MICRO FOCUS | CORBA® An OMG Standard

Discover the Future of CORBA

Micro Focus IIOP Domain Boundary Controller (I-DBC) 4.1.0

Administrator's Guide

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
https://www.microfocus.com

2022-02-11

Contents

Part IV Appendices

Preface

The *Micro Focus IIOP Domain Boundary Controller*™ (I-DBC) provides a comprehensive solution for the secure passing of IIOP traffic across domain boundaries. It enables applications based on CORBA and Enterprise JavaBeans (EJB) to interact over network boundaries as in a Business-to-Business e-commerce scenario.

The Micro Focus I-DBC follows well established firewall design principles. It provides principal authentication, cryptographic message protection, fine-grained access control, support of network address translation, security audit, and protection against denial-of-service attacks. It is independent of ORB products, does not require any modification of the application software, and is easily manageable at a centralized point.

Audience

This guide is aimed at administrators managing the Micro Focus IIOP Domain Boundary Controller environment. It comprises three parts:

- Part I, "Concepts and Components" discusses the concepts of the DBC including:
  - general problems concerning the passing of CORBA traffic across firewalls ("Introduction to I-DBC")
  - a detailed description of I-DBC software components ("I-DBC Architecture")
- Part II, "Configuration" describes the set up and configuration of the I-DBC System including:
  - a detailed description on how to use the Administration Console to configure the I-DBC software components ("Configuration of DBC components" and the following chapters),
  - a description of trust relations between DBC components and the required keys and certificates used within the DBC ("Installing Keys and Certificates"),
  - a troubleshooting guide ("Troubleshooting").
- Part III, "Managing Security Policies and Protecting Applications" describes:
  - how access control policies are managed in the DBC and how to use the Administration Console to define such a policy ("Security Policies"),
- how to define regular expressions ("Regular Expressions")Part IV, "Appendices" provides detailed descriptions of different topics related to the Domain Boundary Controller:
  - "Audit Events" lists the audit events used in the I-DBC System,
  - "Error Messages and System Exceptions" lists system exceptions and error messages,
  - "SSL Ciphers" gives more background information on SSL and the ciphers recommended to be used with the I-DBC.

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

Contacting Micro Focus

Our Web site gives up-to-date details of contact numbers and addresses.

Further Information and Product Support

Additional technical information or advice is available from several sources.

The product support pages contain a considerable amount of additional information, such as:

- The WebSync service, where you can download fixes and documentation updates.
- The Knowledge Base, a large collection of product tips and workarounds.
- Examples and Utilities, including demos and additional product documentation.

To connect, enter http://www.microfocus.com in your browser to go to the Micro Focus home page.

Note:

Some information may be available only to customers who have maintenance agreements.

If you obtained this product directly from Micro Focus, contact us as described on the Micro Focus Web site, http://www.microfocus.com. If you obtained the product from another source, such as an authorized distributor, contact them for help first. If they are unable to help, contact us.

Information We Need

However you contact us, please try to include the information below, if you have it. The more information you can give, the better Micro Focus SupportLine can help you. But if you don't know all the answers, or you think some are irrelevant to your problem, please give whatever information you have.

- The name and version number of all products that you think might be causing a problem.
- Your computer make and model.
- Your operating system version number and details of any networking software you are using.
- The amount of memory in your computer.
- The relevant page reference or section in the documentation.
- Your serial number. To find out these numbers, look in the subject line and body of your Electronic Product Delivery Notice email that you received from Micro Focus.

Contact information

Our Web site gives up-to-date details of contact numbers and addresses.

Additional technical information or advice is available from several sources.

The product support pages contain considerable additional information, including the WebSync service, where you can download fixes and documentation updates. To connect, enter http://www.microfocus.com in your browser to go to the Micro Focus home page.

If you are a Micro Focus SupportLine customer, please see your SupportLine Handbook for contact information. You can download it from our Web site or order it in printed form from your sales representative. Support from Micro Focus may be available only to customers who have maintenance agreements.

