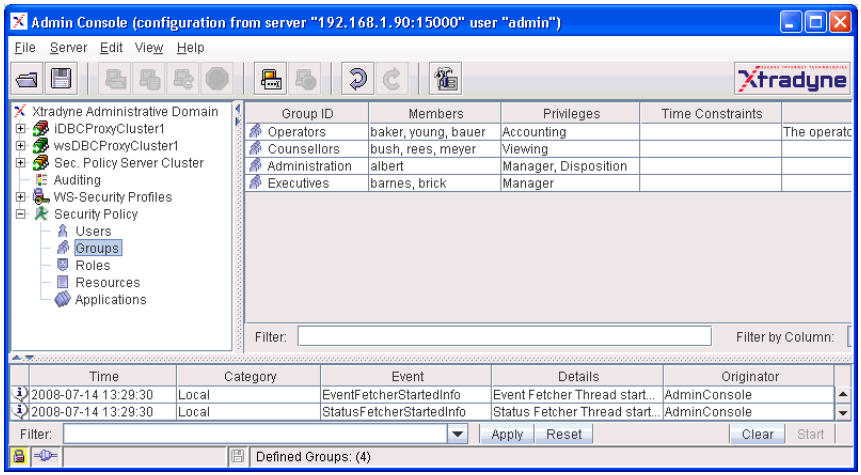


Fig. 14. Admin Console – Group List

To create or edit a new group, double-click on the entity, or use the menu **Edit➜Edit Properties** or the context menu (via the right mouse button). In any case a “Group Properties” panel is displayed, that allows you to configure the group’s properties.

Constraints

Note that the constraints that can be defined on the Constraints tab are the same for each type of entity (user, group, role, resource), therefor, constraints are explained only once in section “User Properties – Constraints” on page 251.

1.6.1 Group Properties – General

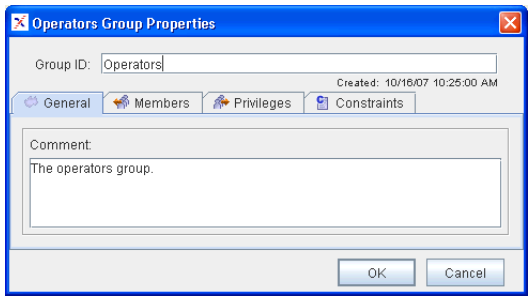


Fig. 15. General Group Properties: General

The general properties of a group are almost the same as the user properties. The name of the group can be changed in the top field of the window without changing its identity.

1.6.2 Group Properties – Members

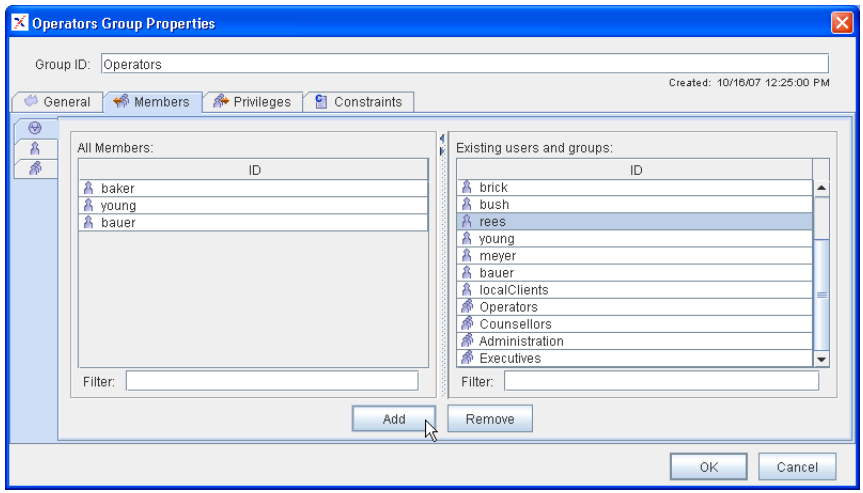


Fig. 16. Group Properties: Members

The Members tabbing pane shows the users and other groups that are members of the group with the specified group ID. Remember that a group *has* members (other groups and users), which are displayed on the Members tabbing pane, and *can be member* else-

where (of other groups, roles, and resources) which is shown on the Privileges tabbing pane.

1.6.3 Group Properties – Privileges

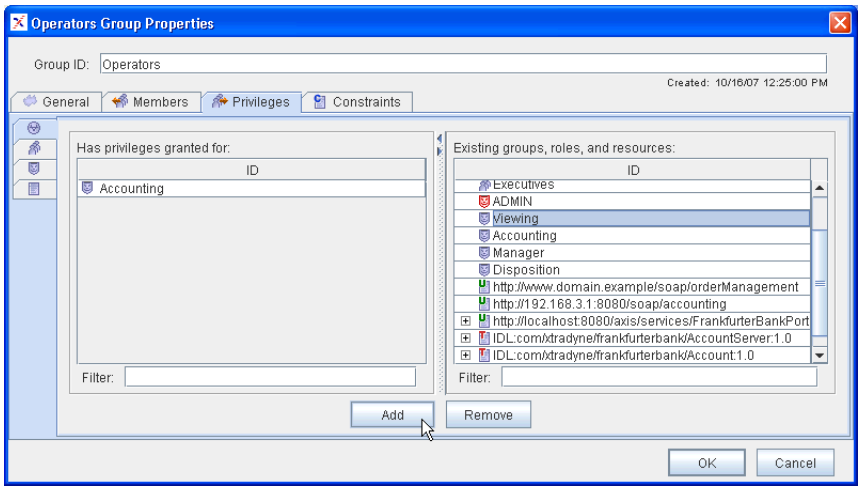


Fig. 17. Group Properties: Privileges

As in the user's privileges panel there is a ⊞ next to the resource if it has specific operations. Privileges can be granted for the whole resource or for single operations. The group's Privileges tabbing pane is quite similar to the user's. Again, you'll find the tabs for viewing the direct memberships on the left hand side. The tab lists all memberships of this group. The tab lists only the group memberships. The tab lists the roles the group is assigned to. Finally, the tab lists the resources the group is explicitly granted permission to access.

granting privileges for the selected group

To grant that group privileges for other groups roles or resources, select the entities in the window on the right side and press the "Add" button. The entities will now appear on the left side of the "Group Properties – Privileges" panel.

1.7 Roles

From a configuration point of view, roles look much like groups:

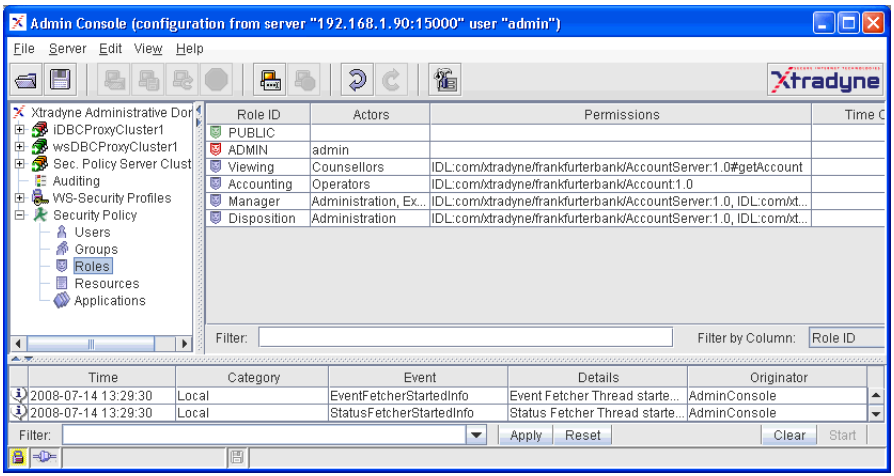


Fig. 18. Admin Console – Role List

PUBLIC and ADMIN role

Note the special roles "PUBLIC" and "ADMIN". These are predefined roles which cannot be deleted or created. The "PUBLIC" role cannot grant permissions to other entities explicitly. It is designed for granting general access to public resources or to public operations of a resource (for more details see section "Public Access" on page 293). The "ADMIN" role is designed for granting permissions for administrative tasks like starting or stopping DBC Proxies (for details see "Role Properties – Administration" on page 261).

To create or edit a new role, double-click on the entity, use the menu **Edit→Edit Properties**, or the context menu (via the right mouse button). In any case a "Role Properties" panel is displayed, that allows you to configure the role's properties.

Constraints

Note that role constraints that can be defined on the Constraints tab are the same for each type of entity (user, group, role, resource), therefor, constraints are explained only once in section "User Properties – Constraints" on page 251.

1.7.1 Role Properties – General

You can configure roles just like groups, specifying a role ID and a comment.

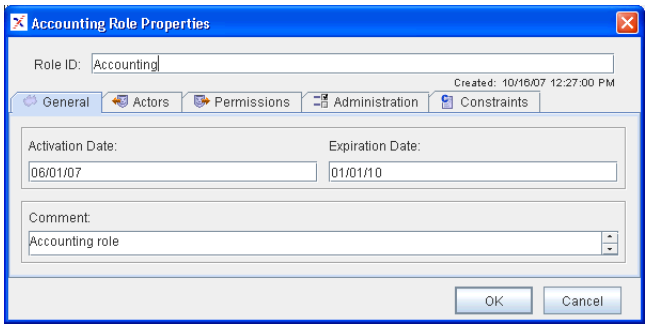


Fig. 19. General Role Properties: General

1.7.2 Role Properties – Actors

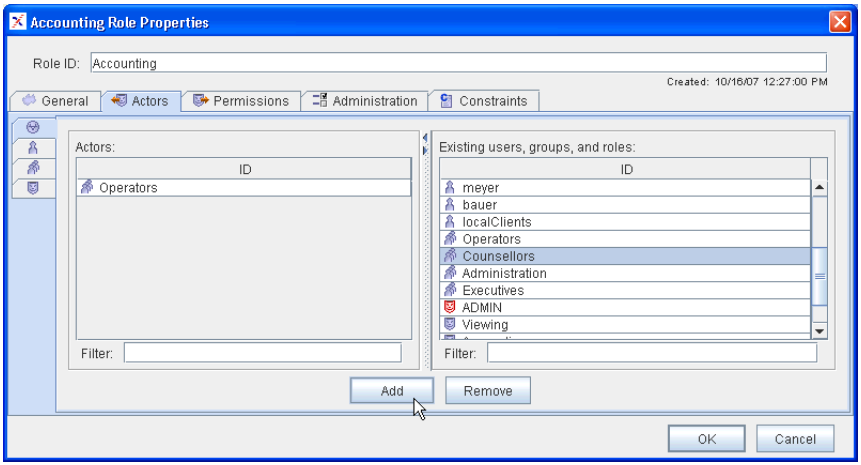


Fig. 20. Role Properties: Actors

Roles have Actors to whom they pass on the permissions received from resources or roles. Actors are groups, or roles, and users (see also Figure 1, “Components of the access control policy,” on page 236).

1.7.3 Role Properties – Permissions

The role's Permissions tabbing pane is similar to the user's and group's privileges tabbing pane. Roles receive permissions from resources or other roles (see screenshot below).

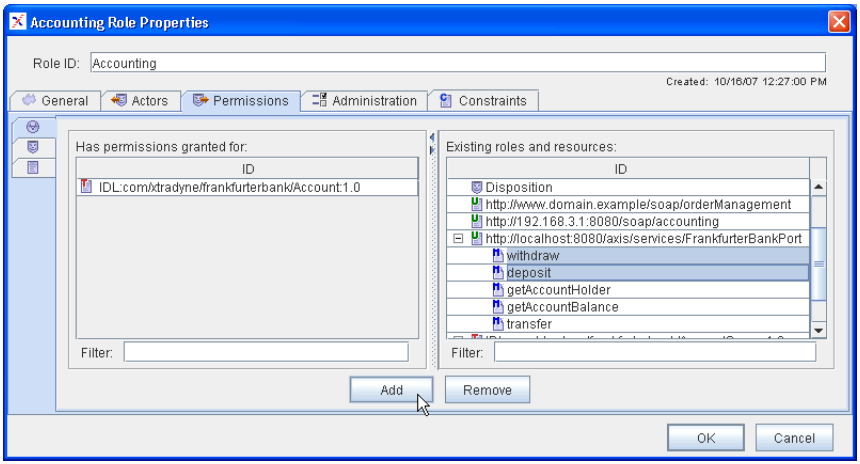


Fig. 21. Role Properties: Permissions

As usual there is a ⊞ next to the resource if it accepts specific operations. In this case permission can be granted for the whole resource or for single operations.

Again, you can choose those entities from the existing roles and resources on the right side for which the role "Has permissions granted for" (the panel on the left side). There are also tabs on the left side of the panel. The tab combines the roles and resources for which this role has permission. The tab shows the role's memberships in other roles (super roles). Finally, the tab lists the allowed access to resources.

granting permissions for the selected role

To grant a role permissions to access resources or memberships in other roles, select the entities in the window on the right side and press the "Add" button. The entities will now appear on the left side of the "Role Properties – Permissions" panel. As explained earlier, granting a role X membership to the role Y causes X to inherit Y's permissions. Hence, X can be viewed as "senior to" Y as it has all permissions that Y has, but may have additional permissions of its own.

1.7.4 Role Properties – Administration

The “Administration” tabbing pane lists a role’s administrative permissions, i.e., its rights to perform certain administrative tasks such as creating new users, or restarting the Security Policy Server (SPS). To grant such a right to a role, check the box in the “Allow” column for a specific administrative right.

Note that the permissions granted to the role ADMIN are inherited by the predefined user `admin` which is a member of that role by default. The ADMIN role has the rights to read, edit, and save all parts of the configuration by default. Furthermore, this role may start, restart, and stop the Security Policy Server and the DBC Proxy.

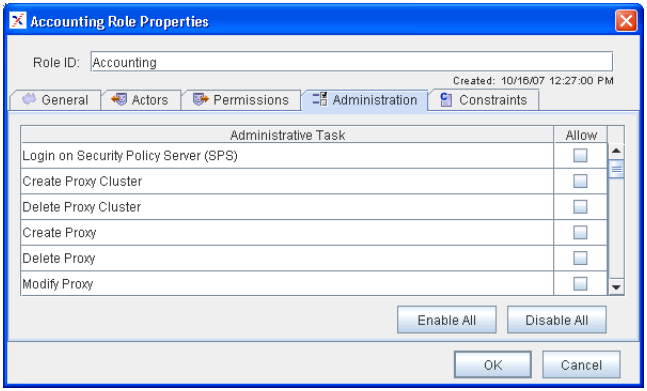


Fig. 22. ADMIN Role – Granted administration rights

Note that for renaming components (proxies, user, groups, roles etc.) add and delete rights are required.

There are some administrative tasks that apply to the SPS Client (a command line interface to the SPS). The SPS client is explained in “Administrative Rights for SPS Client Operations” on page 48.

1.8 Resources

Resources point to target services. These services are Web Services specified by their URL[1]. Clients only target a virtual resource in the DBC Proxy, not the actual target which is protected by the DBC. The original target URLs are used only in the access control policy. The actual target is defined by the resource mapping, see section “Resource Mappings” on page 150.

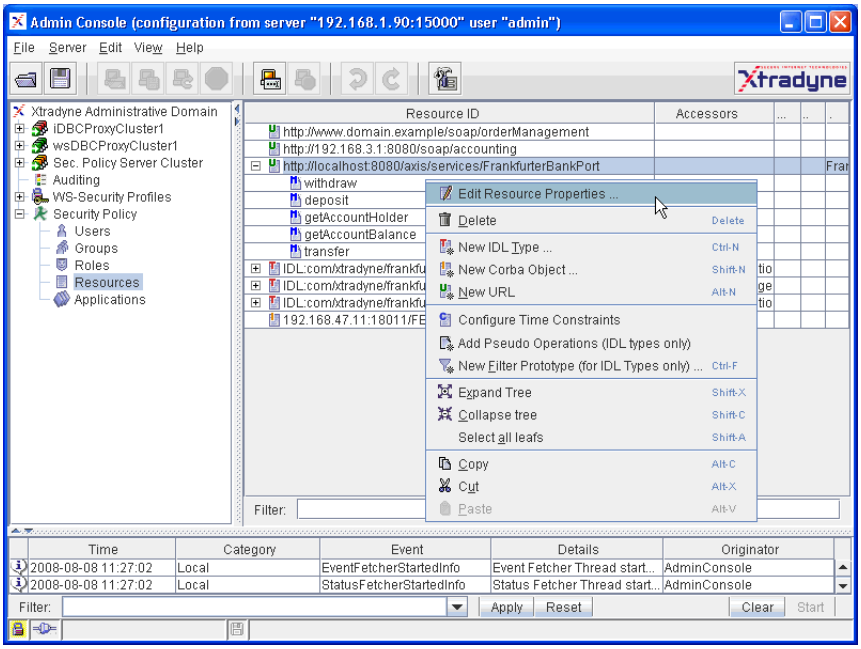


Fig. 23. Admin Console: Resource List

The resources pane shows the list of resources, with the resource ID, the resource’s accessors, filters, time constraints, and a comment.

Adding a Resource

To add a new resource, select **New URL...** from the context menu (right-click in the resources table).

1 They may as well be CORBA Services specified by their IOR. IOR Resources are only of interest when configuring I-DBC’s and are not explained here.

If a WSDL document describing the interface that you want to expose is available the exposure wizard may be used instead of defining the resource manually (please refer to section “WSDL Exposure Wizards” on page 264). Operations offered by a resource can be viewed by clicking the ⊞ next to the resource.

When double-clicking on an operation the Accessors panel for this operation will pop up (see also “Resource Properties – Accessors” on page 291).

1.8.1 WSDL Exposure Wizards

The Admin Console offers an exposure wizard to import WSDL descriptions of a resource. The WSDL import wizard can be accessed via the menu item **File ➜ Import WSDL...** or via the context menu on the “Security Policy – Resources” panel.

On the first panel of the WSDL exposure wizard you choose a file or supply a URL where the WSDL description can be retrieved.

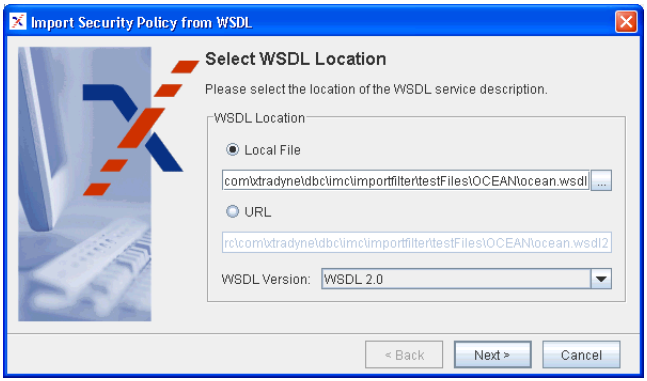


Fig. 24. The exposure wizard: Select Location

When choosing file as location type, you can select the WSDL file with a file selector. When choosing URL, please supply the URL pointing to the WSDL document for the service that you want to expose. The Admin Console will then retrieve the WSDL document for your service from that location. A sample URL could look like this:

`http://falcon:7001/WS-DBCExamples/FrankfurterBank?WSDL`.

WSDL Version

Choose the proper WSDL version from the drop-down box.

select operations to import

The wizard then extracts a service description from the WSDL document. The next step is to choose the resources and operations that you want to import into the security policy.

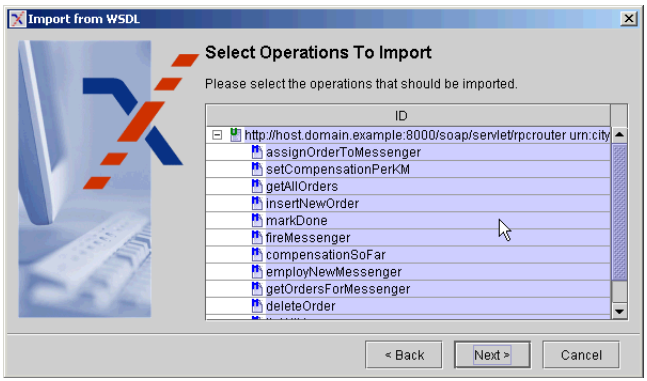


Fig. 25. The exposure wizard: Select Operations to Import

The next step is to configure a resource mapping for exposure to clients, i.e., a logical name with which the client can contact the protected service through the proxy. The virtual service URL used by a client is formed by appending this name as a path suffix to the proxy address, for example, `http://falcon:8000/name` for a WS-DBC Proxy on host `falcon`, which accepts requests on port `8000`. This resource mapping can later be viewed and edited on the “WS-DBC Proxy – Resource Mappings” panel, see section “Resources Mappings” on page 150 for a detailed description.

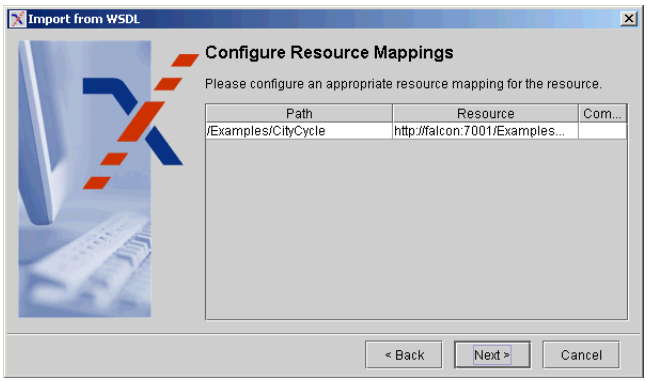


Fig. 26. The exposure wizard: Configure a Resource Mapping

conflicts while importing

If the resource that you want to import already exists in the security policy the Admin Console will show a panel and you can choose to “Update”, “Overwrite”, or “Skip” the

conflicting resource. When choosing “Update” all policy definitions for this resource will be kept and only those entities that do not already exist in the policy will be imported. Note that if you choose “Overwrite” the resource properties will be reset with the default security policy settings!