Part I

Concepts and Components

In this part

This part describes the central concepts underlying the Micro Focus I-DBC. "Introduction to I-DBC" discusses general problems that arise when trying to securely pass IIOP traffic over domain boundaries. "I-DBC Architecture" discusses the product features of the Micro Focus IIOP Domain Boundary Controller.

This part contains the following chapters:

Introduction to I-DBC

Internet-wide CORBA- and EJB-based interactions require a direct connection to the public Internet. If not carefully controlled, a direct link to the public Internet leads to a broad range of security risks. A key challenge is to eliminate the security risks while continuing to use existing or upcoming technology not specifically built for high-security scenarios.

Typically, firewalls are installed at the network boundaries to meet the requirements like authentication, fine-grained authorization and access control, cryptographic message protection, and the ability to securely run services and distributed applications without having to modify them.

Unfortunately, object-oriented middleware such as CORBA and EJB applications are not able to work across firewalls. The reasons for this are explained in the following section. "The Solution" presents how these problems are solved by Micro Focus's Domain Boundary Controller.

The Problems

There are various problems when trying to use the Internet Inter-ORB Protocol (IIOP) across today's firewalls:

- Location-transparency and the dynamic allocation of addressing resources as done by CORBA middleware make it difficult to know in advance which hosts and ports will be used for transactions.
- Addressing information contained in a remote request is invalidated when crossing a network address translating router.
- You may want to run a CORBA application in the cloud (Amazon AWS, MS Azure, Google Cloud).
- You may want to run a CORBA application in a container (Docker).
- You may want to run a CORBA application in a virtual environment (VMWare vSphere, VMWare vCloud).

IP Addresses and Port Numbers

IP addresses and TCP port numbers play an essential role in the binding between a CORBA client and a remote target CORBA object. Generally, CORBA clients rely on the addressing information contained in Interoperable Object References (IORs) to contact CORBA servers: they establish a direct TCP connection to the target server, using the IP address and port number found in the IOR of the target object. Unfortunately, a CORBA server creating an IOR is typically not aware of any firewalling or address translation strategies performed at the domain boundaries.

Packet Filter Firewalls

Packet filter firewalls located between client and server are likely to be configured in a way that requests from the client will be blocked. The application will be unable to complete the remote invocation. In order to enable all CORBA invocations on objects behind the packet filter firewall, the firewall would have to be opened for connections to all hosts running CORBA servers and a broad range of port numbers: any port a CORBA server could be listening on. Considering the security implications of such a firewall

configuration, this is not an option. The protection at the domain boundary would be severely compromised.

Firewalls at the Internet Boundary

The problem appears as soon as IIOP traffic has to pass a packet filter firewall, regardless of the location of that packet filter firewall. In the most obvious case, IIOP is spoken end-to-end over the Internet and thus has to cross packet filter firewalls at the border to the public Internet.

Firewalls Within a Corporate LAN

In another, less obvious case, a servlet enhanced web server uses HTTP to interact with thin clients over the Internet. Internally it uses IIOP to access business logic implemented inside the intranet. Usually, web servers or application servers that are accessible from the public Internet must be separated from the intranet with packet filter firewalls. Thus, the IIOP traffic exchanged between the servlets at the web server and internal application servers must pass at least one packet filter firewall. Also, in large organizations, firewalls separate different divisions or departments. Any CORBA service which needs to cross department boundaries, is faced with the firewall problem.

Network Address Translation

The previously described problems become even more severe if Network Address Translation (NAT) is used. In this case, an IOR produced by a CORBA server behind the NAT packet filter firewall contains the IP address of the server host in the local network. If this IOR is delivered to external clients on the other side of the NAT firewall, any connection attempts to the CORBA server using the IP address contained in this IOR will fail. The internal IP address is not valid in the public Internet. NAT routers are not only employed at Internet boundaries. They are also used internally, for example, after a corporate merger to connect two networks.

Also, sensitive information about the infrastructure behind the NAT packet filter firewall will leak to the outside via this IOR, e.g., the IP address and parts of the object key. NAT properties and their implications are discussed in more detail in section “Basic NAT Properties”.