As the last step you can select an existing application or create a new one to add the imported entities to. Applications are explained in detail in section “Applications (Application Domains)” on page 295. If you don’t want to import into an application, just press the “Next” button.

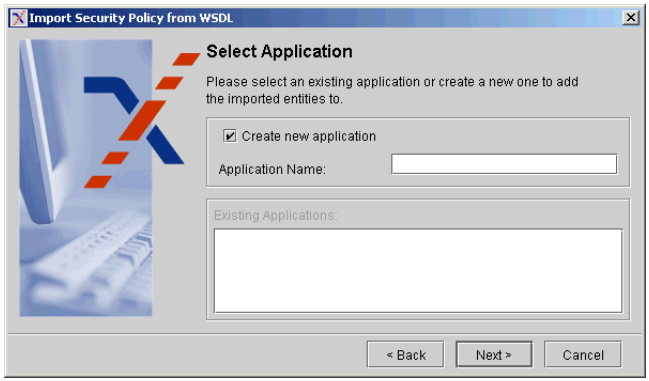


Fig. 27. The exposure wizard: Select Application

The WSDL import is complete now, but note that the security policy for the imported resources should be refined. By default, the imported resources have no accessors, so the WS-DBC will not allow any access (unless the resources already existed in the policy before importing and accessors have already been defined). Please set the security policy within the “Resource Properties” panel and associate the resources with its accessors as described in following sections.

1.8.2 Resource Properties

To edit a new resource, double-click on it or use the menu **Edit➜Edit Resource Properties** or the context menu (via the right mouse button). In either case, the “Resource Properties” panel pops up. This panel has several tabbing panes General, Incoming Policy, Outgoing Policy, Interface, Filters, Accessors, and Constraints.

Constraints

Note that the constraints that can be defined on the Resource Properties Constraints tab are the same for each type of entity (user, group, role, resource), therefor, constraints are explained only once in section “User Properties – Constraints” on page 251.

1.8.3 Resource Properties – General

The General tabbing pane defines general properties of a resource. The Resource ID can be changed in the text field at the top of the panel. Additionally, you can specify whether the resource ID shall be used as the service location. Furthermore WSDL GET Requests can be allowed by enabling the respective check box.

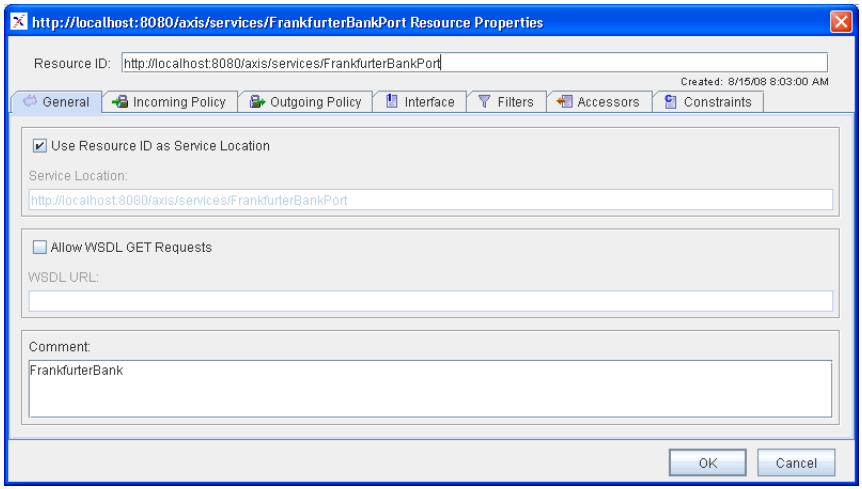


Fig. 28. General Web Service Resource Properties.

Resource ID

Web Service resources are defined by an ID. This ID may be an arbitrary string or may denote the service location.

Use Resource ID as Service Locator

When this box is checked, the resource ID will be taken as Service Locator (target URL). Note that in this case, the ID has to be a valid URL (`http://<host>:<port>/<servicepath>`). The host can be specified by its resolvable name or its IP address. The port number is optional.

An optional XML namespace can be defined on the Interface panel, see “Resource Properties – Interface” on page 281.

Allow WSDL GET Requests

WSDL Bootstrap

Some Web Services frameworks use WSDL for bootstrapping purposes (see also discussion in section "Using the WS-DBC in combination with WSDL" on page 42). To use such a mechanism through the WS-DBC, check the "Allow WSDL GET requests" box and supply the actual URL for the service WSDL in the text box. If the resource was imported from a WSDL using the Exposure Wizard (cf. above) and the WSDL was obtained from a URL, then this URL will already be shown in the text field. For WSDL imported from local files, a URL has to be explicitly entered to support WSDL GET requests.

Clients use the following convention to retrieve the WSDL for a service: If, for example, the exposed service URL is `http://proxy/servicepath`, then the WSDL is accessed using a HTTP GET to `http://proxy/servicepath?WSDL`. The WS-DBC Proxy reacts to this request as follows:

- retrieve information about the resource, which includes the actual URL for the WSDL (supplied in the text box as described above).
- If the resource has WSDL access marked as allowed, then the proxy will retrieve the WSDL from the URL given in this text field. Note that no access control policies for this resource are checked at this stage.
- The addressing information found in the WSDL will be changed so that only the exposed service address, not the actual target address, is visible to clients.
- The WS-DBC Proxy returns the WSDL to the client.

For security reasons, the WS-DBC Proxy does not use the client's original HTTP GET request but creates its own request, so no client input is forwarded into the interior network.

1.8.4 Resource Properties – Incoming Policy

The Incoming Policy panel of a Web Service resource defines authentication, protection, and message analysis policies for SOAP requests and responses sent to the DBC Proxy.

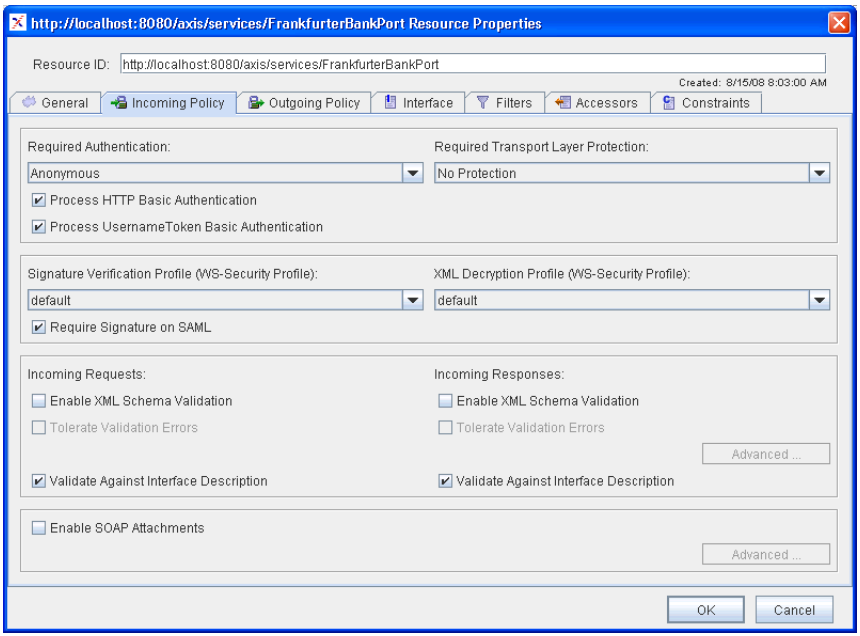


Fig. 29. Web Service Resource Properties: Incoming Policy

Required Authentication

The minimum *Required Authentication* levels apply to all operations of a resource. The WS-DBC supports the following authentication mechanisms (listed from the weakest to the strongest):

required authentication

- *anonymous*
- *Source IP Address*: authentication by IP address
- *Basic Authentication*
- *SSL Client Authentication* (512 bit RSA, 1024 bit DSS)
- *SAML Assertion*
- *Blocked*: “emergency stop” to disallow any access

Note that the chosen authentication method is the minimum required method, i.e., if you require at least SSL X.509 credentials, authentication via SAML assertion will also be accepted. The WS-DBC will look first for authentication information for the stronger mechanism (e.g. SAML) and only consider other mechanisms if no authentication information for that level is available.

Selecting anonymous means that the WS-DBC will accept unauthenticated messages, or any of the above authentication mechanisms. Note that if a client offers an authentication method and the authentication fails, the WS-DBC Proxy will reject the authentication attempt even if the required authentication is anonymous.

Required Transport Layer Protection

required protection

The *Required Protection* describes the level of encryption/integrity protection on the connection. This level applies to all operations of a resource. Possible values for this level are *No Protection* or *SSL Encryption*. If you choose *SSL Encryption*, the WS-DBC Proxy will check if SSL is used on the connection.

The mere possession of access permissions to invoke operations on a resource may not always suffice – the access request must be transmitted via a connection which meets at least the levels of authentication and protection required by the resource.

Process HTTP Basic Authentication

If you check the “Process HTTP Basic Authentication” box, the WS-DBC will authenticate incoming message using HTTP Basic Authentication. If HTTP Basic Authentication succeeds, but the password verification fails, the request will be rejected and an *AuthenticationBasicAuthenticationFailure* event is triggered.

If the “Process HTTP Basic Authentication” box is unchecked the WS-DBC Proxy will ignore HTTP Basic Authentication header parts.

Process UsernameToken Authentication

UsernameTokens (UT) are part of the WS-Sec Specification. A UsernameToken is used to transport a username/password scheme for authentication and is represented as a child of the WSSE header element.

There are two variants of passwords: `wsse:PasswordText` and `wsse:PasswordDigest`. If the type attribute is missing, `wsse:PasswordText` is assumed. The WS-DBC currently only supports plain text password, because most user registries do not expose user passwords in plain text (which would be needed to calculate the digest and verify it on the receiving site).

If you check the “Process UsernameToken Authentication” box, the WS-DBC will authenticate incoming messages by their UsernameToken element. Any failure in parsing this element (missing elements, wrong name spaces, unexpected password type) will result in an event *XMLWSSEUsernameTokenInfo* and the UT element will be ignored. If authentication by UsernameToken succeeds, but the password verification fails, the request will be rejected and an *AuthenticationBasicAuthenticationFailure* event is triggered.

If the “Process UsernameToken Authentication” box is unchecked the WS-DBC Proxy will ignore UsernameTokens, no parsing is done on the UT element, so no failures will be detected. If you want to check the correctness of the UT element structure, but do not want the WS-DBC to check authentication, you can use XML Schema validation (see below).

Signature Keys / XML Decryption Keys

SOAP messages may carry a SAML assertion, or they may be encrypted. To validate a signature or to decrypt the message the WS-DBC needs appropriate key profiles (WS-Security Profile). Such a profile can be chosen from the respective drop down box. For more information on how to define key profiles, please see “SSL and WS-Security Profiles” on page 165.

Require Signature on SAML

If it is required that SAML assertions in incoming messages are signed by a digital signature, activate the check box “Require Signature on SAML”. The WS-DBC will then reject any messages that carry an unsigned SAML assertion.

Note that the WS-DBC checks all incoming messages for valid signatures and that invalid signatures will always be rejected. This does not imply that all messages must be signed!

Incoming Requests/Incoming Responses

XML Schema Validation

XML Schema Validation for incoming SOAP message requests and responses can be enabled by activating the checkbox (an XML wellformedness check is always carried out).

XML schema validation properties

Press the “Advanced” button in the lower right corner to bring up the “XML Schema Validation Properties” panel. Define the directory where the schemas are located (by default `<INSTALLDIR>/wsdbc/adm/schemas`) and the list of required schema files. To add a schema file click with the right mouse button into the “XML Schema

File" text field and choose **Add Schema Filename** from the context menu. Only the listed files in the given directory will be used for schema validation for this resource.

Schema files will be loaded on-demand, i.e., when a resource for which schema validation is configured is first accessed. Schemas will be loaded in the order in which they are listed. Any other schema or DTD files that are referenced from (and thus required by) the schemas in this directory must also be placed there. DTD files in this directory are only loaded when referenced by a schema. Note that schema files must not be loaded repeatedly: Do not list the same schema file several times and make sure that different schemas do not import the same schema files.

For testing purposes you may check the box "Tolerate Validation Errors". In this case the WS-DBC Proxy will not block messages that do not validate correctly. This option should **only** be used for testing purposes. Leaving this box checked in a production environment imposes a sever security risk!

Validate Against Interface Description

Check the "Validate Against Interface Description" box for incoming requests or responses, respectively to enable validation. The SOAP body of the incoming request or response will be validated against the interface description defined on the "Operations" tab (see section "Resource Properties – Interface" on page 281).

Preparing Schema Files for Use by the WS-DBC

Before copying schemas to the configured directory, they should be carefully inspected. In principle, schemas are allowed to import other schemas and DTDs from any location, including locations on the Internet. However, the latter case is forbidden by the WS-DBC Proxy for security reasons, so you need to make sure that any XML schema that you provide yourself will find all other schemas that it requires in that directory. Invalid or syntactically ill-formed schema definitions will simply be ignored by the WS-DBC Proxy (an *XMLSchemaLoadingFailure* audit event will be sent). It is therefore important to provide correct schema definitions. We recommend using XML editing tools when writing schema definition files to avoid syntactic errors.

The `wsdl2schema` and `schematest` Utility

XML schema files can be generated from Web Service definitions in WSDL using the `wsdl2schema` utility. Additionally, you should always check your schemas using the command `schematest`. The `wsdl2schema` and `schematest` utilities are described in detail in Chapter 5 "WS-DBC Tools" on page 49 of the Deployment Guide.

Schemas Included in the WS-DBC Installation

The following table lists the schemas included in the WS-DBC installation. Schemas can be found in the directory `<INSTALLDIR>/wsdbc/adm/schemas`, which is also the default schema location:

Schema	**Meaning**
`namespace.xsd`	needed by SOAP 1.2 schema
`oasis-sstc-saml-schema-assertion-1.0.xsd`	schema for validating SAML assertions
`soap_1_1_encoding.xsd`	schema for SOAP 1.1 encoding of data types
`soap_1_1_envelope.xsd`	schema for validating SOAP 1.1 envelopes
`soap_1_2_encoding.xsd`	schema for SOAP 1.2 encoding of data types
`soap_1_2_envelope.xsd`	schema for validating SOAP 1.2 envelopes
`soap_1_2_rpc.xsd`	needed by SOAP 1.2 schema
`wsdl.xsd`	schema for validating WSDL
`wsse.xsd`	schema for validating WS-Security elements
`xmldsig-core-schema.xsd`	schema for validating XML digital signatures
`XMLSchema.dtd`, `datatypes.dtd`	needed by `xmldsig-core-schema.xsd`

Note that these schemas are not automatically loaded by the WS-DBC. All required schemas need to be listed for a resource (as described above).

Enable SOAP Attachments

When checking the “Enable SOAP Attachments” box, policies for SOAP attachments may be defined, i.e., SOAP attachments of a special MIME type can either be allowed or denied by the WS-DBC Proxy, or sent to an external attachment filter (e.g., a virus scanner). If this box is not checked the WS-DBC discards all SOAP messages that carry attachments. Note that you have to define MIME actions to allow SOAP attachments to pass. As long as no action is specified, the WS-DBC will discard all SOAP messages that carry attachments.

Press the “Advanced” button next to the check box to bring up the “SOAP Attachment Properties” panel. To add a new action right-click into the MIME Actions table and select **Add MIME Action** from the context menu. Three different actions can be selected

from the drop down menu (when clicking into the actions field): allow, deny, or external. Also a MIME Type has to be defined for each action.

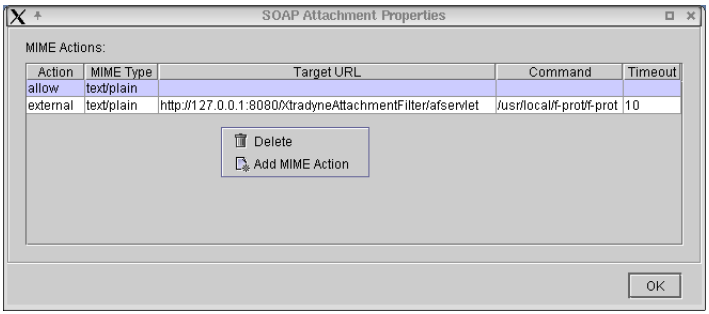


Fig. 30. SOAP Attachments Properties

allow deny action

When selecting the allow action all messages carrying an attachment of the specified type will be allowed. When selecting the deny action all messages carrying an attachment of the specified type will be denied. The list of applicable actions for an attachment's MIME type will be evaluated from top to bottom, with the last rule overriding any previous rules in case of conflict.

external action

When selecting the external action an additional Target URL, Command, and Timeout may be defined. The URL denotes an external attachment filter and the command will be evaluated by this filter. The timeout (in seconds) constrains the time the WS-DBC Proxy will wait for the external attachment filter's response. If the external call does not return in time the message carrying the attachment will be rejected.

Note that when defining different actions for the same MIME type the last entry in the table will take precedence.

1.8.5 The Xtradyne SOAP Attachment Filter Servlet

The Xtradyne WS-DBC comes with a pre-packaged WAR archive containing a servlet reference implementation that can be used to filter SOAP attachments. This WAR file is located in `<INSTALLDIR>/wsdbc/adm` and named `XtradyneAttachmentFilter.war`. The servlet itself receives HTTP POST messages from the WS-DBC Proxy, checks the content of the message by executing an external command (e.g., a virus scanner), and returns an HTTP reply that reports the outcome of the command.

Installation

The web archive can be deployed on all servlet containers that conform to the servlet specification, e.g., Apache Tomcat, BEA WebLogic, etc. After successful deployment, the servlet is reachable under the path `/XtradyneAttachmentFilter/afservlet`. When pointing your browser to that path a status page will be displayed. If the status is ok, the servlet has been successfully deployed. Please note that the servlet only accepts HTTP POST requests, so pointing your browser to it may yield an error depending on the servlet container.

The following lists the steps necessary to deploy on Apache Tomcat 4.1.30. For other servlet containers, please consult the vendors manual on how to deploy WAR files in your environment.

- Download and install Apache Tomcat as per Tomcat installation instructions.
- Stop Tomcat (if it is started).
- Copy the file `XtradyneAttachmentFilter.war` from `<INSTALLDIR>/wsdbc/adm` to the `webapps` directory of the Tomcat installation.
- Start Tomcat. The servlet should now be available under the path mentioned above.

Securing the Installation

Using the Xtradyne SOAP attachment filter servlet without additional measures imposes a security risk. The servlet itself executes any command it receives. The servlet container should therefore be configured to accept only local connections, i.e., on address 127.0.0.1 so that it is not remotely accessible. Furthermore, the servlet container should run with a non-privileged user identity with limited capabilities, i.e., **not** `root`. The third and most secure option is to modify the servlet itself to only accept a well-known set of external commands instead of arbitrary commands.

Detailed Operation of the Servlet

The following explains the servlet's inner workings and points out the functionality on which the WS-DBC Proxy relies when communicating with it. This is relevant when modifying or re-implementing your own attachment filter. The Java sources can be found under `WEB-INF/src` in the WAR file.

1. The servlet receives an HTTP POST request (mandatory).
2. The request body contains the data (attachment) that is to be filtered (mandatory).

3. The request contains an HTTP header field named “X-Attachment-Filter-Command” that carries the command to execute. The content is the string from the “Command” field of the MIME action table of the Admin Console (optional).
4. The servlet copies the HTTP message content to a temporary file.
5. The servlet executes the command that it received via the HTTP header in step 3. The name of the temporary file is appended to the command and the call is wrapped into a shell call. On UNIX, the shell is invoked using `/bin/sh -c <real command>` and on Windows `cmd /c <real command>` is used.
6. The servlet reads the `stdout` and `stderr` channels of the command and waits until it returns. The exit code is recorded.
7. The servlet assembles the HTTP reply. The HTTP reply status code is not relevant to the WS-DBC Proxy. Depending on the exit code, the servlet adds an HTTP header field named “X-Attachment-Filter-Result” with the only allowed values of “allow”, “deny” or “error” (mandatory).
8. The servlet copies the `stdout` and `stderr` output into the reply body, using the Content-Type “text/plain”. Setting the body is optional, but only bodies with Content-Type “text/plain” will be used in event notifications and log messages by the WS-DBC Proxy.
9. The HTTP reply is sent.

1.8.6 Resource Properties – Outgoing Policy

The Outgoing Policy panel defines the policy for SOAP requests and responses sent from the WS-DBC Proxy to Web Services and clients.

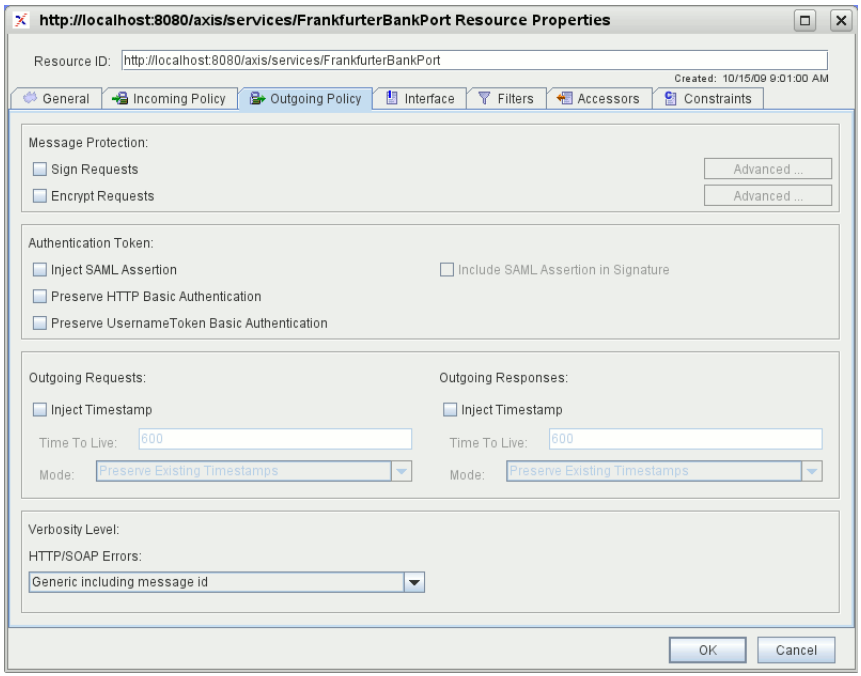


Fig. 31. Web Service Resource Properties: Outgoing Policy

Sign Requests

When checking this box the SAML assertion will be bound to the message by signing the assertion and the message body. Note that this applies only to assertions created by the WS-DBC. Existing assertions in incoming messages are not signed.

Bring up the “Advanced Configuration Properties” dialog by pressing the “Advanced” button. Configure the “Signature Creation Profile” and “Digital Signature Scopes”.

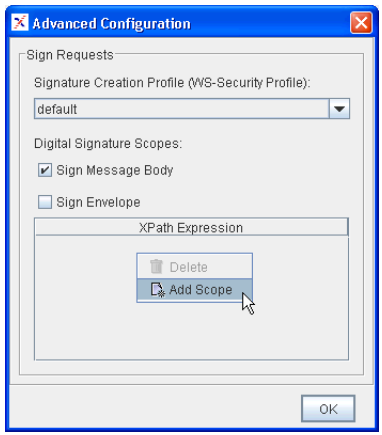


Fig. 32. Outgoing Policy: Sign Message – Advanced Configuration Properties

Signature Creation profile

Choose a “Signature Creation” profile from the drop-down menu. The profile defines the keys and certificates used to create a signature. This profile can be defined on the “WS-Security Profiles – Signature Creation” panel (for details see “SSL and WS-Security Profiles” on page 165).

Digital Signature Scopes

Digital Signature scopes define those parts of the SOAP Message that shall be digitally signed. If you check the “Sign Message Body” box the message body will be signed, if you choose “Sign Envelope” the envelope will be signed. Additionally, XPath expressions can be defined that select those parts of the message that shall be signed. To add a scope right-click into the scopes table and select “Add Scope” from the context menu.

Note that if you want SAML assertions to be signed, use the “Include SAML Assertion in Signature” checkbox in the “Authentication Token” section (see below).

Encrypt Requests

When checking this box the SOAP message body will be encrypted. Bring up the “Advanced Configuration Properties” dialog by pressing the “Advanced” button on the right side. Configure the “XML Encryption” profile and encryption scopes.

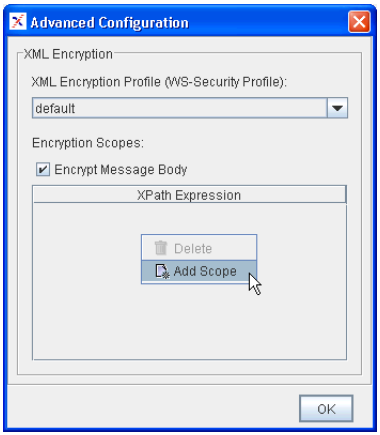


Fig. 33. Outgoing Policy: Encrypt Message – Advanced Configuration Properties

XML Encryption Keys

Chose an “XML Encryption” profile from the drop-down menu. The profile defines the keys used to encrypt the SOAP message. This profile can be defined on the “WS-Security Profiles – XML Encryption” panel (for details see “SSL and WS-Security Profiles” on page 165).

Encryption Scopes

Encryption scopes define those parts of the SOAP Message that shall be encrypted. If you check the “Encrypt Message Body” box the message body will be encrypted. Additionally, XPath expressions can be defined that select those parts of the document that shall be encrypted.

Authentication Token

The following authentication options apply to authentication tokens in outgoing messages.

Inject SAML Assertion

If this box is checked a SAML assertion will be injected into the SOAP message request. If the message already contains a SAML assertion, this assertion is verified and passed on. In this case, the WS-DBC does not inject an assertion.

If you do not check the “Inject SAML Assertion” box, the WS-DBC will not inject SAML assertions but will pass on assertions already contained in the message. This can be useful for services gathering information from SAML assertions.

Note that no SAML assertions will be injected when forwarding anonymous messages, even if the SAML forwarding policy is set.

Include SAML Assertion in Signature

When you check this box, the SAML assertion will be digitally signed (if you additionally configure that messages will be signed). This box can only be checked if the “Inject SAML Assertion” box is checked.

Preserve HTTP Authorization Header

An incoming HTTP message may carry an HTTP Authorization Header (for HTTP Basic Authentication). If you check the “Preserve HTTP Authorization Header” the WS-DBC will leave the Authorization Header in the message. If you don’t check this box the WS-DBC will remove the HTTP Authorization Header from the message.

Preserve UsernameToken Authorization Header

If the incoming request carries a UsernameToken element, this configuration option allows for removing the UT element before the request is sent to its destination. If no UT element was in the request, this option has no impact.

The specification also allows for a single username statement to be transmitted without any password. This is not supported by the WS-DBC and will be ignored. An event *XMLUsernameTokenInfo* is triggered to indicate that the WS-DBC does not trust username statements without any credentials.

Timestamp Creation

Timestamps may be created for outgoing Requests and Responses. Additionally, the Time To Live (TTL) in seconds can be defined and the mode can be selected (preserve existing timestamps or replace them).

Verbosity Levels

Define the verbosity level of HTTP/SOAP error messages that are returned to the client by the WS-DBC Proxy. Three levels are available:

- *No output at all*: A HTTP/SOAP error containing no information about the error that occured.
- *Generic including message id*: The WS-DBC Proxy generates a generic error message. For a list of generic HTTP/SOAP error messages, please refer to Appendix B, ”Error Messages and System Exceptions”.
- *Detailed*: A HTTP/SOAP error containing a detailed description of the error reason (resembling the messages in the log file).

Verbosity Levels can be defined for a resource or for a WS-DBC Proxy. Verbosity levels for a resource (on this panel) take precedence over verbosity levels defined for the WS-DBC Proxy (configurable on the WS-DBC Proxy panel).

1.8.7 Resource Properties – Interface

The Interface panel allows to define the properties of the interface including the operations the interface offers. Operations listed here may be used to define access control rules on operation level. Additionally, when the “Validate Against Interface Description” checkbox on the “Incoming Policy” tab is activated the WS-DBC will validate incoming

messages against the interface specified here. If this validation fails, the WS-DBC will block the message.

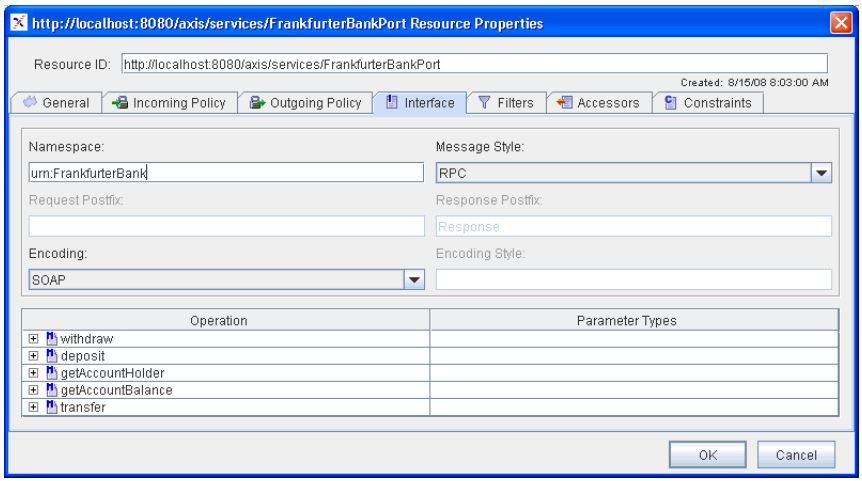


Fig. 34. Web Service Resource Properties: Interface

Interface Properties

In the upper part of this tab you can define properties of the interface, like the namespace, message style, request and response prefix, encoding and encoding style:

- **Namespace**: In the upper part of the panel you can define an optional namespace. Incoming requests and responses must use the namespace defined here.
- **Message Style**: Choose the Message Style from the drop-down menu (RPC or Document Style). By default the message style is RPC.
- **Request and Response Postfix**: Enter the postfix if required. The request postfix can only be entered when **Document Style** is chosen as message style.
- **Encoding/Encoding Style**: Choose the encoding (SOAP, Literal or Encoded). In case of **SOAP** incoming requests and responses must use the data types defined by the SOAP specification. In case of **Literal** the referenced data types must conform to the XMLSchema standard. In case of **Encoded** you can define your own data types: enter the namespace in the “Encoding Style” text field.

Note that the namespace, message style, encoding and encoding style can be defined per operation as well (see “SOAP Requests / SOAP Responses” on page 284).

Operations Table

On the left side of the operations table you see the operation name. Click on the ⊞ next to an operation to view the associated parameters. To edit an operation use the context menu which pops up when right-clicking into the table. You can choose to “Edit” or “Delete” a chosen operation, or insert a “New Operation” or “New Parameter” (Web Service Resources only).

Add/Edit an Operation

You can add or edit operations by choosing **New Operation...** or **Edit...** respectively from the context menu. This will bring up the operation properties dialog:

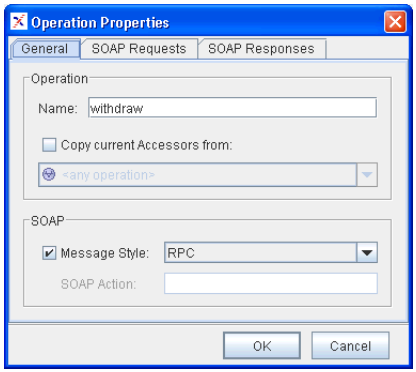


Fig. 35. Edit an operation’s parameters – General tab

Operation

copy an accessor list

In the Operation part of the “General” tab, you can enter the name of the operation for which you want to define permissions. When clicking the check box “Copy current Accessors from”, you can choose an operation from the drop-down menu. The accessors of the chosen operation will be copied to the newly created operation rule (cf. “Resource Properties – Accessors” on page 291).

SOAP

In the SOAP section of the “General” tab you can define SOAP-specific settings for the operation.

- **Message Style**: If you would like to set a different message style for the operation than for the whole resource, activate the checkbox and choose the message style

from the drop-down menu (RPC or Document Style). By default the message style is the one set for the whole resource.

- **SOAP Action**: In case of document style SOAP it might be necessary to define the SOAP action here to enable the DBC to determine the targeted operation.

SOAP Requests / SOAP Responses

On the “SOAP Requests/SOAP Responses” tab the namespace, document name, and encoding style for request /response messages for the selected operation may be defined. This may be required if namespace, document name, or encoding style of this operation’s request/response message differ from the global settings (i.e., from the settings that apply to all operations of this resource, which are defined on the Interface tab).

- **Namespace**: Define a namespace for this operation’s request/response messages To do so, activate the checkbox and enter the namespace in the text field.
- **Document Name**: If the message style is “Document Style”, you may define the document name here. Activate the checkbox and enter the document name in the text field.
- **Encoding/Encoding Style**: Select an encoding style from the drop-down box (SOAP, Literal, or Encoded). If **Encoded** is selected, define the encoding style in the corresponding text field.

Define Operation Parameters for Web Service Resources

Parameters and their types have to be defined in the policy if you want to do content filtering. Note that it is sufficient to specify those operation parameters for which filter rules will be defined.

Choose “New Parameter” in the context menu. Enter the parameter’s name in the “Name” text field of the dialog box. Choose a parameter type from the drop-down menu offered by the “Type” field.

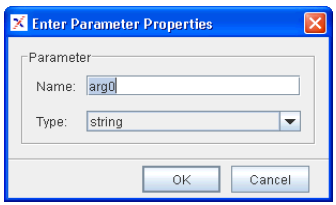


Fig. 36. Define an operation’s parameters

The “Type” drop-down menu offers a selection of parameter types as defined in the XML Schema specification. This includes different variants of strings (`normalizedString`, `Name`, `QName`, etc.), integers (`negativeInteger`, `unsignedInt`, `long`, etc.), boolean, and the type `unknown` for complex data types. Note that filters cannot be created for this parameter type. For a detailed description of XML Schema types please refer to XML Schema Part 2: Datatypes, W3C Recommendation, 2 May 2001, `http://www.w3c.org/XML/Schema`.

1.8.8 Resource Properties – Filters

content inspection

The DBC can perform content inspection by validating the messages that are sent to a service. The **Filters** tabbing pane lets you define filter rules for operations. Filter rules can be combined with the boolean operators AND and OR.

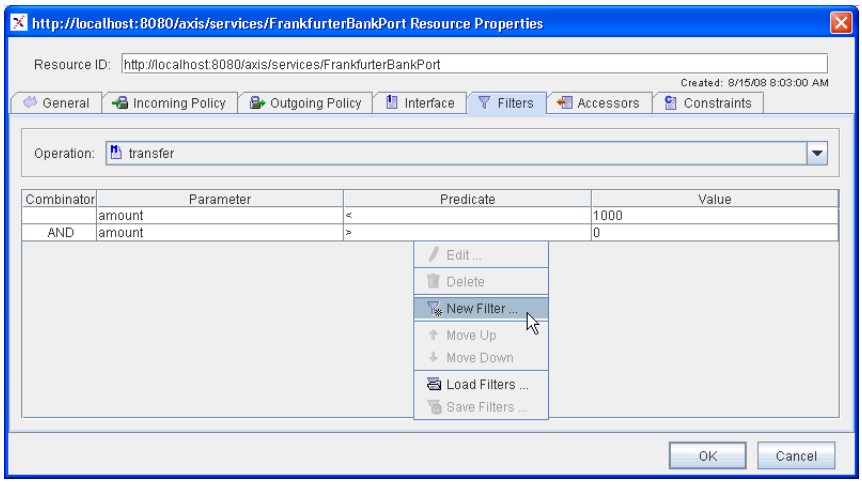


Fig. 37. Resource Properties: Defining Message Filter Rules

Note that it is not possible to create a filter expression if the operation has no parameters. In this case, import the WSDL file first (choose **File ➜ Import WSDL...**). Operation parameters can also be added manually (on the operations panel, see “Define Operation Parameters for Web Service Resources” on page 284).

Creating Filter Expressions

To define a filter rule for a certain operation select the operation from the drop down menu in the upper part of the Filters tabbing pane. Then right-click into the filter table and choose **New Filter...** from the context menu. The “Filter Properties” panel pops up (see screenshot below).

You may define filter rules that apply to all SOAP requests or responses, choose “All Requests” or “All Responses” from the drop-down menu respectively. Note that when choosing “All Requests” or “All Responses” only XPath expression may be defined as filters (cf. “XPath Expressions” on page 287).

Filter Properties

On the “Filter Properties” panel you can choose between defining a “Standard Filter Term” or an “XPath Expression” by selecting the filter type from the “Filter Type” drop-down menu. The appearance of the lower part of the panel depends on the chosen value.

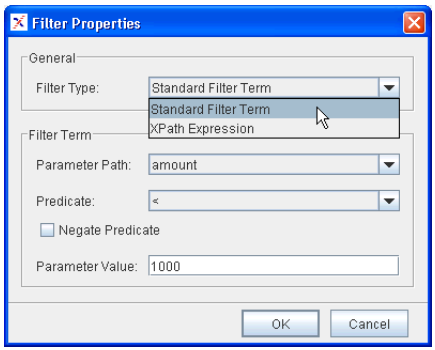


Fig. 38. Resource Properties: Defining Filter Properties

Standard Filter Terms

Choose the parameter name from the “Parameter” drop-down menu. Choose a predicate from the “Predicate” drop-down menu. Enter the parameter value into the text field (cf. screenshot above). Depending on the type of the chosen parameter, different kinds of predicates are available. The following table lists the parameter types and the available predicates for each type.

Type	Predicate	Meaning
boolean	“is”	
numeric	= =, >, <, >=, <=	
string	“equal”, “equalNoCase”,	The string equals the parameter value. The string equals, the case is not considered.
	“contains”,	The string contains the parameter value.
	“startswith”,	The string starts with the parameter value.
	“length <”, “length >”,	The string is shorter/longer than the parameter value.
	“length <=”, “length >=”	The string is shorter/longer than or equal.
	“length = =”,	The string length equals the parameter value.
	“match”	The string matches the parameter value.

When selecting the predicate “match” the value must be a regular expression. For example, the regular expression “XD_.*” will match all strings that start with “XD_”. For a detailed description of XML regular expressions please refer to “WS-DBC – Regular Expression Syntax” on page 301.

regular expressions

Note that it is not possible to create a standard filter expression if no parameters have been defined for an operation. In this case import the WSDL file first (choose **File ➜ Import Resource from WSDL...**) or define operation parameters on the Interface panel (cf. “Resource Properties – Interface” on page 281). Also note that it is always possible to define XPath expressions (see next section).

XPath Expressions

The XML Path Language (XPath)[1] supports addressing parts of an XML document. XPath *expressions* are evaluated against a document’s logical structure to identify a set of nodes that represent document entities. The set of nodes may further be constrained to meet certain selection criteria.

The WS-DBC supports the definition of XPath expressions as filters for request and response messages. These filters can be defined for all incoming request and response messages, or for messages targeting a certain operation.

1 W3C: XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999, `http://www.w3.org/TR/xpath`

The messages will be rejected if the XPath filter expressions do not match, i.e., does not yield a positive result. The result of evaluating the XPath expression is converted to boolean value as defined in the XPath specification. This is equal to using the boolean function of XPath directly like `boolean(<original xpath expression>)`. In case a response message from a Web Service to a client is rejected, an error message is returned to the client (rather than to the server).

When defining XPath expressions as filters, you need to know the general layout of the XML messages you are targeting. Also, you need a good understanding of the XPath language syntax because expressions are entered textually rather than visually and the Admin Console does not provide syntax checking of these expressions. Ill-formed expressions will only be detected at run time. We recommend using the `xpathtest` tool to check XPath expressions syntactically and semantically before using them in the WS-DBC. For a detailed description of this tool, please refer to Chapter 5 “WS-DBC Tools” on page 49 of the Deployment Guide.

defining XPath expressions

To define XPath expressions, select **New Filter ...** from the context menu and choose “XPath Expression” in the “Filter Type” drop-down menu. Fill in the XPath expression in the text field below.

Combining Filter Expressions

Each row of the “Filter” table shows a filter expression (these can be standard filter expressions or XPath expressions). These expressions can be combined by using the boolean operators (AND and OR). To enter/modify the operator, click into the “Combinator” field and choose the appropriate combinator from the drop-down menu. Filter rules are evaluated from top to bottom. AND and OR have the same precedence.

Example

An expression A AND B OR C AND D, where A, B, C and D are simple filter expressions is evaluated like this: (((A AND B) OR C) AND D).

Filter Templates

You can define filter templates by selecting one or more filters from the list and saving them as a filter template: Choose **Save Filters ...** from the context menu. You can name each filter template and load it for different operations.

1.8.9 Exporting and Importing Filter

Filter definitions can be exported into a file or imported from a file respectively.

exporting filters

To export filter definitions, choose **File ➜ Export ➜ Export Filters...** from the menu bar. The Export Filter Wizard will open up. You can select the resources for which filter definitions shall be exported. In the next panel you can select a file into which the filters will be saved (by default filters will be saved in a file with a `.config` extension).

importing filters

To import filter definitions, choose **File ➜ Import ➜ Import Filters...** from the menu bar. The Import wizard will open up. On the first panel you can choose the file from which the filter definitions will be imported (by default `.config` extension). On the next panel the wizard will list the resources found in the file. Select the resources for which filters shall be imported and press the “Finish” button.

If the resource for which the filter shall be imported does not exist in the model the wizard will show a list of available resources. You can either select a resource from the list or press the “Cancel” button to skip filter import for this resource.

Note that existing filters will be replaced by imported filters.

Note that filters can only be imported when the operation name, the parameter name and the parameter type are identical. If this is not the case filters for this method are not imported. Existing filters for this operation remain untouched.

When the filter import is complete, the wizard will show the “Filter Import Report”. This report lists which filters have been imported successfully and which resources or operations have been skipped and why.

1.8.10 Resource Properties – Accessors

The previous sections explained how protection and authentication requirements for a resource are specified. On the Accessors tabbing pane you can define which entities (users, groups, or roles) are granted access to a resource. Access can be granted to an entire resource, i.e., to all its operations at once or in a more fine-grained fashion: different access rights can be defined for the different operations of a resource. These operation-specific rules override the resource default, i.e., you may define stricter or *less* (!) strict rules.