Virtual Private Networks

A common approach for establishing extranet configurations involves the use of Virtual Private Networks (VPNs). Here, encrypted links across the public Internet tunnel traffic from one domain to another. Problems arise when an individual domain which is part of the VPN has a security hole, e.g., an uncontrolled channel to the public Internet. In this case, any partner domain attached via the common VPN to this insecure domain is at risk. That is, in addition to protecting the tunnels across the public Internet, the connection of the VPN to each domain must be protected by a firewall. This firewall enforces the separation of domains with regard to the security responsibilities and trust relationships. If CORBA middleware is applied, the firewall causes exactly the same problems as described in the previous sections.

The Solution

The only viable solution to the aforementioned problems is an application level firewall. The Micro Focus IIOP Domain Boundary Controller (I-DBC) is a functionally enhanced IIOP firewall component. It is applied as part of an existing firewall installation at the domain boundaries between an administrative domain and the public Internet. Operating as an application layer gateway for IIOP the I-DBC protects the Intranet from illegal access while enabling inter-domain interactions of distributed business applications. The I-DBC inspects and modifies IIOP messages passing through, thus enabling a secure transmission of CORBA requests and replies across packet filter firewalls and NAT routers. The I-DBC filters out hostile and destructive messages and provides connection management. Additionally, it enforces fine-grained access control to guarantees that only trusted peers can connect.

The I-DBC is a pluggable security solution. Business applications do not have to be adapted. The product is suitable for use with any distributed CORBA 2.x compliant application as well as with Enterprise Java Beans using RMI over IIOP. The Micro Focus I-DBC provides external monitoring which makes it suitable for use in traffic redirection scenarios for high availability demands.

This document is specifically for Micro Focus I-DBC for Cloud, CORBA and Virtual Environments (CCVE), which is based on the standard edition of I-DBC. However, both this guide and the ***Deployment Guide*** cover features that are only present in the enterprise edition.

Product Features

The I-DBC supports arbitrarily complex CORBA- and EJB-based business-to-business applications as well as inter- and intra-organization information systems. While being independent of specific ORB products, it supports elementary special features of leading CORBA products. Employing the I-DBC does not require any modification of the application software. Features of the I-DBC include:

- The support of every CORBA or EJB interaction pattern, i.e., the passing of object references between all parts of the application.
- Secure and controlled traversal for inter-domain object invocations across existing packet filter firewalls.
- The handling of Network Address Translation (NAT).
- **Authentication**: access to CORBA Services can be restricted to authenticated clients only. The I-DBC supports the following mechanisms for client authentication:
  - Anonymous access
  - Authentication by IP-address
  - Public Key X.509 certificate-based client authentication (SSL)
  - User ID/password schemes
  - Two-way authentication with RSA SecurID
- **Message protection**: communication between clients and the I-DBC can be protected using SSL to ensure integrity and confidentiality.
- The I-DBC can perform **content inspection** by validating the parameters of incoming messages. Flexible filter rules for operation parameters can

be defined with the Administration Console. Messages carrying arguments that do not conform to the filter rules will be rejected.

- **Access control**: the I-DBC performs authorization for incoming requests with access policies based on:
  - the resource IOR
  - the operation name in the request
  - the arguments of an operation
- Existing user and policy management can be integrated by using **LDAP**-enabled directory servers.
- The I-DBC provides **auditing** facilities, which can be configured to send event notifications.
- For **high availability and/or scalability** demands multiple I-DBCs can be operated in a cluster. The service of the I-DBC will still be provided even if some hardware or software component fails.
- I-DBC configuration and policy administration are conveniently managed using the **Administration Console**.

I-DBC Components

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

- the *I-DBC Proxy*, which transparently intercepts CORBA requests, replaces addressing information in passing IORs, and enforces security policies,
- the *Security Policy Server* (SPS), which manages these policies, and
- the *Administration Console*, an advanced GUI tool, which allows for convenient configuration of the whole I-DBC system and managing security policies.