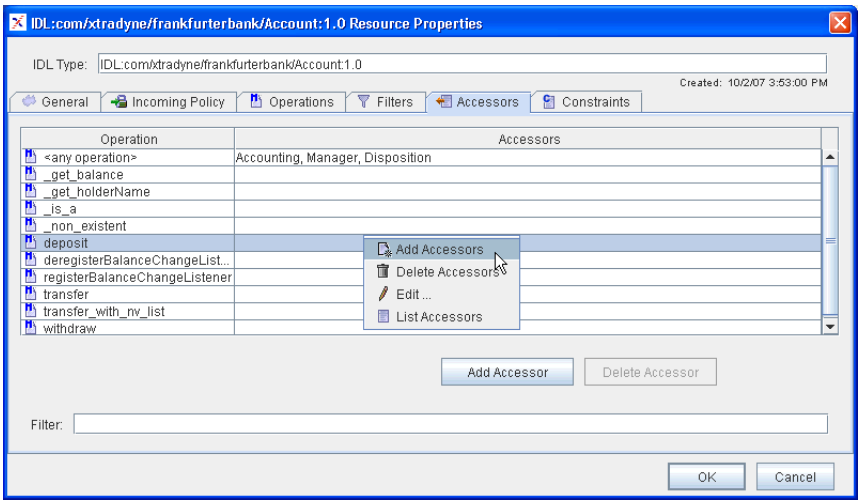


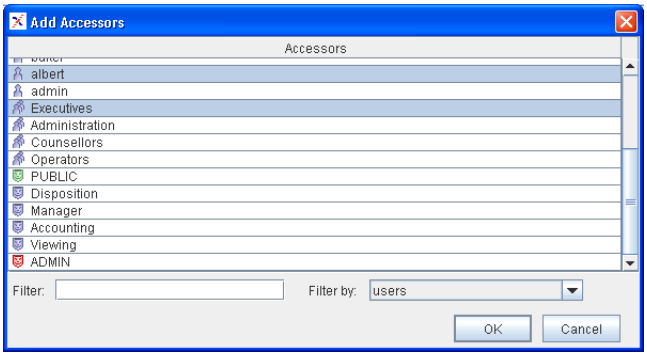
Fig. 39. Resource Properties – Accessors

Configuring Accessor Lists

adding accessors

On the Accessors tabbing pane a list of all operations defined for this resource and their corresponding accessors is displayed (see Figure 39). Double-click on the operation you want to define accessors for or select the operation and press the “Add Accessors” button (more than one operation can be selected at a time). A new panel will pop up which shows a list of the existing users, groups, and roles. Select those entities from the list that you want to grant access to the resource. For convenience, you can sort the list by clicking on the “Accessors” column. You may also use the filter and select the entity type

(users, groups, roles, or all) to which the filter shall apply. Resource Properties – Selecting Accessors



After pressing the “OK” button, the selected entities will appear in the accessors list of the corresponding operation on the left side of the panel.

removing accessors

If you want to remove accessors from the list, press the “Delete Accessors” button. The add and remove accessors actions are also accessible via the context menu. The context menu also offers to edit accessors – the selected accessors will overwrite a previous accessor list.

Note that the choice given to the user which users, groups, or roles shall be added or removed from the list of accessors depends on which users, groups, or roles are already accessors of this resource. For example, if a user is part of the accessors list of this resource, this user is not listed when adding accessors. Also note that the buttons for adding and removing accessors are disabled when the list would be empty anyway.

listing accessors

Selecting “List Accessors” from the context menu brings up a dialog which displays a list of this operation’s accessors. This may be useful when many accessors are defined and you are looking for a specific entry. The displayed list of accessors can be sorted and filtered for better handling.

same access rights for all operations

If you want to assign the same access rights for all operations of a resource no special operation rules have to be defined. Select `<any operation>` in the operation list and define accessors as described before.

Note that the `<any operation>` accessor list applies only to operations that are **not** explicitly defined in the list of operations (on the Operations tabbing pane). Delete those operations from the list of operations for which the `<any operation>` access rules shall apply. Note that if you delete operations for which parameter filters were defined, these filter rules will be lost.

Public Access

To grant public access to a resource, you can add the role "PUBLIC" to the accessor list. This is usually useful for bootstrapping like access to a naming service, etc.

PUBLIC Role

Note that adding the role "PUBLIC" to the `<any operation>` accessor list implies that all operations which are not defined in the Admin Console will be granted public access by the DBC! It is not recommended to do this! Instead you should leave the `<any operation>` accessor list empty, thus denying access for operations that are not explicitly mentioned.

1.9 Applications (Application Domains)

Application domains group roles and resources into one logical unit. This is useful when several different services are administered with the Admin Console. For each service (resource) you can create an application and add the associated roles to this application. Apart from gaining a more structured overview you can additionally configure application-specific administrative rights which enforce access control on the policy itself. In this way, you can have several administrators working with the Admin Console but these administrators can be given the rights to configure only a certain application.

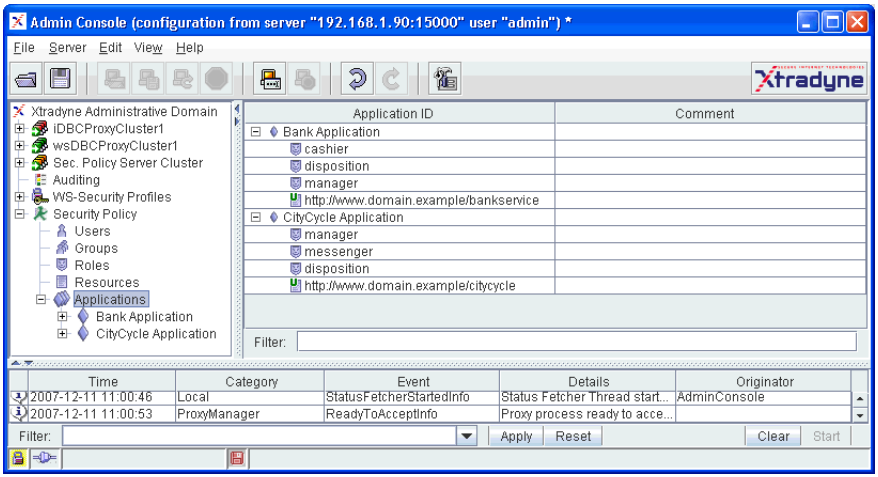


Fig. 40. Application Domains

Adding Applications

To add a new application right-click into the applications table or on the “Applications” item in the navigation tree on the left part of the Admin Console. Choose **New Application** from the context menu. Click on the ⊞ next to the application to view the roles and resource of the application.

Application Roles and Resources

The role and resource panels of applications are exactly the same as the roles and resources panels presented before. Please refer to section “Roles” on page 258 and section “Resources” on page 259 for an explanation of these panels.

Application Properties – General

To bring up the “Application Properties” panel right-click on the application in the application table and choose **Edit Application Properties...** from the context menu.

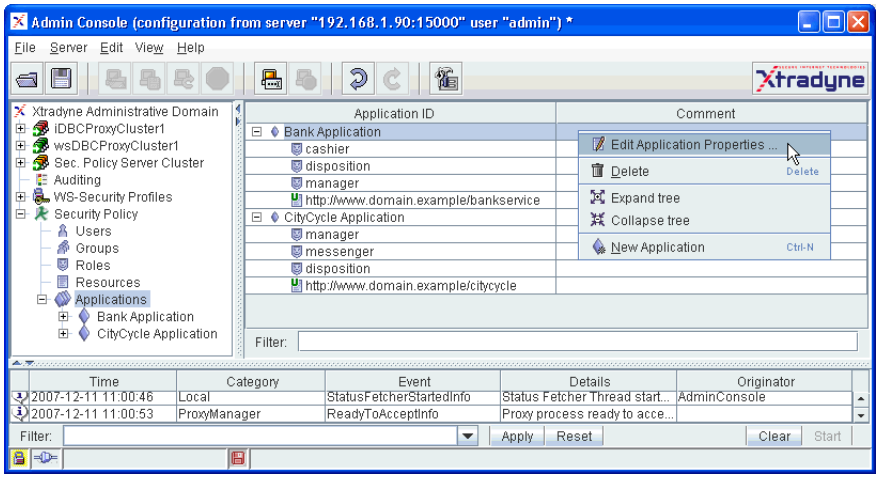


Fig. 41. Application Domains - Edit Properties

The Application Properties panel has three tabs. On the “General” tab a comment can be entered and on the “Administration” tab administrative rights can be configured.

Application Properties – Administration

On the “Administration” tab you can configure administrative rights for the application. We have already seen how to configure global administrative rights on the Role Properties – Administration panel (see page 228 for details), where rights for accessing all parts of the configuration file can be configured (e.g. the right to delete or add a Proxy).

In contrast to that, the administrative rights that can be configured for applications apply only to those parts of the configuration that are specific for that application (e.g. create or delete roles within a certain application).

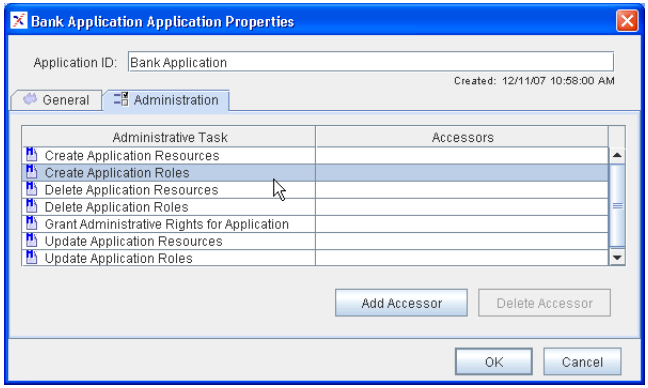


Fig. 42. Application Domains - Administration

administrative rights for applications

Configure which user, group, or role will be allowed to administer the application. Choose one of the following administrative tasks from the drop-down menu in the upper panel:

- Create Roles
- Update Roles
- Delete Roles
- Create Resources
- Update Resources
- Delete Resources
- Grant Administrative Rights

Note that to rename an entity (application, role, or resource) add and delete rights are required.

Assign an existing user, group, or role from the window on the right side of the panel and press the “Add” button. The selected entities will now appear in the “All Accessors” list on the left side of the panel.

Note that the user that creates the application will by default be allowed to perform all administrative tasks. But of cause different administrative rights can be configured after creating the application.

Example

There are two services (“Service1” and “Service2”) that shall be administered with the Admin Console. You can create an application for each service (as depicted in the figure below): “Application1” groups together the resource (“Service1”) and roles (e.g. “Role”, “App1Role1”, and “App1Role2”). “Application2” groups together the resource (“Service2”) and roles (e.g. “Role”, “App2Role1”, and “App2Role2”).

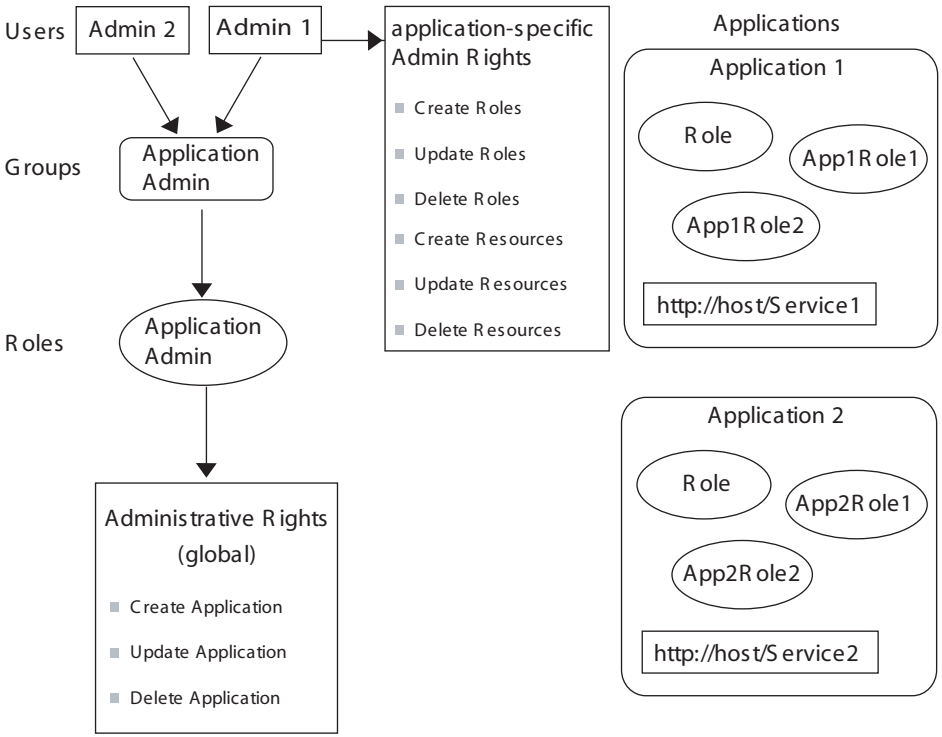


Fig. 43. Example Application Domains and administrative rights

There are two persons in charge of the administration of these two services (“Admin1” is in charge of “Application1” and “Admin2” is in charge of “Application2”). However, “Admin1” is not allowed to edit anything that is outside of “Application1” and vice-versa.

Both administrators are put in the group “Application Admin” which is assigned to the role “Application Admin”. For this role the administrative rights to create, update, and delete an application are granted (on the Roles – Administration panel).

Now the application-specific access rights are configured: On the “Application – Administration” panel of “Application1” the user “Admin1” is granted the administrative rights to create, update, and delete roles and resources. On the “Application – Administration” panel of “Application2” the user “Admin2” is granted the administrative rights to create, update, and delete roles and resources.

Note that roles are scoped for application domains, i.e., there can be different roles of the same name in different applications (e.g. the role “Role” in the example above). To uniquely identify roles they are prefixed with the application for which they are defined, e.g. Role@Application1. In contrast to that resources have to be unique in the configuration. Also note that a globally defined role (on the “Security Policy – Roles” panel) can be configured as accessor of a resource defined in an application.

Moving Global Roles and Resources to an Application

If you have defined global Roles and Resources and would like to put these into an application you may use the Cut and Paste mechanism in the Admin Console: Create a new application, then select the global roles you would like to move and choose **Cut** from the context menu. Go to the application’s roles panel and paste the roles there. Repeat the same steps for the resources you would like to move. All relations between roles and resources will be preserved, i.e., they will be updated to fit the entities’ new location in the application.

CHAPTER

2 *Regular Expressions*

Regular expressions can be used to define filter rules for parameter checking. The following section describes regular expressions understood by the WS-DBC.

2.1 WS-DBC – Regular Expression Syntax

A regular expression describes strings of characters. It's a pattern that matches certain strings and doesn't match others. The regular expressions used in the WS-DBC for parameter checking follows the W3C standard for XML regular expressions (as defined in XML Schema Part 2: Datatypes, Appendix F: Regular Expressions, W3C Recommendation, 2 May 2001, `http://www.w3c.org/XML/Schema`).

A regular expression consists of zero or more branches, separated by '|', matching anything that matches any of the branches. A branch is zero or more quantified atoms, concatenated. It matches a match for the first, followed by a match for the second, etc.; an empty branch matches the empty string.

2.1.1 Quantifiers

A quantified atom is an atom possibly followed by a single quantifier. Without a quantifier, it matches the atom. The quantifiers, and what a so-quantified atom matches, are:

- * a sequence of 0 or more matches of the atom,
- + a sequence of 1 or more matches of the atom,
- ? a sequence of 0 or 1 matches of the atom,
- {m} a sequence of exactly m matches of the atom,
- {m,} a sequence of m or more matches of the atom,
- {m,n} a sequence of m through n (inclusive) matches of the atom; m may not exceed n.

The forms using '{' and '}' are known as bounds.

Example

The '?' makes the preceding token in the regular expression optional. E.g.: 'colou?r' matches both "colour" and "color".

The '*' tries to match the preceding token zero or more times, i.e., the expression 'xtra*' will match for the string "xtr", "xtra", and "xtraaaa". Evaluating the string "xtradyne" against the regular expression 'xtra*' will also yield 'true', because "xtra" is a substring of the "xtradyne" and the regular expression is unterminated.

2.1.2 Atoms

An atom is either:

- A *normal character*: A normal character is any XML character that is not a metacharacter. Metacharacters are: ., \, ?, *, +, {, }, (,), [,]. These metacharacters have

special meanings in regular expressions, but can be escaped with a \ to be treated like a normal character.

- A *character class*: a list of characters enclosed in '[]' (see next section),
- A *parenthesized regular expression*: '('`regExp`')', where `regExp` is any regular expression.

Example

You can make several tokens optional by grouping them together using round brackets, and placing the question mark after the closing bracket. E.g.: 'Oct(ober)?' will match "Oct" and "October".

2.1.3 Character Classes

A character class is a list of characters enclosed in '[]'. It normally matches any single character from the list. If the list begins with '^', it matches any single character (see below) not from the rest of the list.

If two characters in the list are separated by '-', this is shorthand for the full range of characters between those two, e.g., '[0-9]' matches any decimal digit. Two ranges may not share an endpoint, so, e.g., 'a-c-e' is illegal.

Character classes can be subtracted from one another, for example: [0-9] - [4-9] = [0-3].

Example

The expression '[1-9][0-9]{3}' defines a number between 1000 and 9999.

2.1.4 Single Character Escapes

Escapes begin with a '\' followed by an alphanumeric character, Character-entry escapes exist to make it easier to specify non-printing and otherwise inconvenient characters in regular expressions:

- \n the newline character (#xA),
- \r the return character (#xD),
- \t the tab character (#x9)
- \char `char`, where `char` is one of: \, |, ., -, ^, ?, *, +, {, }, (,), [,].

Example

The regular expression `[^\n]` matches any character that is not a new line. Remember that in a character class expression the '^' at the beginning of the list matches any single character not included in the rest of the list.

2.1.5 Category Escapes

The set containing all characters that have property X can be identified with a category escape `\p{X}`. The complement of this set is specified with the category escape `\P{X}`, so that `[\P{X}] = [^\p{X}]`. The table below lists the recognized character properties.

Category	Property	Meaning
Letters	L	Letter, Any
	Lu	Letter, Uppercase
	Ll	Letter, Lowercase
	Lt	Letter, Titlecase
	Lm	Letter, Modifier
	Lo	Letter, Other
Marks	M	Mark, Any
	Mn	Mark, Nonspacing
	Mc	Mark, Spacing Combining
	Me	Mark, Enclosing

Category	Property	Meaning
Numbers	N	Number, Any
	Nd	Number, Decimal Digit
	Nl	Number, Letter
	No	Number, Other
Punctua-tion	P	Punctuation, Any
	Pc	Punctuation, Connector
	Pd	Punctuation, Dash
	Ps	Punctuation, Open
	Pe	Punctuation, Close
	Pi	Punctuation, Initial quote (may behave like Ps or Pe, depending on usage)
	Pf	Punctuation, Final quote (may behave like Ps or Pe, depending on usage)
	Po	Punctuation, Other
Symbols	S	Symbol, Any
	Sm	Symbol, Math
	Sc	Symbol, Currency
	Sk	Symbol, Modifier
	So	Symbol, Other
Separators	Z	Separator, Any
	Zs	Separator, Space
	Zl	Separator, Line
	Zp	Separator, Paragraph
Other	C	Other, Any
	Cc	Other, Control
	Cf	Other, Format
	Cs	Other, Surrogate (not supported by Schema Recommendation).
	Co	Other, Private Use
	Cn	Other, Not Assigned (no characters in the file have this property).

2.1.6 Block Escape

Block escapes can be used to identify sets of ASCII characters, for example, `\p{isBasicLatin}` denotes basic latin characters. For a complete list of block escape values, please refer to XML Schema Part 2: Datatypes, Appendix F: Regular Expressions, W3C Recommendation, 2 May 2001, `http://www.w3c.org/XML/Schema`.

2.1.7 Multi-Character Escapes

Multi-character escapes provide shorthands for certain commonly-used characters:

Escape	Shorthand for	Description
.	[^\n\r]	Any character except \n (new line) and \r (return).
\s	[#x20\t\n\r]	Whitespace, i.e., #x20 (space), \t (tab), \n (newline), and \r (return).
\S	[^\s]	Any character except those matched by \s.
\i		First character in an XML identifier, i.e., any letter (\p{L}), the character "_", or the character ":".
\I	[^\i]	Any character except those matched by \i.
\c		Any character that might appear in the built-in NMTOKEN datatype (see the XML Recommendation for the complex specification of a NameChar).
\C	[^\c]	Any character except those matched by \c.
\d	\p{Nd} or [0-9]	Any decimal digit.
\D	[^\d]	Any character except those matched by \d.
\w	[#X0000-#x10FFFF]-[\p{P}\p{Z}\p{C}]	Any character that might appear in a word (all characters except the set of \p{P}(punctuation), \p{Z}(separators), and \p{C}(other characters)).
\W	[^\w]	Any character except those matched by \w.

Example

The regular expression `[0-9]{1,3}` is equivalent to `\d{1,3}`.

2.1.8 Examples

This section gives some examples on how to define regular expressions.

Define a Regular Expression for an IP-Address

IP addresses are commonly expressed as four decimal numbers, each representing a byte value, separated by periods, for example: '205.245.172.72'. A regular expression that describes an IP-address can be defined as:

```
\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
```

`\d{1,3}\.` states that we expect 1 to 3 digits followed by a period. Instead of `\d` you can also use the shorthand `[0-9]` or `\p{N}` (character property "Number", see "Category Escapes" on page 304).

Define a Regular Expression for a Date

Let's define an expression for a special date format, for example: YYYY-MM-DD. A regular expression for this date format would be:

```
\d{4}\-\d{2}\-\d{2}
```

This expression defines that there must be four digits followed by a dash, then 2 digits followed by a dash and then 2 more digits. A string that matches this expression is, for example, "2003-12-12".

Define a Regular Expression for a Floating Point Number

A floating point number consists of an optional sign, that is followed by zero or more digits followed by a dot and one or more digits (a floating point number with optional integer part, e.g., -3.56, +.95). The regular expression for this is:

```
[\-\+]?[0-9]*\.[0-9]+
```

Floating point numbers also include integers, e.g., +3. The regular expression for this is:
`[\-\+]?[0-9]+`

Combining the two definitions with '|' (or), we can write:

```
[\-\+]?([0-9]*\.[0-9]+|[0-9]+)
```

We can modify this regular expression to also matches floating point numbers like 1e3, 1., and .1 (this time using '\d' instead of '[0-9]' to denote digits):

```
[\+\-]?(\d+\.?\d*|\.\d+)([eE][\+\-]?\d+)?
```

Regular Expressions for Sets

Let's say we have a set of strings and want to express that the value must be one of this set. The set is, for example, either "0", "55" or a string starting with at least one "a". The corresponding regular expression would look like this:

```
((0)|(55)|(a.*))
```

Note that `.*` matches any sequence of characters, even the sequence of length 0 (values that match the above expression are, for example: "0", "55", "a", "ab1", "abb2").

CHAPTER

3 Protecting Web Services – Example

This chapter presents the example Web Service "Frankfurter Bank". This is a simple fictitious bank application included in the DBC installation to illustrate different features of the WS-DBC.

The basics of the Frankfurter Bank application are presented in the Quick Installation Guide ("Protecting Web Services – Example" on page 19). This chapter assumes that you are already familiar with the Frankfurter Bank application, i.e., the topics explained in the Quick Installation Guide:

- installing and running the Frankfurter Bank application,
- running the application across the WS-DBC (defining a resource mapping),
- protecting the application with the WS-DBC by defining a simple access control policy including the definition of:
  - users (in our example the users `meyer` and `bauer`)
  - different authentication mechanisms
  - access rights for the whole resource and single operations

In the following we will cover the following topics:

- model engineering with groups and roles ("Model Engineering" on page 310),
- defining parameter filter rules ("Content Filtering" on page 314).

3.1 Model Engineering

Recall the simple model from the Quick Installation Guide (depicted in figure 44). We created user representations in our policy for the two account managers Mrs. Meyer and Mr. Bauer. Then we granted these users direct access to the Frankfurter Bank resource.

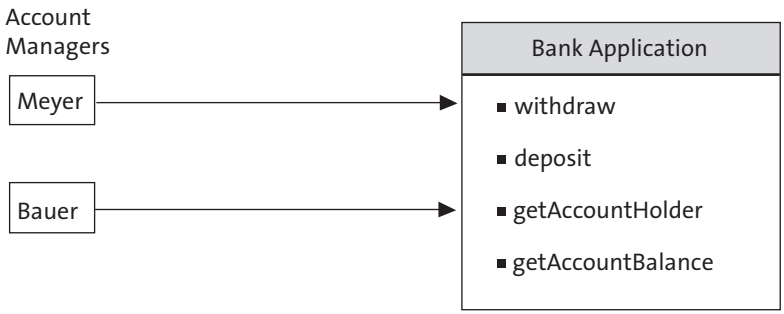


Fig. 44. Simple Frankfurter Bank Model

Using this "short-cut" is not the recommended way to do so but was chosen in the Quick Installation example to have a quick demonstration. The DBC's access control policy comprises four components: Users, Groups, Roles, and Resources. In the Quick Installation example we have seen that resources represent services. Roles represent real-life tasks and receive the necessary *permissions* to invoke *operations* on resources to fulfill these tasks. Users, on the other hand, can be put into groups (representing, for example, teams, departments, etc.).

We recommend that users are not directly granted access to a resource, but are assigned to groups that receive their permissions from roles. For flexibility, short-cuts (as used in the Quick Installation example) are allowed.

Extended Model

In our extended example, Mrs. Meyer was offered the position of a bank counsellor. Counsellors give advice to bank customers on how to invest their money and don't deposit and withdraw but just view the account data.

To reflect this in our model, we define the groups:

- `Counsellors` with Mrs. Meyer as member and
- `Operators` with Mr. Bauer as member,

Then we define the roles:

- `Viewing` and give the group `Counsellors` privileges for this role, and
- `Accounting` and give the group `Operators` privileges for this role.

The extended model is depicted in figure 45.

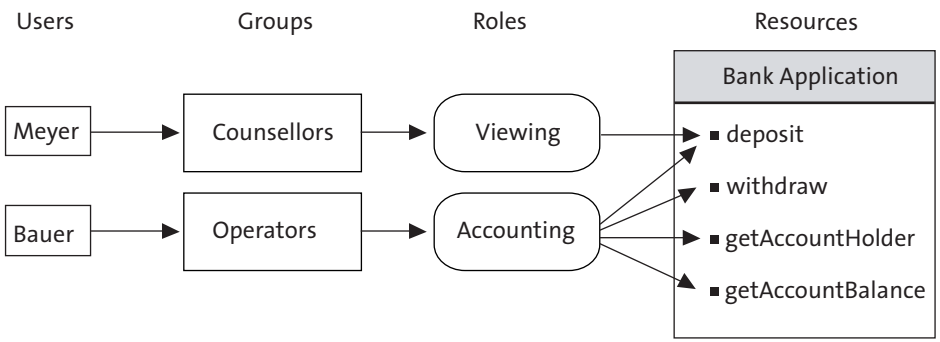


Fig. 45. Simple Model with groups and roles

Define Groups

Let's use the Admin Console to define this model. The two users with the IDs `meyer` and `bauer` were already defined in the quick installation example, so we will start by defining the groups. Open the groups pane (below the "Security Policy" node in the navigation tree) and create a new group via the context menu or via **Edit➜New Group**. The Group ID is `Operators`. Go to the **Members** tabbing pane and select `bauer` in the list on the right (as depicted in the screenshot below).

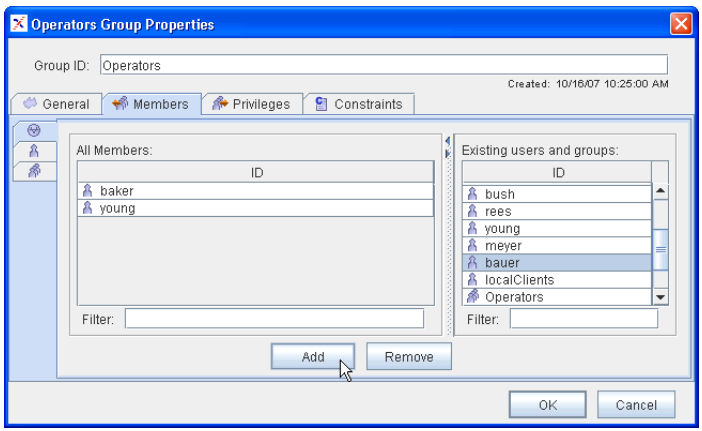


Fig. 46. Add a member to the group `Operators`

Click the “Add” button to add the user as a member of the Operators group (You may select multiple entries in the list while pressing the CTRL-key or a range of entries by pressing the SHIFT-key). Click the “OK” button. Now create the group Counsellors and add user meyer as member.

Define Roles

Now open the roles pane (below the “Security Policy” node in the navigation tree) and create a new role by using the context menu on the role list area or via the menu **Edit➜New Role**. Enter Accounting as role ID. Then go to the Actors tabbing pane, select the group Operators and add them to the actors list (as depicted in the screenshot below). Now create the role Viewing and add the group Counsellors as actor.

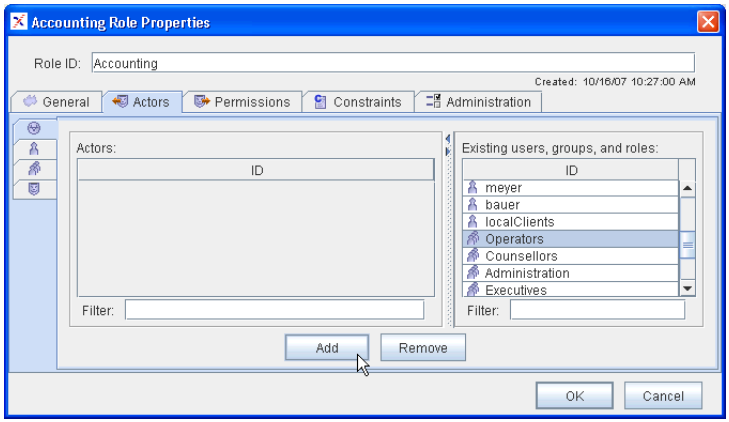


Fig. 47. Add an actor to the role Accounting

Define Accessors to the protected Resources

As a last step we want to define accessors to the protected resources. Recall the quick installation example: We imported the Frankfurter Bank WSDL into the Admin Console. The Frankfurter Bank resource (cf. figure 48) shows up in the Admin Console’s list

of protected resources (select “Resources” below the “Security Policy” node in the navigation tree).

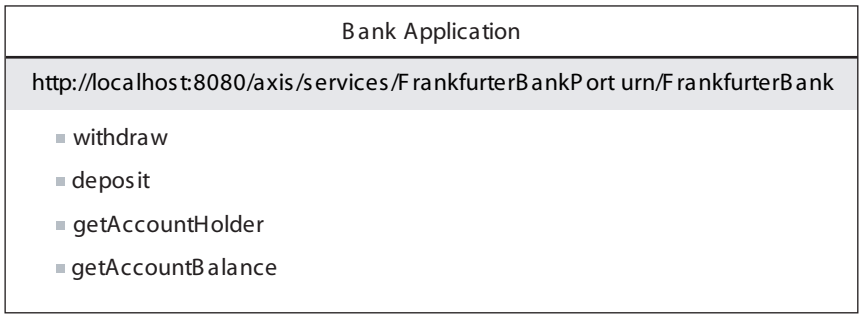


Fig. 48. Frankfurter Bank WSDL

In our policy we want to define the following: The role `Viewing` shall only be allowed access to the operations `getAccountHolder` and `getAccountBalance`. The role `Accounting` shall be allowed full access to all operations. To define this open the resources pane. Double-click on the `FrankfurterBank` resource and go to the Accessors tabbing pane (see screenshot depicted in figure 49).

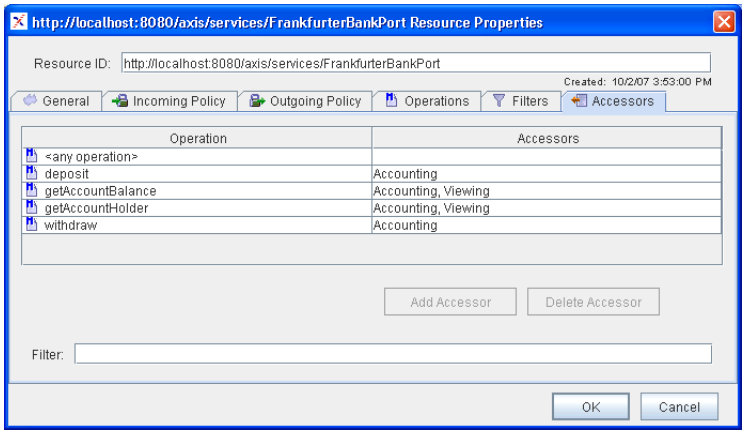


Fig. 49. `FrankfurterBank` Resource Accessor list

define an accessor list

Allow the role `Accounting` access to all operations of the `FrankfurterBank` resource. To do this, select all operations in the list and then, press the “Add Accessors” button. A new panel will pop up which shows a list of the existing users, groups, and

roles. Select the role Accounting and press the “Add” button (as depicted in the screenshot below). Remove all other accessors from the accessors list.

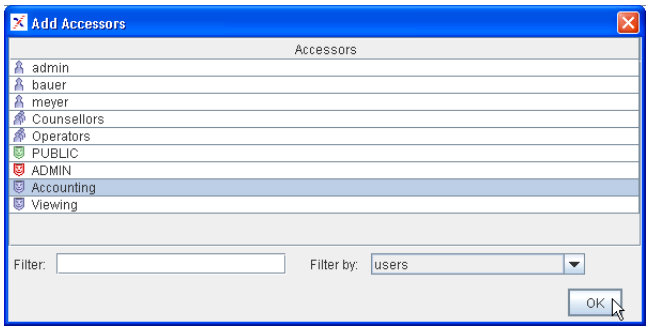


Fig. 50. Add role Accounting as accessors

Now give the two role Viewing access to the operations getAccountHolder and getAccountBalance. Save the configuration to the Security Policy Server (**CTRL-W** or **Server - Write to Security Policy Server**).

We suggest that you play around a bit with the example. If you configured the security policy as suggested Mr. Bauer should be allowed to view accounts and deposit and withdraw money. Mrs. Meyer should be only allowed to view account data. Try to access an operation for which Mrs. Meyer has no access granted, for example, withdraw money and confirm that the “No Permission” window will appear.

3.2 Content Filtering

In this section we will give an example on how to define parameter filter rules. The Frankfurter Bank application has been extended to provide an operation for transferring money between accounts. To get the extended user interface, start the Frankfurter Bank client, go to the “Configuration” tab, check the “Show Transfer Panel” box and press “OK”. Select an account (e.g. “1234”) and proceed.

A “Transfer” tabbing pane will show up on the account data panel for the transfer operation (cf. screenshot below).

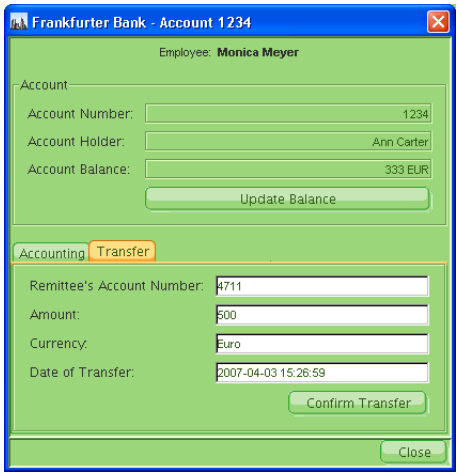


Fig. 51. Frankfurter Bank client with transfer panel

Parameter Filter Rules for the `transfer` *operation*

Let’s look at the definition of the `transfer` operation. It has five parameters: `currentAccount` of type `int`, `amount` of type `int`, `currency` of type `string`, `account` of type `int`, and `date` of type `string`.

We want to define a filter rule for the `transfer` operation which states that:

- the currency has to be Dollar or Euro (case insensitive),
- the allowed maximum amount depends on the currency (1000 EUR or 1500 $),
- the date must be 19 characters long.

Bring up the resource properties panel of the `Account` resource and go to the “Filters” tab. Select the `transfer` operation from the drop-down menu (cf. figure 52).

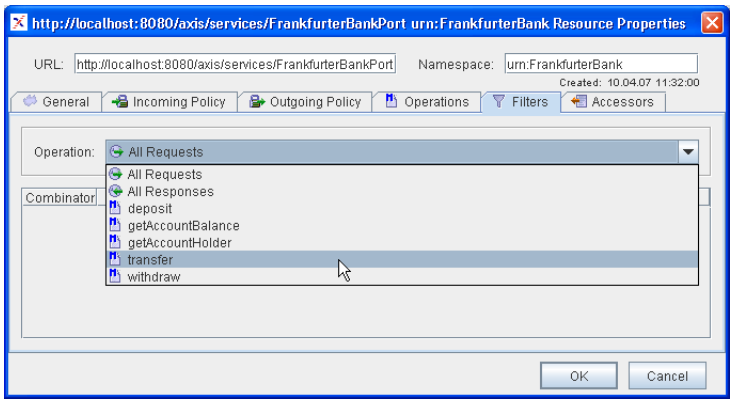


Fig. 52. Frankfurter Bank Example: Select the `transfer` operation

Then right-click into the filter rules area and select “New Filter” from the context menu. This will bring up the Filter Properties dialog (cf. figure 53).

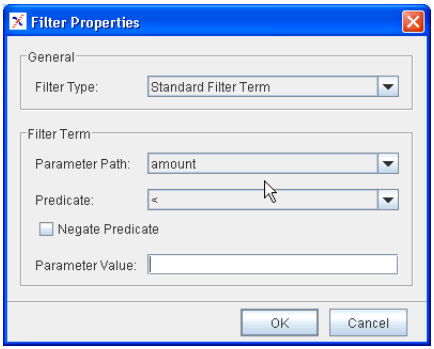


Fig. 53. Filter Properties dialog

The first filter rule shall state that the maximal amount that can be transferred in euro is 1000. Select “<” from the “Predicate” drop-down menu, enter “1000” in the “Parameter Value” field and press the “OK” button. The rule appears in the filter list:

Combinator	Parameter	Predicate	Value
	amount	<=	1000

Define the next part of the rule (currency is euro and date is 19 characters long). By default, when adding a new rule, the combinator is AND, which is correct here. The rule list now looks like this:

Combinator	Parameter	Predicate	Value
	amount	<=	1000
AND	currency	equalNocase	euro
AND	date	length==	19

The next step is to define a rule that states, that the maximum amount that can be transferred in dollars is 1500. Hint: To change the combinator, click into the combinator field and select OR from the drop-down menu.

The complete filter rule looks like this:

Combinator	Parameter	Predicate	Value
	amount	<=	1000
AND	currency	equalNocase	euro
AND	date	length==	19
OR	amount	<=	1500
AND	currency	equalNocase	dollar
AND	date	length==	19

Save the configuration to the Security Policy Server, start the Frankfurter Bank client and server and verify that the policy is enforced by the DBC.

Parameter Filter Rule for the Date Format

As a next step we want to ensure that only dates of the format `YYYY-MM-DD HH:MM:SS` are allowed. For this purpose we will use the "match" predicate and define a regular expression for the date parameter.

The year, month, and day part consists of four digits followed by a dash then two digits followed by a dash and another two digits followed by a space – as a regular expression this is:

```
[0-9]{4}-[0-2][0-9]-[0-3][0-9]\s
```

The hour, minute and second consist of two digits followed by a colon (has to be escaped) – as a regular expression this is:

```
[0-2][0-9]\:[0-5][0-9]\:[0-5][0-9]
```

To make a string match the exact string (without any other preceding or trailing characters) the regular expression is put in brackets, started with a `^` and ended by a `$`.

The complete regular expression to match the given date format is:

```
^([0-9]{4}-[0-2][0-9]-[0-3][0-9]\s[0-2][0-9]\:[0-5][0-9]\:[0-5][0-9])$
```

A complete description of regular expressions understood by the DBC is given in “Regular Expressions” on page 301.

To configure the above regular expression for the date parameter in the Admin Console, choose the match predicate for the date parameter and enter the regular expression in the value field. Combined with the rules defined before, the filter list now looks as depicted below.

Combinator	Parameter	Predicate	Value
	amount	<=	1000
AND	currency	equalNocase	euro
AND	date	length==	19
AND	date	match	^([0-9]{4}-...)$
OR	amount	<=	1500
AND	currency	equalNocase	dollar
AND	date	length==	19
AND	date	match	^([0-9]{4}-...)$

Fig. 54. Filter list with date rule

PART

4 *APPENDICES*

This part includes appendices that might be helpful to explain certain issues in more detail:

- Appendix A “Audit Events” on page 321 lists all events that can be generated by the WS-DBC and explains for what they are used.
- Appendix B “Error Messages and System Exceptions” on page 339 lists HTTP and SOAP error messages that are received by the client and WS-DBC error log messages.
- Appendix C “SSL Ciphers” on page 355 gives a description of recommended SSL Cipher suites.

APPENDIX

A Audit Events

Audit events are very helpful to detect misconfigurations as well as, for example, finding out if an unauthorized user tries to gain access to a protected resource. The first section lists all events in alphabetical order with a short description. The second section lists the most important failure events and discusses the possible reasons for these failures.

A.1 Events in alphabetical order

The following table lists audit events for all Xtradyne products that can be configured with the Admin Console - i.e., the I-DBC and the WS-DBC. The I-DBC is the IIOP Domain Boundary Controller, Xtradyne's DBC for CORBA, and the WS-DBC is the Web Services Domain Boundary Controller, Xtradyne's DBC for Web Services.

hidden events

Some audit events are not listed in the Admin Console for configuration. These events are called *hidden* events. Hidden events mostly indicate fatal error conditions and should not occur in day-to-day operation.

Note that in the *Category* which is displayed in a separate column in the event browser, is part of the event name, i.e., when you look at the event log (e.g., log file) a *LoginInfo* event with the category *SecurityPolicyServer* will, for example, be displayed as *SecurityPolicyServerLoginInfo* event.

Events	Description	I-DBC	WS-DBC	Hidden
Category: ADF (Access Decision Function)		✔	✔	
AdminRequestDeniedFailure	A request to administer a certain part of the configuration has been rejected.			
AdminRequestSuccess	A request to administer a certain part of the configuration succeeded.			
DNSyntaxFailure	Syntax of LDAP DN is invalid.			✔
RequestAcceptedSuccess	A request has been accepted.			
RequestDeniedFailure	A request has been rejected by the ADF, see page 332 for details.			
TimerFailure	An ADF request timed out, indicates overload situation.			✔
Category: ASManager (Access Session Manager)		✔		
ReplicationTerminateAS-RequestInfo	The replication service has requested termination of Access Session (AS).			
ReplicationTerminateIOR-RequestInfo	The replication service has requested termination of an IOR request.			
SessionCreationGrantedInfo	The creation of an access session has been granted.			
SessionTerminationGranted-Info	Access Session termination has been granted.			
Category: Authentication		✔	✔	
BasicAuthenticationFailure	The Proxy successfully authenticated a user with HTTP Basic Authentication (see page 333).			
BasicAuthenticationSuccess	The Proxy successfully authenticated a user with Basic Authentication.			
GSSUPAuthentication-Failure	The Proxy failed to authenticate a client with the given CSIv2/SAS authentication context.			
GSSUPAuthentication-Success	The Proxy successfully authenticated client with a CSIv2/SAS authentication context.			
HTTPBasicAuthentication-Failure	Unused, kept for downward compatibility.			✔
HTTPBasicAuthentication-Success	Unused, kept for downward compatibility.			✔
IPbasedAuthenticationFailure	Failed to authenticate user by ip address.			

Events	Description	I-DBC	WS-DBC	Hidden
IPbasedAuthenticationInfo	No mapping for IP to user id available.			
IPbasedAuthentication-Success	Successfully authenticated user by ip address.			
SAMLAssertionFailure	The Proxy failed to authenticate a user with a SAML assertion (see page 333).			
SAMLAssertionSuccess	The Proxy successfully authenticated a user with a SAML assertion.			
Category: Authorization		✔	✔	
ConstraintEvaluationFailure	Authorization failed because the evaluation of a constraint of the requesting entity evaluated to false.			
Category: ChildCommunication		✔	✔	
MessageReceptionFailure	Exception while reading object from child.			✔
MessagingThreadFailure	Internal error in messaging mechanism. Previous message has no events.			✔
ReadingFailure	Failure while trying to read input from child.			✔
TerminationSuccess	Child has terminated without error.			✔
Category: DBCAuthenticator		✔		
AuthenticationFailure	Authentication using the DBC authenticator interface has failed.			
AuthenticationInfo	Information about every step of the authentication process.			
AuthenticationSuccess	Authentication using the DBC authenticator interface has succeeded.			
Category: EventChannel		✔	✔	
MessageFailure	Event message does not comply with event specification.			✔
Category: LicenseManager		✔	✔	
LicenseExpiredFailure	The license of your DBC expired. You can get a new test license by contacting your customer support representative.			
LimitReachedInfo	License limit reached (for example, number of connections exceeded).			
Category: Persistence (stores and reloads IOR Table content)		✔		

Events	Description	I-DBC	WS-DBC	Hidden
ActivationInfo	Persistence activation state changed.			
FileCreateFailure	Persistence file creation failed.			
FileCreateSuccess	Persistence file created successfully.			
FileDeleteFailure	Persistence file deletion failed.			
FileDeleteSuccess	Persistence file deleted successfully.			
FileLoadFailure	Persistence file load failed.			
FileLoadSuccess	Persistence file loaded successfully.			
FileWriteFailure	Writing to persistence file failed.			
Category: PolicyRepository		✔	✔	
ChangeInfo	The policy has been changed by the Admin Console.			
IntegrityFailure	Integrity of policy repository has been compromised.			
MiscFailure	LDAP exception, file open exception or entry expired.			✔
RetrieveFailure	The policy information defining access rights for the requested target was not found in the repository (see also page 333).			
SyntaxFailure	Wrong format of entity identifier.			✔
ValidityFailure	The repository entry is not valid, i.e., current date is not between the entry's activation date and expiration date.			
Category: Proxy			✔	
HTTPConnectionAccepted-Success	An incoming HTTP connection has been accepted.			
HTTPConnectionClosed-Success	An HTTP connection has been successfully closed, the TCP socket is released.			
HTTPConnection-EstablishedSuccess	An outgoing HTTP connection has been established, a TCP socket is allocated.			
HTTPConnectionFailure	Unable to establish an outgoing or incoming HTTP connection.			
HTTPMessageReceivedInfo	HTTP message received.			
HTTPMessageSentInfo	HTTP message was successfully sent.			
InternalErrorFailure	An internal error occurred, please contact customer support.			

Events	Description	I-DBC	WS-DBC	Hidden
InvalidHTTPFailure	The HTTP request is syntactically or semantically wrong, see also page 335.			
MessageFilterEvaluation-Failure	A message was rejected because its parameters did not conform to a filter rule.			
MessageFilterEvaluation-Success	A message's parameters were evaluated successfully against a filter.			
MessageFilterFailure	Something went wrong when processing the filter, e.g., the filter was syntactically incorrect. You should not see this event when defining filters with the professional mode of the Admin Console.			
MessageProcessedInfo	Successfully processed a SOAP message.			
PolicyValidityFailure	Accessed entity of security policy not valid.			
ResourceMappingFailure	For this resource there is no entry in the resource table.			
ResourceMappingSuccess	A mapping was found in the resource table for the requested resource.			
SOAPAttachmentFilter-Failure	Filter does not allow SOAP attachment. The message carrying the attachment was discarded.			
SOAPAttachmentFilter-Success	Filter allowed message carrying the SOAP attachment to pass.			
WSDLRetrievalDenied-Failure	Denied retrieval of WSDL via HTTP GET.			
Category: ProxyEngine		✔		
GIOPBiDirContextInfo	A bidirectional IIOP service context has been created or received.			
GIOPConnection-AcceptedFailure	The Proxy has been unable to accept an incoming GIOP connection on a TCP/SSL acceptor. This may happen when the client opens a connection and immediately closes it again, e.g., due to a bug in the client application.			
GIOPConnection-AcceptedSuccess	An incoming GIOP connection has been accepted on a TCP/SSL acceptor by the Proxy.			

Events	Description	I-DBC	WS-DBC	Hidden
GIOPConnection-AssociationExpiredInfo	Incoming and outgoing GIOP connections are associated to each other. These associations are stored in an association table. If either the incoming or outgoing GIOP connection is closed the association will be deleted, this event will be sent, and the remaining connection will be closed as well (i.e., you will see a GIOPConnectionClosedSuccess event).			
GIOPConnectionClosed-Success	The GIOP connection has been successfully closed, incoming or outgoing TCP sockets are freed.			
GIOPConnection-EstablishedFailure	The Proxy tried but failed to establish an outgoing GIOP connection. Possible reasons are that no connector ports are available or I/O Errors like connection refused (no server listening on the requested port). The client will receive a CORBA TRANSIENT exception via the incoming GIOP connection.			
GIOPConnection-EstablishedSuccess	An outgoing GIOP connection has been successfully established by the Proxy, a TCP socket is allocated.			
GIOPConnectionFailure	An error was detected on a GIOP connection, the TCP socket is released.			
GIOPConnectionLimit-Failure	The limit for incoming GIOP connections has been reached (cf. “Connection Limiter” on page 153).			
GIOPMessageEnqueuedInfo	A GIOP message has been enqueued.			
GIOPMessageReceivedInfo	A GIOP message has been received. The event properties include the message type and destination.			
GIOPMessageSentFailure	A GIOP message could not be sent.			
GIOPMessageSentInfo	Information about the sent GIOP message (message type and destination).			

Events	Description	I-DBC	WS-DBC	Hidden
IORCheckFailure	This event is intended to support the integration process of the DBC with the application. This event occurs if data in the traffic stream looks like an IOR but does not suffice all constraints. Usually, this happens if a malformed object reference (IOR) is returned by an application server, passing the Proxy. Due to the fact that malformed IORs of an application server cause interoperability problems but are hard to detect, the threshold for this event is very low, and might be triggered misplacedly. After the integration process has been finished the specific event may be deactivated.			
IORDeproxifiedInfo	This event refers to reverse proxification.			
IORProxifiedInfo	This event indicates that the Proxy already knows the detected IOR.			
IORProxifiedNewInfo	This event indicates that the Proxy has detected and proxified a new original IOR.			
InternalFailure	An internal error occurred.			
ParameterCheckFailure	Parameter check failed.			
ParameterCheckSuccess	Parameter check succeeded.			
ProxyObjectAccessError	Failed to access a proxy object.			
ProxyObjectAccessInfo	A proxy object was accessed.			
Category: ProxyManager		✔	✔	
AliveInfo	Keep the TCP connection busy.			✔
ChildExitedInfo	The proxy (child) has exited normally when shutting down.			✔
ChildSignalledFailure	Child exited due to signal.			✔
FailedOverInfo	Proxy manager failed over to a different SPS.			
FileError	The ProxyManager failed to open a file.			
InternalFailure	Exception during thread run.			✔
ReadyToAcceptInfo	The proxy manager has started and configured all engines, which are now ready to process client connections.			
RequestLimitFailure	Number of requests in queue limit reached, request dropped.			✔
StartedInfo	The proxy manager has started.			

Events	Description	I-DBC	WS-DBC	Hidden
ThreadLimitInfo	Thread limit has been reached. No further threads can be created.			
ThreadLowWaterMark-Success	The system has recovered and that the number of threads has dropped below this mark.			✔
ThreadThresholdInfo	The number of threads has reached the warning level. This warning level is not yet critical.			✔
UnknownChildFailure	Child process could not be found.			✔
Category: ProxyProcess		✔	✔	
CreatedInfo	The proxy process has started.			✔
ProxyManagerCommunicationFailure	Decoding a message from the proxy manager has failed.			✔
Category: Replication		✔		
IORTableCopyFailure	IOR table copy mode has failed. This event should only be seen when the cluster is restarted.			
IORTableCopyStartedInfo	IOR table copy mode has been initiated.			
IORTableCopySuccess	IOR table copy mode has finished successfully.			
LocalModeInfo	The local mode changed.			
SetPeerStateInfo	Peer state has changed.			
Category: SSLAuthentication		✔	✔	
CertificateFailure	The Proxy failed to authenticate a user with an SSL certificate, see page 335.			
CertificateSuccess	The Proxy successfully authenticated a user with an SSL certificate.			
CertificateVerificationFailure	Failed to verify a certificate.			
OCSPFailure	Error while processing OCSP on-line validation.			
Category: SSLTransport		✔	✔	
ConfigurationInfo	The instantiation of an SSL Profile failed, this is not a problem as long as the profile is not used. If the profile is used you will get a MissingConfiguration Exception.			
HandshakeFailure	SSL detected error while in handshake mode, see page 336.			
InternalFailure	An SSL internal error occurred.			✔

Events	Description	I-DBC	WS-DBC	Hidden
NoPeerCertInfo	No Peer DN available, because the SSL configuration was not set to require client authentication.			
PeerRecognizedInfo	SSL client authentication has been successful (the peer DN has been recognized by the Proxy).			
Category: SecurityPolicyServer		✔	✔	
AccessFailure	A client failed to access the SPS.			
ClientDisconnectedInfo	Client disconnected from the SPS.			✔
ConfigDeploymentFailure	Deploying a config on a ProxyManager failed.			
ConfigChangeDetailInfo	This event provides detailed information about the changes that were applied to the configuration of the Security Policy Server.			
ConfigChangeInfo	The configuration of the SPS has been changed.			
ConfigReadInfo	A new configuration was written to the SPS.			
ConfigWrittenFailure	The writing of a new configuration to the Security Policy Server failed.			
ConfigWrittenSuccess	The writing of a new configuration to the Security Policy Server was successful.			
ConfigurationFailure	There was an error in the configuration.			✔
ConfigurationIntegrityFailure	Integrity of the configuration file (dbc.config) was compromised (e.g., when editing the configuration outside the Admin Console).			
LoginFailure	The login to the Security Policy Server failed.			
DBCConfigDeployment-Failure	Deploying a config on a ProxyManager failed.			
LoginInfo	Status about a client login attempt.			
LoginSuccess	The login to the SPS succeeded.			
MarkerInfo	This event can be triggered in the Admin Console, for example, to mark the start of a test run.			
PeerConnectionEstablished-Failure	Failed to establish a connection to the peer SPS.			
PeerConnectionEstablished-Success	Successfully established a connection to the peer SPS.			
PolicyChangeDetailInfo	Detailed information about a policy change.			
PolicyChangeInfo	The security policy changed.			

Events	Description	I-DBC	WS-DBC	Hidden
SkippedEventsInfo	The ring buffer holding the events was too short.			
StartedInfo	The Security Policy Server has started.			
SynchronizationFailure	The synchronization of SPSs in a cluster failed. Critical failure, verify network connectivity between SPS hosts. Check the status of each SPS.			
SynchronizationSuccess	Synchronization with SPS succeeded.			
Category: XML			✔	
DSigCreationFailure	Signature Creation failed.			
DSigCreationSuccess	Successfully signed message.			
DSigVerificationFailure	XML Digital Signature validation failed, see page 337.			
DSigVerificationSuccess	XML Digital Signature validation succeeded.			
DecryptionFailure	Failed to decrypt XML.			
DecryptionSuccess	Succeeded to decrypt XML.			
EncryptionFailure	Failed to encrypt XML.			
EncryptionSuccess	Succeeded to encrypt XML.			
InterfaceValidationFailure	Validation of SOAP message against the interface's definition failed.			
InterfaceValidationSuccess	Validation of SOAP message against the interface's definition succeeded.			
ParsingFailure	The XML parsing of the message failed. This can happen because the XML contained in the message is not well formed.			
SOAPStructureAnalysis-Failure	The Proxy received syntactically incorrect SOAP envelope.			
SchemaLoadingFailure	The Proxy failed to load an XML schema file. This might be due to an invalid or syntactically ill-formed schema. Also check if the schema file location is defined correctly (on the "Resources Incoming Policy" panel).			
SchemaLoadingSuccess	The Proxy successfully loaded an XML schema file.			
SchemaValidationFailure	XML schema validation failed, see page 337.			
SchemaValidationSuccess	XML schema validation succeeded.			

Events	Description	I-DBC	WS-DBC	Hidden
UnknownSecurityToken-Failure	Unknown WS Security token. If an unknown security token is regarded as critical enable this event. If not, you may enable the *XMLUnknownSecurity-TokenInfo* event.			
UnknownSecurityTokenInfo	Unknown WS Security token. If an unknown security token is not regarded as critical but you wish to be notified you may enable the this event. If it is regarded as a failure enable the *XMLUnknownSecurityTokenFailure*event.			
UnsignedSAMLFailure	Unsigned SAML Assertion in SOAP Message. If you require SAML assertions to be signed activate this event. If not, you may enable the *XMLUnsigned-SAMLInfo* event.			
UnsignedSAMLInfo	Unsigned SAML Assertion in SOAP Message. If you do not require SAML assertions to be signed but wish to be notified when a message carrying an unsigned SAML assertion passes (e.g. for testing purposes) activate this event. Otherwise enable the *XMLUnsignedSAMLFailure* event.			
ValidationFailure	XML validation failed.			
WSSEUsernameTokenInfo	A WS-Security user name token was identified. This event yields information about the user that was identified by this token.			

A.2 Important Audit Events

This section lists the most important events, explains the information contained in them, and gives possible reasons why it was sent.

Event	I-DBC	WS-DBC	Page
ADFRequestDeniedFailure	✔	✔	page 332
AuthenticationBasicAuthenticationFailure		✔	page 333
AuthenticationSAMLAssertionFailure		✔	page 333
PolicyRepositoryRetrieveFailure	✔	✔	page 333
ProxyResourceMappingFailure		✔	page 334
ProxyHTTPConnectionFailure		✔	page 334
ProxyInvalidHTTPFailure		✔	page 335
SSLAuthenticationCertificateFailure	✔	✔	page 335
SSLTransportHandshakeFailure	✔	✔	page 336
XMLSchemaValidationFailure		✔	page 337
XMLDSigVerificationFailure		✔	page 337

ADFRequestDeniedFailure

A request has been rejected by the access decision function (ADF). Possible reasons are:

- “Insufficient Authentication”: The client’s authentication level is insufficient: Check the value of the authentication level (“ALA”) in the event details. E.g., the client’s ALA is “UserId/Password” but the resource’s incoming policy has been configured to require SSL authentication. Adapt the client to use the required ALA or change the resource’s authentication policy with the Admin Console.
- “Insufficient Connection Protection”: The message protection level is insufficient: Check the value of the protection level (“PLA”) in the event details. E.g., the PLA is “NoProtection” but the resource requires SSL encryption for incoming mes-

sages. Adapt the client to use the required protection mechanism or change the resource's protection policy with the Admin Console.

- "Method access denied": The client has insufficient access rights: Access is denied following the rules in the resource's access control policy, see "User Properties – Privileges" on page 250 on how to assign the user access rights to a resource.
- "Target access denied": The client has insufficient rights to access the target, see "User Properties – Privileges" on page 250 on assigning access rights.

"No rule set in Policy Repository for ...": There is no policy for the requested resource. In some error conditions an additional *PolicyRepositoryRetrieveFailure* event is generated, please see next section for details.

AuthenticationBasicAuthenticationFailure

The DBC failed to authenticate user with Basic Authentication. This can either be HTTP Basic Authentication or UsernameToken authentication. In case of HTTP Basic Authentication this can have the following reasons:

- a syntax error in the basic authentication header of the client's HTTP message,
- an unknown authentication method in the basic authentication header of the client's HTTP message (the only supported method is "Basic"),
- the client provided a user name which is unknown to the DBC,
- the client provided the wrong password.

AuthenticationSAMLAssertionFailure

The Proxy failed to authenticate a user with a SAML assertion. The SAML assertion is structually invalid. This doesn't necessarily indicate a fatal failure, also see note above.

PolicyRepositoryRetrieveFailure

No policy information defining access rights for the requested target was found in the repository. A possible reason is that you did not define a resource in the policy for the targeted URL/IOR. When using the **WS-DBC** additional reasons may be:

- The URL does not match because there is a typo in the resource Id.
- An attempt to retrieve a resource's WSDL through the DBC when the resource does not allow WSDL GET Requests. In this case, you will see the target urn URN:XD_PSEUDO_WSDL_NS in the event details (for instructions on how to allow WSDL Get requests for a resource, please see "Allow WSDL GET Requests" on page 268).

- The resource mapping is incorrect. Start the Admin Console and go to the “Resource Mappings” pane. Check if the path and the resource part are correctly given in the table. Does the resource really exist on your application server?

If you believe the resource should exist, carefully check the characters shown in the failure message and make corrections to the target resource. In particular, be careful for an exact match of whitespace, IP address, and port numbers. Also note that the specification for a mapped resource has a slightly different syntax than the resource identifier itself; the resource identifier will accept an optional message XML namespace URN specifier. Finally, ensure that all changes have been committed to the Security Policy Server.

ProxyResourceMappingFailure

A client accessed the Proxy with a path that is not in the resource mapping table. Start the Admin Console and configure the resource mapping (for details please refer to section “Resource Mappings” on page 150). Also make sure that the client uses the correct URL.

ProxyHTTPConnectionFailure

The Proxy is unable to establish or accept an HTTP connection, due to one of the following reasons:

- “connection refused”, “no route to host”, “connection failed for sd=...”: The Proxy is unable to establish a connection to its communication partner (another DBC or the application server). Make sure that the communication partner is up and running.
- You configured a URL with protocol prefix `https` in the resource mapping table, but the application server offers only plain HTTP.

ProxyInvalidHTTPFailure

The HTTP request is syntactically incorrect. This can have one of the following reasons:

- "Invalid HTTP request line...": The Proxy received an invalid HTTP request (e.g., connecting with HTTPS on a plain TCP port).
- "Invalid Content Type": The Proxy received an HTTP request where the content type is neither empty, nor "text/xml" or "application/soap+xml".
- The HTTP request message is neither "GET nor "POST".
- The HTTP request is of type "GET" and contains a SOAP message body.
- An HTTP request was received on a connection initiated by the Proxy.
- An HTTP reply was received on a connection accepted by the Proxy.
- An HTTP GET request was received. HTTP GET requests are only allowed to retrieve WSDL documents. Other GET Requests are blocked by the Proxy.

SSLAuthenticationCertificateFailure

The client's SSL certificate is trusted and valid, but the Proxy failed to recognize it because the user is not known to the DBC. Start the Admin Console, go to the "Security Policy – Users" panel, and check that the user is included in the list (section "User Properties – General" on page 242 for details). Also check if the DN in the certificate equals the one given for this user (consult the "Certificate DN" property of this event). Make sure all DN components are listed in the same order as in the request.

Note that this does not necessarily indicate a fatal failure, The Proxy tries all configured authentication methods for a user until it finds the proper one for this client. For example: There are two authentication methods defined for a user in the Admin Console: SSL authentication and authentication by IP-address. The user contacts the Proxy authenticating by his IP-address. The Proxy will first try to authenticate the user with SSL. This will fail and an *SSLAuthenticationCertificateFailure* event will be sent. The Proxy will then try to authenticate the user by his IP-address. This will succeed.

SSLTransportHandshakeFailure

The Proxy detected an error while in SSL handshake mode. This can have one of the following reasons:

- "unknown protocol": the client tries to connect with plain IIOP to an SSL listener. In this case, the client should use SSL or a plain IIOP acceptor should be configured with the Admin Console (on the "External Interface" panel).
- "peer did not return a certificate": the client uses certificates that are not trusted by the Proxy. During SSL handshake the Proxy sends a list of DNs of trusted CA certificates to the client. The client sees that his certificate will not be trusted by the Proxy and doesn't provide a certificate. Please append the client's CA certificate to the Proxy's file of trusted CA certificates, see section "Making the DBC Proxy Trust External Certificates" on page 205 for details.
- "self signed certificate in chain": the client uses certificates that are not trusted by the Proxy. During SSL handshake the Proxy sends a list of DNs of trusted CA certificates to the client. Although the client has no valid certificate it returns a certificate chain. Please add the client's CA certificate to the "trusted CA Certificates" file, see section "Making the DBC Proxy Trust External Certificates" on page 205 for details.
- "sslv3 alert certificate unknown": the client does not trust the Proxy CA certificate. Please consult the manual of the client application on how to make the client trust the Proxy CA certificate.
- "alert bad certificate at client": the communication partners use incompatible SSL versions. If you think that this is the reason for the handshake failure, please contact customer support.
- "certificate expired": To determine the validity dates of generated certificates, you can use the script `printcert.sh` located on the SPS host in the directory `<INSTALLDIR>/sps/bin`, and in the `bin` directory on the proxy host. For example:

  ```
  printcert.sh ../adm/SPSCert.der
  ```

  The certificate will be printed in a readable form including the validity dates that look, for example, like this:

  ```
  Validity
  Not Before: Sep 16 09:41:56 2003 GMT
  Not After : Nov 22 09:41:56 2013 GMT
  ```

  If the "Not Before date" lies in the future, the certificate is not yet valid. In this case, correct the time settings on you computer and re-run the script that generates the keys and certificates (see section "Generating Keys" on page 90 for details).

XMLSchemaValidationFailure

The XML contained in the SOAP message did not validate against the installed XML schemas. This can occur because the message contains unknown data types, or differs structurally from what is specified in XML schemas. Try to run the client again but disable schema validation in the Proxy (with the Admin Console on the "Resource – Incoming Policy" panel), if the error persists, then the client application has provided invalid XML. The client application will need to be modified to produce valid XML.

XMLDSigVerificationFailure

XML Digital Signature validation failed. This event can have one of the following reasons:

- The signer is trusted but the signature is cryptographically invalid. This implies that the data has lost its integrity. If you believe this is the case, investigate whether someone has deployed a "man-in-the-middle" attack, modifying the content of secure messages. Contact your security administrator as is appropriate.
- The signer is untrusted. Check if the CA certificate of the signer is known to the Proxy, see section "Making the DBC Proxy Trust External Certificates" on page 205 for details.

APPENDIX

B Error Messages and System Exceptions

*The first section of this chapter lists error messages that might occur in the log files of the DBC Proxy and the Security Policy Server. When operating a **WS-DBC** clients may receive HTTP error messages (“HTTP Error Messages” on page 353) or SOAP error messages (“SOAP Error Messages” on page 353). When operating an **I-DBC** clients may receive CORBA system exceptions which may be raised or passed on by the Proxy (“CORBA System Exceptions and Minor Codes” on page 344). In most cases the DBC Proxy will also generate an audit event. Audit Events are explained in appendix “Audit Events” on page 321. Please refer to that appendix for a more detailed description of possible causes.*

B.1 DBC and SPS Error Messages

The following sections explain the format of I-DBC error messages as they appear in the log files (`proxy.log` and `sps.log` located in the `adm` directory of the Proxy and the SPS) and list some common error messages, explaining the cause and how to fix the errors.

B.1.1 Error Message Format

Some common error messages are explained below. A log file message usually looks like this example:

```
2001-08-01 14:05:03.365185:Level 2:<BackEnd>[8806::00000402] ↵
     SecurityServer.cc:262 void Watchdog::run() none -   ↵
     Watchdog caught exception IOException(1):           ↵
     Address lookup failed for host 'host.domain':       ↵
     Unknown host:Operation not permitted (Called from: )
```

A log file entry consists of the following fields:

- the date in sortable form, i.e., starting with the year as the most significant information, with microsecond precision
- the debugging level (Level 2)
- the module identifier (Backend)
- the process id (8806)
- the thread identifier (00000402)
- the source file name (SecurityServer.cc)
- the line number (262)
- the function name (Watchdog::run())
- a detailed error description

For you as the user, only the detailed description contains any relevant information. Technical support though will request the whole message when diagnosing a problem.

B.1.2 Common Error Messages

The following paragraphs will only refer to the error description (last bullet of the list above) when describing common errors due to misconfiguration. The table below lists the error messages and where explanations can be found.

Error Message	**Page**
License not found	341
Address lookup failed	341
Error while reading key file	341

Address already in use	342
Server Socket bind failed	342
System Exception “No Permission”	342
Cannot set SSL private key	343
Invalid key file format	343
Cannot open SSL certificate file	343
Unable to get local issuer certificate	344
Client does not accept Server’s certificate	344

License not found

```
Error loading license <filename>
```

Copy the license file (`license.txt`) you received with your software package or via email on the Security Policy Server host into the `adm` directory of the SPS and the Proxy. You can obtain a license from your reseller or directly from Xtradyne (mail to `info@xtradyne.com`).

Address lookup failed

```
Address lookup failed for host ‘host.domain.example’...
```

This can happen if the machine or its fully qualified name is not known to DNS. In that case use the literal IP addresses or fix DNS. Use the Admin Console to assign proper host names or IP addresses to the DBC interface. You can do this on the “Network Interfaces” panel of the Security Policy Server and the DBC Proxy, see “DBC Proxy Network Interfaces” on page 136 and “Management Interface” on page 149. Don’t forget to restart the DBC Proxy.

Error while reading key file

```
error while reading key file 'SPSKey.der': No such file or
directory
```

You forgot to generate keys and certificates. Refer to section “Generating Keys” on page 90 on how to create a default set of keys and certificates.

Another reason for this messages might be that the Security Policy Server has been started from the wrong directory. This can not happen when installing the DBC with the

provided installer, but it will happen if you move the files to a different directory after installation.

Address already in use

```
System exception `COMM_FAILURE'
Reason: Address already in use
Completed: no
Minor code: 1330577416 (bind() failed)
```

This exception occurs when a Security Policy Server is already running or another service uses the same port you configured for the Security Policy Server interface. You can check for listeners on Linux with:

```
netstat -tnlp | grep 15000
```

Replace the default 15000 with the port number you chose for the control connection. The output will read:

```
tcp 0 0 0.0.0.0:15000 0.0.0.0:* LISTEN 9282/sps
```

If the process name is SecurityServer (as shown above), all should be well, i.e., a Security-Server is already running. In this case, check why another Security Policy Server was started. If the process name denotes a different process, you must decide whether to stop and disable this process in the future or set the control connection listener to use a different port.

Server Socket bind failed

```
server socket bind failed for 192.168.1.5:8884: Cannot
assign requested address
```

This message means that you have chosen a host name or IP address for a Proxy which could be resolved but for which there is no interface on the host where the DBC Proxy runs. Check the settings you made in the Admin Console for the Proxy. Restart the DBC Proxy after you corrected the settings.

System Exception "No Permission"

```
CORBA_SystemException caught: System exception
`NO_PERMISSION'
```

There are two possible reasons for this error message:

- If you set the control connection properties for the DBC Proxy to *not* use SSL but for the Security Policy Server to use SSL, the message shown above will appear.

No control connection will be established. You need to rerun `proxyconfig.sh` located in the `bin` directory of the Proxy to set the correct value.

- The DBC Proxy will reject the Security Server's connection attempt if the CA certificate available to the DBC Proxy is different from the CA used by the Security Policy Server. Transfer the keys you generated with the `generatekeys.sh` script on the Security Policy Server to the DBC Proxy host (the file is called `ProxyKeys.tar`). Unpack the tar archive into the `adm` directory of the Proxy.

Cannot set SSL private key

```
Cannot set SSL private key
SSL.publicSSL.crypto.RSA.keyFiles.privateKey.key
```

There was a problem with the private key you set in the SSL configuration for at least one of the SSL Acceptors. One of the following reasons might apply:

- The opaque key you entered does not match the certificate file. This happens if you simply copy a configuration from another DBC configuration. Certificates are always read from files but the private keys are stored in the configuration. Therefore the certificate referred to by the filename stored in the configuration might be a different certificate. Replace the opaque key in the dialog box with the key matching the certificate.
- The opaque key you entered is invalid, e.g., because it is incomplete or was modified from the original while entering it into the dialog box. Make sure the key is valid.

Invalid key file format

```
SSL.privateSSL.crypto.RSA.keyFiles.privateKey.format:
Invalid key file format DER
```

Only PEM encoded keys are allowed. You can convert keys from DER to PEM with the script `<INSTALLDIR>/sps/bin/der2pem.sh`.

Cannot open SSL certificate file

```
SSL.publicSSL.crypto.RSA.keyFiles.certificate.filename:
Cannot open SSL certificate file ProxyChain.pem

XDN:Failure:ProxyManagerInternalFailure:exception=
Q218SSLContextProvider21FailingBuildException(0):
SSL.publicSSL.crypto.RSA.keyFiles.certificate.filename:
Cannot open SSL certificate file ProxyChain.pem
```

The file `ProxyChain.pem` was not found on the DBC Proxy host. Make sure the file exists in the `adm` directory of the Proxy and that it is readable by the user `xtradyne`.

If you store the key directly in the configuration file, the configuration depends on files stored on the DBC Proxy if the certificate is not included in the configuration.

Unable to get local issuer certificate

```
XDN:Failure:SSLTransportCertificateFailure:Message=unable
to get local issuer certificate:error=20:
```

This error message indicates that the issuer of a client certificate cannot be found in the database (by default the file `TrustedCAs.pem`). Add the client's CA certificate (in PEM format) to the database (for details, please refer to section "Making the DBC Proxy Trust External Certificates" on page 205).

Client does not accept Server's certificate

```
XDN:Failure:SSLTransportHandshakeFailure:"sslv3 alert bad
certificate"
```

This error message indicates that the client does not accept the server's certificate. There are two possible reasons:

- On the connection from the client to the DBC Proxy: the client does not trust the DBC Proxy's certificate. You must configure your client to accept the DBC Proxy's certificate (`ProxyChain.pem`) or the DBC Proxy's CA certificate (`ProxyCACert.pem`). For details, please refer to "Integrating the DBC with Applications" on page 206.
- On the connection from the DBC Proxy to the Server: the server provides a list of trusted CAs that it is willing to accept, but the DBC Proxy's certificate was not issued by one of those listed CA's. Configure your server to also trust the I-DBC Proxy's certificate (`ProxyChain.pem`) or the I-DBC's CA certificate (`ProxyCACert.pem`). For details, please refer to "Integrating the DBC with Applications" on page 206.

B.2 *CORBA System Exceptions and Minor Codes*

CORBA System Exceptions apply only to the **I-DBC**.

CORBA System Exceptions are used to indicate errors which may occur during the invocation of an operation on a CORBA Object. Typically, CORBA System Exceptions

are raised by the middleware, i.e., the client-side ORB or the server-side ORB. To give further details on the cause and origin of an error a CORBA System Exception includes a minor exception code. This is a 32-bit value which contains a 20-bit Vendor Minor Codeset Id, VMCID (high order bits) and a 12-bit error code (low order bits).

To diagnose the cause of exceptions it is important to know whether the exception was raised in the I-DBC or raised by the server (in this case, passed on to the client by the I-DBC). Exceptions raised by the I-DBC carry the VMCID “0x58540”.

B.2.1 BAD_OPERATION

A BAD_OPERATION system exception is raised by the I-DBC if the caller invokes an unknown operation on a DBC object, e.g., DBC Authenticator or DBC Name Service.

Minor code (hexadecimal/decimal value)	**Description**
NoSuchProxyOperation *0x58540001* *1481900033*	Requested operation is not known by the DBC service object.

B.2.2 BAD_PARAM

A BAD_PARAM system exception is raised by the I-DBC if the caller passes a illegal value to an operation invocation on a DBC object, e.g., DBC Authenticator or DBC Name Service.

Minor code (hexadecimal/decimal value)	**Description**
NamingServiceNilObject *0x58540001* *1481900033*	A nil object reference has been passed to a bind() or bind_context() operation call on the DBC Naming Service.
NamingServiceIllegalValue *0x58540002* *1481900034*	An illegal value has been passed to an operation call on the DBC Naming Service, i.e., an empty name has been provided.

B.2.3 COMM_FAILURE

A COMM_FAILURE system exception is raised by the I-DBC if a communication failure occurs while an operation request is in progress. This can happen if you have an access session-restricted license and the maximum number of access sessions has already been created.

Minor code (hexadecimal/decimal value)	**Description**
UnspecificFailure *0x58540000* *1481900032*	Unable to connect to the peer ORB or connections is lost while an operation request is in progress.
ConnectionTimedOut *0x58540001* *1481900033*	The connection request to the peer ORB has timed out. This may happen if the peer does not accept the connection (process hangs or is overloaded).
ConnectionRefused *0x58540002* *1481900034*	The connection request to the peer ORB has been refused. This may happen if the peer has no listener at the requested endpoint.
NoRouteToHost *0x58540003* *1481900035*	Unable to connect to the peer ORB as there is no route to the peer host. This may happen if the peer is down or routing information is not configured properly on the DBC host (or on a host in the transit network).
ConnectionClosed *0x58540004* *1481900036*	The client connection has been closed while there are pending requests and the client is waiting for reply. This may happen if the DBC is shutdown or the server connection was closed during message processing.

Minor code (hexadecimal/decimal value)	Description
ConnectionShutdown *0x58540005* *1481900037*	The client connection has been forcibly closed while there are pending requests and the client is waiting for reply. This may happen if the DBC is shutdown or the server connection was closed during message processing.

B.2.4 INITIALIZE

A INITIALIZE system exception is raised by the I-DBC if the caller has invoked a DBC service function, which could not be initialized.

Minor code (hexadecimal/decimal value)	Description
NamingServiceInit *0x58540001* *1481900033*	The DBC Proxy Name Service has not been initialized.

B.2.5 INTF_REPOS

A INTF_REPOS system exception is raised by the I-DBC if the caller called the _interface() pseudo operation to obtain the reference to Interface Repository for a DBC object.

Minor code (hexadecimal/decimal value)	Description
IfRepositoryNotAvailable *0x58540001* *1481900033*	No interface repository available for DBC object service (DBC Authenticator, DBC Name Service).

B.2.6 IMP_LIMIT

A IMP_LIMIT system exception is raised if message processing has exceeded a hard or soft implementation limit.

Minor code (hexadecimal/decimal value)	**Description**
ReceivedMesssageTooLarge *0x58540001* *1481900033*	The configured maximum message size has been exceeded.

B.2.7 INTERNAL

An INTERNAL system exception is raised if the I-DBC Proxy has run into an error state, which does not allow for graceful recovery of the message processing.

Minor code (hexadecimal/decimal value)	**Description**
UnexpectedException *0x58540001* *14819000333*	Unexpected exception caught during message processing.
PolicyDataTimeOut *0x58540002* *14819000334*	Retrieval of policy data from Security Policy Server timed out. No response from Security Policy Server.

These exceptions may have the following reasons:

- An “UnexpectedException” error may occur if the I-DBC is operated with a bad configuration. The event log will show the *ProxyEngineInternalFailure* audit event which contains the reason of the failure. Please contact Customer support in this case.
- A “PolicyDataTimeOut” error may occur due to the following reasons:
  - The SPS host is overloaded. Check the system load on the SPS host.
  - The SPS hangs or is not available. Restart the SPS.
  - The I-DBC Proxy host is overloaded. Check the system load on the SPS host.

- Connectivity between the I-DBC Proxy host and SPS host is degraded. Contact your network administrator.
- LDAP Server does not provide a timely response. Contact you directory administrator.

B.2.8 NO_PERMISSION

A NO_PERMISSION system exception is raised by the I-DBC if caller of an operation has insufficient privileges.

Minor code (hexadecimal/decimal value)	**Description**
AccessDenied *0x58540001* *1481900033*	Access to the requested resources is denied.
ParameterCheckFailed *0x58540002* *14819000334*	Content check rule for invocation parameters failed.

The “Access Denied” exception can have the following reasons:

- The client could connect to the I-DBC, but the I-DBC denied access according to policy. The event log will show the *ADFRequestDeniedFailure* audit event which contains the principal name (userID) and the requested target and operation. Modify your access control policy to allow that user access to the target object. For further details, please refer to “PolicyRepositoryRetrieveFailure” on page 333.
- The SSL Handshake failed (exception not raised by the I-DBC). The event log will show the *SSLTransportHandshakeFailure* audit event which contains the reason for the handshake failure (for example, the peer’s CA certificate is not trusted). For further details, please refer to section “SSLTransportHandshakeFailure” on page 336.

A *ParameterCheckFailed* error may occur if the content check rule for parameters failed. The event log will show the *ProxyEngineParameterCheckFailure* which contains the reason for the failure, i.e., it shows which filter term has failed and the value of the actual invocation parameter.

B.2.9 MARSHAL

A MARSHAL system exception is raised if the I-DBC has received a malformed GIOP message from the network.

Minor code (hexadecimal/decimal value)	**Description**
WrongGIOPVersion *0x58540001* *1481900033*	The I-DBC Proxy received a GIOP message which contains an illegal version number in the GIOP Header. Legal values are GIOP 1.0, 1.1, 1.2, and 1.3.
InvalidMessageTypeHeader *0x58540002* *1481900034*	The I-DBC Proxy received a GIOP message which contains an malformed message type header.
InvalidMessageBody *0x58540003* *1481900035*	The I-DBC Proxy received a GIOP message which contains an malformed message body.
InvalidAddressing-Disposition *0x58540004* *1481900036*	The I-DBC Proxy received a GIOP 1.2 or 1.3 Request with an invalid addressing disposition. Valid values are KeyAddr(0), ProfileAddr(1), ReferenceAddr(2).
BadTargetProfileAddr *0x58540005* *1481900037*	The I-DBC Proxy received a GIOP 1.2 or 1.3 Request with an invalid Tagged Profile (profile address) for the targeted proxy object. This minor code may also indicate that the enclosed Tagged Profile is not an IIOP Profile with tag id TAG_INTERNET_IOP(0).
BadTargetReferenceAddr *0x58540006* *1481900038*	The I-DBC Proxy received a GIOP 1.2 or 1.3 Request with an invalid IOR (reference address) for the targeted proxy object.

B.2.10 NO_RESOURCES

A NO_RESOURCE system exception is raised by the I-DBC if specific application or system resource is not available for message processing.

Minor code (hexadecimal/decimal value)	**Description**
NoConnector *0x58540001* *1481900033*	No connector available to create transport connection to the requested target. Check the configuration for I-DBC external and internal interfaces.
NoSessionForContinueAuth *0x58540002* *1481900034*	The RSA/ACE Server requested “continue authentication”, but the associated session data is no longer available. This may happen if an authentication process lasts for a long time period.

B.2.11 NO_IMPLEMENT

A NO_RESOURCE system exception is raised by the I-DBC if the requested service or operation is not implemented by the targeted CORBA object on the I-DBC.

Minor code (hexadecimal/decimal value)	**Description**
OperationNotImplented *0x58540001* *1481900033*	The requested operation is not implemented by the targeted DBC service object (DBC Authenticator, DBC Name Service).
PseudoOpNotImplemented *0x58540002* *1481900034*	The requested pseudo operation is not implemented by the targeted DBC service object (DBC Authenticator, DBC Name Service).

B.2.12 OBJECT_NOT_EXIST

A OBJECT_NOT_EXIST system exception is raised by the I-DBC if the requested proxy object is not found.

Minor code (hexadecimal/decimal value)	**Description**
NoSuchProxyObject *0x58540001* *1481900033*	Requested proxy object is not found.
VoidNamingServiceObject *0x58540002* *1481900034*	Operation was invoked on destroyed NamingContext or BindingIterator object. Typically, this happens if an operation on a context object is invoked after destroy() has been invoked by the client.

Commonly created in the I-DBC Proxy: the requested IOR is not available in the I-DBC Proxy. Usually, this just means that the initial IOR is not configured or you just forgot to save your new initial IOR table to the Security Policy Server. For details, please refer to section “IOR Proxification” on page 154.

The server does not know the object referenced by the IOR. If you edited initial IORs by hand, there may be a typo in the IOR (check the values with the Admin Console on the “Initial IOR Table” panel). Another reason might be that the IOR is a transient one and the server was restarted.

B.2.13 TRANSIENT

CORBA TRANSIENT exceptions are not raised by the I-DBC, only passed on. This exception should state whether it was created locally at client side (the client could not contact the I-DBC Proxy) or not (the I-DBC Proxy could not contact the Security Policy Server). Check the availability of the I-DBC Proxy or Security Policy Server respectively with the command:

```
telnet <I-DBCProxyHost> <I-DBCProxyPort>
```

The default ports of the I-DBC Proxy are 8884 for IIOP and 8885 for IIOP/SSL, the default port of the Security Policy Server is 15000. This command should yield “Connected to <I-DBCProxyHost>”. Possible errors are “No route to host”, “connect to

address <I-DBCProxyHost>: Connection refused" or just hang (probably routing is not correctly configured or a firewall is blocking access).

B.3 HTTP Error Messages

HTTP Error messages apply only to the **WS-DBC**.

The following table lists HTTP error messages. Note that HTTP error messages are generally only returned when the message does not contain SOAP. For a list of SOAP Faults, please see page 353.

Error Type	Description
400 Bad Request	A malformed HTTP message has been received from the client. Also generates a *ProxyInvalidHTTPFailure* event (see also page 335).
401 Unauthorized	User unknown or password invalid. Also generates an *AuthenticationHTTPBasicAuthenticationFailure* event (see also page 333).
403 Forbidden	Request has been rejected by the Access Decision Function (ADF). Also generates an *ADFRequestDeniedFailure* event (see page 333).
502 Bad Gateway	A malformed HTTP message has been received from the target resource. Also generates a *ProxyInvalidHTTPFailure* event (see also page 335).

B.4 SOAP Error Messages

SOAP error messages apply only to the **WS-DBC**.

The following table lists the messages that will be contained in the SOAP Faults returned to clients. These are sent as HTTP 500 Internal Server Error messages. The

WS-DBC Proxy generates generic SOAP Error messages. The SOAP client is only provided with minimal information. Detailed error reasons can be found in the event log.

Error Type	**Description**
"Internal Server Error. Please contact the server administrator. Reference message id: <id>"	An arbitrary error occured.
"Access denied."	Access denied according to policy.

APPENDIX

C SSL Ciphers

When defining SSL profiles for the DBC Proxy you can specify a ciphersuite. A predefined ciphersuite definition can be chosen or an arbitrary string label can be entered in openssl-style. The first section of this appendix gives a detailed description of this format. Section two lists the ciphers that we propose to use with the DBC.

C.1 Cipher Suite String Format

The cipher list consists of one or more cipher strings separated by colons. Commas or spaces are also acceptable separators but colons are normally used. The actual cipher string can take several different forms. It can consist of a single cipher suite such as RC4-SHA. It can represent a list of cipher suites containing a certain algorithm, or cipher suites of a certain type. For example SHA1 represents all ciphers suites using the digest algorithm SHA1 and SSLv3 represents all SSL v3 algorithms.

Lists of cipher suites can be combined in a single cipher string using the + character. This is used as a logical and operation. For example SHA1+DES represents all cipher

suites containing the SHA1 and the DES algorithms. Each cipher string can be optionally preceded by the characters !, - or +:

- If **!** is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.
- If **-** is used then the ciphers are deleted from the list, but some or all of the ciphers can be added again by later options.
- If **+** is used then the ciphers are moved to the end of the list. This option does not add any new ciphers it just moves matching existing ones.
- If none of these characters is present then the string is just interpreted as a list of ciphers to be appended to the current preference list.
- If the list includes any ciphers already present they will be ignored: that is they will not moved to the end of the list.
- Additionally the cipher string **@STRENGTH** can be used at any point to sort the current cipher list in order of encryption algorithm key length.

Cipher Strings

The following table lists all permitted cipher strings and their meanings. To obtain the cipher suite denoted by a cipher string use the following command:

```
openssl ciphers -v '<CIPHERSTRING>'
```

Table 1.

Cipher String	Meaning
DEFAULT	The default cipher list. This is determined at compile time and is normally ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH. This must be the first cipher string specified.
COMPLEMENTOFDEFAULT	The ciphers included in ALL, but not enabled by default. Currently this is ADH. Note that this rule does not cover eNULL, which is not included by ALL (use COMPLEMENTOFALL if necessary).
ALL	All ciphers suites except the eNULL ciphers which must be explicitly enabled.
COMPLEMENTOFALL	The cipher suites not enabled by ALL, currently being eNULL.
HIGH	‘‘high’’ encryption cipher suites. This currently means those with key lengths larger than 128 bits.

Table 1.

Cipher String	Meaning
MEDIUM	‘‘medium’’ encryption cipher suites, currently those using 128 bit encryption.
LOW	“low” encryption cipher suites, currently those using 64 or 56 bit encryption algorithms but excluding export cipher suites.
EXP, EXPORT	Export encryption algorithms. Including 40 and 56 bits algorithms.
EXPORT40	40 bit export encryption algorithms
EXPORT56	56 bit export encryption algorithms.
eNULL, NULL	The ‘‘NULL’’ ciphers offer no encryption at all. Because of this they are a security risk and they are disabled unless explicitly included.
aNULL	These cipher suites offer no authentication and thus are vulnerable to a ‘‘man in the middle’’ attack. Their use is discouraged. Currently these are the anonymous DH algorithms.
kRSA, RSA:	Cipher suites using RSA key exchange.
kEDH	Cipher suites using ephemeral DH key agreement.
kDHr, kDHd	Cipher suites using DH key agreement and DH certificates signed by CAs with RSA and DSS keys respectively. Not implemented.
aRSA	`Cipher suites using RSA authentication, i.e., the certificates carry RSA keys.`
aDSS, DSS	Cipher suites using DSS authentication, i.e., the certificates carry DSS keys.
aDH	Cipher suites effectively using DH authentication, i.e. the certificates carry DH keys. Not implemented.
kFZA, aFZA, eFZA, FZA	Ciphers suites using FORTEZZA key exchange, authentication, encryption or all FORTEZZA algorithms. Not implemented.
TLSv1, SSLV3, SSLV2	Protocol versions TLS v1.0, SSL v3.0 or SSL v2.0 cipher suites respectively.
DH	Cipher suites using DH, including anonymous DH.
ADH	Anonymous DH cipher suites.
3DES	Cipher suites using triple DES.
DES	Cipher suites using DES (not triple DES).
RC4	Cipher suites using RC4.

Table 1.

Cipher String	Meaning
RC2	Cipher suites using RC2.
IDEA	Cipher suites using IDEA.
MD5	Cipher suites using MD5.
SHA1, SHA	Cipher suites using SHA1.

C.2 Cipher Suites Offered by the DBC

As part of the SSL configuration in the Admin Console the set of allowed cipher suites is defined by a cipher string. For user convenience a drop-down menu is provided which contains the commonly used cipher string settings used with the product. To list the cipher suites denoted by these cipher strings run the script `showciphers.sh` which is included in the DBC distribution.

Index

A
Acceptors 143
  SSL Options 145
  TCP Options 144
Access Control 44, 45, 235
  Access Control Policy 235
  Access rights for resources 291
  Accessors 291
Access Sessions
  Access Session Management 129, 134
account name default 87, 93
Activation Date 252
Actors 259
Adding and Deleting
  DBC Proxies 135
  Security Policy Servers 185
Address Translation 159
  Mappings for Outgoing Connections 161
ADF
  AdminRequestDeniedFailure 322
  DNSyntaxFailure 322
  RequestAcceptedSuccess 322
  RequestDeniedFailure 322, 332
  TimerFailure 322
Admin Console 103
  Audit Policy 189
  Authentication Mechanisms 242
  Connecting to the SPS 111
  DBC Proxy Cluster 123
  Event Browser 111
  Expert Mode 197
  First Start 109
  General Navigation 115
  General Organization 117
  Groups 254
  Installation Overview 101
  Installed Keys and Certificates 102
  Installing the Software 99
  Management Interface 149
  Management Network Interface (SPS) 186
  Menu Overview 116
  Network Interfaces 136
  Preferences 110
  Replacing Keys and Certificates 211
  Replication Interface 149
  Resource Mappings 150
  Resources 263
  Roles 258
  SSL Keys 101
  SSL Profiles 165
  Tool bar icons 115
  Users 241
  Working Offline 110
Administration Concepts 104
Administration Connection 63, 207
Administrative rights 261
Application connections 204
Applications 295
  Administration 296
  Example 298
ASManager
  ReplicationTerminateASRequestInfo 322
  SessionCreationGrantedInfo 322
  SessionTerminationGrantedInfo 322
Audit Event Browser 119
Audit Policy 189, 190
  Audit Event Categories 192
  Event Consumer 193
  External Command 193
  Syslog 193
Authentication
  BasicAuthenticationFailure 322
  BasicAuthenticationSuccess 322
  GSSUPAuthenticationFailureFailure 322
  GSSUPAuthenticationSuccessSuccess 322
  HTTPBasicAuthentication-Failure 322
  HTTPBasicAuthenticationFailure 333
  HTTPBasicAuthenticationSuccess 322
  IPbasedAuthentication-Failure 322
  IPbasedAuthenticationInfo 323
  IPbasedAuthenticationSuccess 323
  SAMLAssertionFailure 323, 333
  SAMLAssertionSuccess 323
Authentication Mechanisms
  HTTP Basic Authentication 244
  IP Address 245
  RSA SecurID 184
  SAML assertion 243
  User ID/Password 244
  X.509 Certificate authentication 243

Authentication Policy 242
Authentication Token 279
Authentication, Required 269

C

CA Certificates (Trusted CAs) 171, 214
Certificate 202
  Certificate and Private Key 212
  Certificate Authority (CA) 171, 177
  Certificates used by the DBC 202
  Changing the encoding format 216
Certificate, and Private Key 170
ChildCommunication
  MessageReceptionFailure 323
  MessagingThread-Failure 323
  ReadingFailure 323
  TerminationSuccess 323
Ciphers
  Cipher Suite String Format 355
  Ciphers offered by the DBC 358
Communication Links 63
Concepts and Components 23
Configuration
  Configuration Data 105
  configuration file 80, 82
  DBC 103
  LDAP Server 239
  Security Policy 238
  XML Schema Validation 271
Conflicts Resolution Strategy 112
Connectors 146
Content Inspection 44
Control Connection 63, 207
  Using host names or IP addresses 187
Converting DER to PEM encoding 216
CORBA Name Service 152, 163
CSIv2 126, 243

D

DBC Proxy
  Configuration 103
  Initial Configuration 96
  Installation Overview 80
  Installing Keys and Certificates 95
  Management Interface 96
  NAT Addresses for Interfaces 138
  Network Interfaces 136
  Proxy Cluster 123
  Resource Mappings 150
  Startup and Shutdown 83
DBC Proxy Cluster 131
DBCAuthenticator
  AuthenticationFailure 323
  AuthenticationInfo 323
  AuthenticationSuccess 323
Deinstalling
  the DBC 97
  the Security Policy Server 97
Deployment Scenarios
  Screened Host Firewall with WS-DBC 65
  WS-DBC in the DMZ 66
der2pem.sh, conversion script 216
Dictionary Explorer 197

E

Enable SOAP Attachments 273
Encrypt Messages 279
Error Messages 339
  Address already in use 342
  Address lookup failed 341
  Cannot open certificate file 343
  Cannot set SSL private key 343
  Error Message Format 340
  Error while reading key file 341
  Failed to retrieve policies 221
  Invalid key file format 343
  Server Socket bind failed 342
  System Exception ‘No Permission´ 342
EventChannel
  MessageFailure 323
Events
  Alphabetical List 321
  Configuration 189, 195, 196
  Consumer 193
  Event Browser 111
  Event Flow 190
  Priorities 194
Expert Mode 197
  Copy, Move and Delete Entries 199
  Dictionaries 197
  Editing Entries 198
  Insert New Entries 199
Expiration Date 252
Exporting an IOR 154
External Interface 141
  Example 141
  Overview 140
External Key 204

F
Filters
  Predicates 286
  Templates 288
Firewall Configuration 227
Flow Control 153, 164

G
Generating Keys 90
GIOP Connection Timeout 130
Groups 254
  General 256
  Members 256
  Privileges 257

H
High Availability 37
High Load, single WS-DBC Proxy 37
HTTP
  400 Bad Request 353
  401 Unauthorized 353
  403 Forbidden 353
  502 Bad Gateway 353
  Acceptors 143
  Basic Authentication 244
  Connectors 146
HTTP Authorization Header
  Preserve 280
HTTP Basic Authentication 270
HTTPS
  Acceptors 143
  Connectors 146

I
IDL Import Wizard 264
IIOP Acceptor Details 144
  TCP Options 168
IIOP/SSL Acceptor Details 145
Include SAML Assertion in Signature 280
Incoming Policy
  Validate SOAP Request/Response 271
Incoming Responses 271
Initial Configuration
  DBC Proxy 96
  Security Policy Server 89
Initial IORs 154
Installation
  Prerequisites 61
Installation Steps 86
Installation target directory 73
Installing Keys and Certificates 201, 210
  External and Internal Keys 211
  for the Admin Console 101
  on the Security Policy Server 90
Installing the Admin Console 99
Installing the DBC Proxy 93
  Initial Configuration 96
  Installation Overview 80
  Keys and Certificates 95
  Overview of keys and certificates 95
Installing the Security Policy Server 74, 87
  Generating Keys 90
  Initial Configuration 89
  Installation Overview 78
  Overview of Keys and Certificates 91
  Postinstallation Steps 88
Internal Interface 140, 148
Internal Key 204
IOR
  Editing an 155
IOR Export 154
  Address Mappings 161
  Address Translation 159
  Advanced Features 157
  Importing IORs from a file 156
  Initial IORs 154
  proxified object key 157
  proxified Type id 157
IOR Proxification 154
  Proxification Options 158
iPlanet
  importing ldif files 239

K
Keys and Certificates 201
  Changing keys and certificates 211
  Changing the encoding format 216
  External Key 204
  Installing 201
  Internal Key 204
  on the Control and Administration
  Connection 207
  on the DBC Proxy 95
  on the Management Host 102
  on the Security Policy Server 91

L
LDAP

DBC Base DN 239
LDAP Server 238
Configuration 240
Prerequisites 239
License, Installing 88
LicenseManager
LicenseExpiredFailure 323
LimitReachedInfo 323
Listener Details, IIOP 168
Load Balancing 38
Log files 217
Backup 108

M

Management Network Interface (Proxy) 149
Management Network Interface (SPS) 186
Message Authentication 43
Message Protection 46
Message-oriented Security 29
Monitoring the DBC 152, 163
Mounting the CD 71

N

NAT
between the I-DBC Proxy and the SPS 187
Host Names vs. IP Addresses 187
Port Mappings for the I-DBC Proxy Cluster 145

O

Operations
Operation parameters 284
Outgoing Connections to Servers 160
Outgoing Policy 277
Inject SAML Assertion 279
Sign Message 277

P

PAM 127
Permissions for Roles 260
Persistence
ActivationInfo 324
FileCreateFailure 324
FileCreateSuccess 324
FileDeleteFailure 324
FileDeleteSuccess 324
FileLoadFailure 324
FileLoadSuccess 324
FileWriteFailure 324
Pluggable Authentication Modules 127
Policy Decision Point 34
Policy Enforcement Point 34
PolicyRepository
ChangeInfo 324
IntegrityFailure 324
MiscFailure 324
RetrieveFailure 324, 333
SyntaxFailure 324
ValidityFailure 324
Preparing Schema Files 272
Prerequisites
Admin Console 62
Domain Boundary Controller 62
Hardware and Software 61
Security Policy Server 62
Preserve HTTP Authorization Header 280
Preserve UsernameToken Authorization Header 280
Private Key and Certificate 170, 212
Privileges
for Groups 257
for Users 250
Process HTTP Basic Authentication 270
Process Username Token Authentication 270
Product Features 32
Protection, Required 270
proxified Object Key 157
Proxified Type id 157
Proxy
HTTPConnectionAccepted-Success 324
HTTPConnectionClosedSuccess 324
HTTPConnectionEstablished-Success 324
HTTPConnectionFailure 324, 334
HTTPMessageReceivedInfo 324
HTTPMessageSentInfo 324
InternalErrorFailure 324
InvalidHTTPFailure 325, 335
MessageFilterEvaluationFailure 325
MessageFilterEvaluationSuccess 325
MessageFilterFailure 325
MessageProcessedInfo 325
PolicyValidityFailure 325
ResourceMappingFailure 325, 334
ResourceMappingSuccess 325
SOAPAttachmentFilterFailure 325
SOAPAttachmentFilterSuccess 325
WSDLRetrievalDeniedFailure 325
ProxyEngine

GIOPBiDirContextInfo 325
GIOPConnection-AcceptedFailure 325
GIOPConnection-AcceptedSuccess 325
GIOPConnection-AssociationExpiredInfo 326
GIOPConnectionClosedSuccess 326
GIOPConnection-EstablishedFailure 326
GIOPConnection-EstablishedSuccess 326
GIOPConnectionFailure 326
GIOPConnectionLimitFailureFailure 326
GIOPMessageEnqueuedInfo 326
GIOPMessageReceivedInfo 326
GIOPMessageSentFailure 326
GIOPMessageSentInfo 326
InternalFailure 327
IORCheckFailure 327
IORDeproxifiedInfo 327
IORProxifiedInfo 327
IORProxifiedNewInfo 327
ParameterCheckFailure 327
ParameterCheckSuccess 327
ProxyObjectAccessError 327
ProxyObjectAccessInfo 327
ProxyManager
AliveInfo 327
ChildExitedInfo 327
ChildSignalledFailure 327
FailedOverInfo 327
FileError 327
InternalFailure 327
ReadyToAcceptInfo 327
RequestLimitFailure 327
StartedInfo 327
ThreadLimitInfo 328
ThreadLowWaterMarkSuccess 328
ThreadThresholdInfo 328
UnknownChildFailure 328
ProxyProcess
CreatedInfo 328
ProxyManagerCommunicationFailure 328
Public Role 258, 293

Q

Quick Start
Typical Configuration Steps 104

R

Replication
IORTableCopyFailure 328
IORTableCopyStartedInfo 328
IORTableCopySuccess 328
LocalModeInfo 328
Replication Interface 149
SetPeerStateInfo 328
Require Signature on SAML 271
Required Authentication 269
Required Protection 270
Resource Mappings 150
Resource Properties
Accessors 291
General 266, 267
Incoming Policy 269
Operations 281
Outgoing Policy 277
Resources 263
Restart
the Proxy 121
the Security Policy Server 121
Roles 258
Actors 259
Administration 261
General 259
Permissions 260
Public Role 259

S

SAML
authentication 243
Injecting SAML Assertions 45, 279
SAML Assertion
Include in Signature 280
Scalability 37
Schema Validation
Examples 55
Schemas importing Schemas 272
schematest 272
Schemas
Preparing Schema files 272
Screened Host Firewall with WS-DBC 65
Securing Web Services with the WS-DBC 31
Security Assertions 29
Security Functionality
Access Control 45
Content Inspection 44
Message Authentication 43
Message Protection 46
Security Policy 238
DBC Base DN 239

Defining 237
General Navigation 241
Storage Location 238
Security Policy Server 36
Authentication Mechanisms 242
Cluster Properties 184
General Settings 185
Installation Overview 78
Installing Keys and Certificates 90
Installing the 87
Installing the SPS 74
Management Interface 89
Prerequisites 62
Properties 185
Startup and Shutdown 82
SecurityPolicyServer
AccessFailure 329
ClientDisconnectedInfo 329
ConfigChangeDetail-Info 329
ConfigChangeInfo 329
ConfigReadInfo 329
ConfigurationFailure 329
ConfigurationIntegrityFailure 329
ConfigWrittenFailure 329
ConfigWrittenSuccess 329
LoginFailure 329
LoginInfo 329
LoginSuccess 329
MarkerInfo 329
PolicyChangeDetailInfo 329
PolicyChangeInfo 329
SkippedEventsInfo 330
StartedInfo 330
SynchronizationFailure 330
Server not reachable 226
Services
CORBA Name Service 152, 163
DBC Monitoring 152, 163
Signature Creation Profile 176
Signature Verification Profile 178
SOAP Attachments
Enable SOAP Attachments 273
Filter Servlet 274
SSL accelerator hardware 133
SSL Profile 165
CA Certificates 171, 177
Certificate Authority 171, 214
Ciphersuite 168
Client Authentication 168
Defining an SSL Profile 211
Example SSL Profile 211
for specific interfaces 211
General Settings 166, 212
Private Key and Certificate 170
SSL Version 167
SSLClient 165
SSLServer 165
SSLAuthentication
CertificateFailure 221, 328
CertificateSuccess 328
OCSPFailure 328
SSLTransport
CertificateFailure 222
ConfigurationInfo 328
HandshakeFailure 221, 328, 336
InternalFailure 328
NoPeerCertInfo 329
PeerRecognizedInfo 329
Status Scripts 84, 218

T

TCP Connection timeouts 227
TCP Keep Alive 228
Traffic Redirector 39
Troubleshooting 217
Address already in use 342
Address lookup failed 341
Admin Console 225
Application doesn't react at all 231
bad certificate 344
Cannot open SSL certificate file 343
Cannot set SSL private key 343
Comm_Failure 230
Common error messages 340
CORBA.COMM_FAILURE Exception 227
CORBA.NO_PERMISSION Exception 227
CORBA.TRANSIENT Exception 226
Error message format 340
Error while reading key file 341
Invalid key format 343
Linux, Solaris Startup 226
No Connection 231
No Listener for IOR Proxification 230
No Permission 231
Problems with logging on SPS 226
Scripts do not work 219
Server not reachable 226
Server Socket bind failed 342

SSL Transport Handshake Failure 344
Starting the Admin Console 225
System Exception 'No Permission´ 342
System Exception 'No Resources' 230
Unable to get local issuer certificate 344
User has no access 227
Trust Relationships 63
Trust Stores and Trusted CAs 203
Trusted CA's 171
Typical Deployment Scenarios 63

U

UserID/Password authentication 244
Username Token Authentication 270
UsernameToken
Preserve Authorization Header 280
Users 241
Authentication Mechanisms 242
General Properties 242
Privileges 250

V

Verbosity Levels 281
versioning of policies 105
Virtual Address 142

W

Web Service Description Language (WSDL) 49
Web Services and SOAP 25
Web Services Security 29
Working Offline 110
WS-DBC Components 34
WS-DBC in the DMZ 66
WS-DBC Proxy 34
WS-DBC Use Cases 41
Introduction 41
Sender-side and target-side WS-DBCs 47
sender-side WS-DBC 42
Single target-side WS-DBC 43
target-side WS-DBC 42
WSDL 49
Bootstrapping 50
usage in the WS-DBC 50
WSDL Get Requests 268
WS-Security Profiles 175
Signature Creation 176
Signature Verification 178
XML Decryption Profile 181
XML Encryption 180

X

XML
DecryptionFailure 330
DecryptionSuccess 330
DSigCreationFailure 330
DSigCreationSuccess 330
DSigVerificationFailure 330, 337
DSigVerificationSuccess 330
EncryptionFailure 330
EncryptionSuccess 330
ParsingFailure 330
SchemaLoadingFailure 330
SchemaLoadingSuccess 330
SOAPStructureAnalysisFailure 330
UnknownSecurityTokenFailure 331
UnknownSecurityTokenInfo 331
UnsignedSAMLFailure 331
UnsignedSAMLInfo 331
ValidationFailure 330, 337
WSSEUsernameTokenInfo 331
XML Decryption Keys 271
XML Decryption Profile 181
XML Digital Signature 46
XML Encryption Profile 180
XML Processing 49
XML Schema Validation 51, 271
Finding out necessary schemas 56
Schemas included in the WS-DBC 273
XPath expressions 287