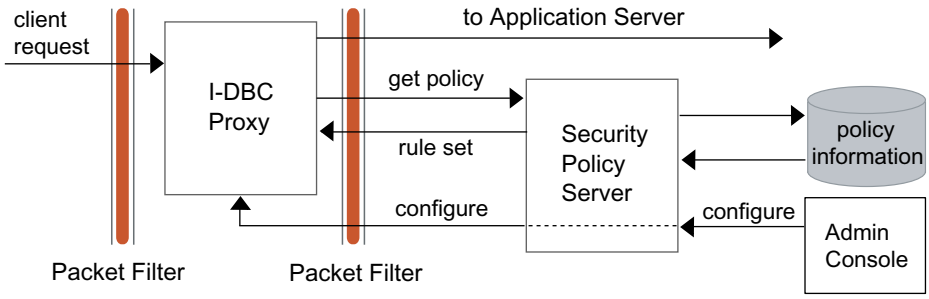


Figure 1 Separation of I-DBC Components

Figure 1 illustrates how a request is processed by the I-DBC System. The main stronghold towards the public Internet is realized by the *I-DBC Proxy*. The Security Policy Server and the Administration Console are usually located inside the protected internal network, i.e., behind the I-DBC Proxy and any existing packet filters.

The I-DBC Proxy supports strong encryption on both external and internal links. On internal links encryption is optional, but recommended. On external links, i.e., for any interactions taking place across the public Internet, it is highly recommended to use strong SSL encryption with full key length to ensure confidentiality and integrity.

The Security Policy Server comprises components that support Authentication, Authorization, Auditing, and Administration. These components interact to provide controlled and accountable access to resources at application servers in the internal network. Common standards are supported to allow a seamless integration into an existing security and authorization infrastructure.

The single components of the I-DBC are explained in more detail in “I-DBC Architecture”.

Summary

The Micro Focus IIOP Domain Boundary Controller (I-DBC) is an application-layer firewall dedicated to the controlled and secure transfer of IIOP traffic across an enterprise’s domain boundary. The I-DBC analyzes the structure of IIOP messages and headers passing through. It makes selective forwarding decisions and enforces fine-grained access control at object and operation level. The processing of IIOP messages enables distributed business applications to operate across existing packet filter firewalls without compromising the protection established at the enterprise’s domain boundary. There is no need to modify existing CORBA- or EJB-based business applications.

The I-DBC supports arbitrarily complex CORBA- and EJB-based business-to-business applications as well as inter- and intra-organization information systems. The I-DBC integrates with existing user and policy management by using LDAP-enabled directory servers. While being independent of specific ORB products, it supports elementary special features of leading CORBA products.

I-DBC Architecture

In this chapter, we will describe the software architecture of the I-DBC system. We will introduce and explain many terms specific to the product and the problem domain. If you have never configured an I-DBC before, we suggest you read this chapter before proceeding to Part 2. Also, if you're just interested in some of the inner workings of the I-DBC, this is the right place for you.

Physical and Logical View of I-DBC System Components

The I-DBC can be operated with all common architectures to build an organization's firewall. The most typical firewall architecture used in sensitive environments with high load is the screened subnet architecture. Located in between two routers inside the perimeter network, dual-homed host computers provide a very high level of control by running various types of application-level gateways, e.g., an SMTP or HTTP Proxy Server. Towards the Internet, the perimeter network is protected by an exterior router which permits connections to only a small set of services. At the border to the internal network, the interior router protects the internal network both from the Internet and the perimeter network. Direct access from the public Internet will be completely blocked by the interior router thus providing a layer of protection redundancy. Note that each of the routers may be configured to perform Network Address Translation (NAT).

The Micro Focus I-DBC is typically run on such a dual-homed host as described in the screened subnet architecture. All CORBA traffic between clients in the public domain (Internet) and servers in the protected domain is routed through the I-DBC. The Security Policy Server, responsible for Authentication, Authorization, Auditing, and Administration is located on hosts inside the internal network.

The following diagrams show where I-DBC Components can be distributed in a network environment. The upper part of Figure 2 depicts the screened subnet architecture (network view). The lower part presents the abstract logical view of I-DBC components.

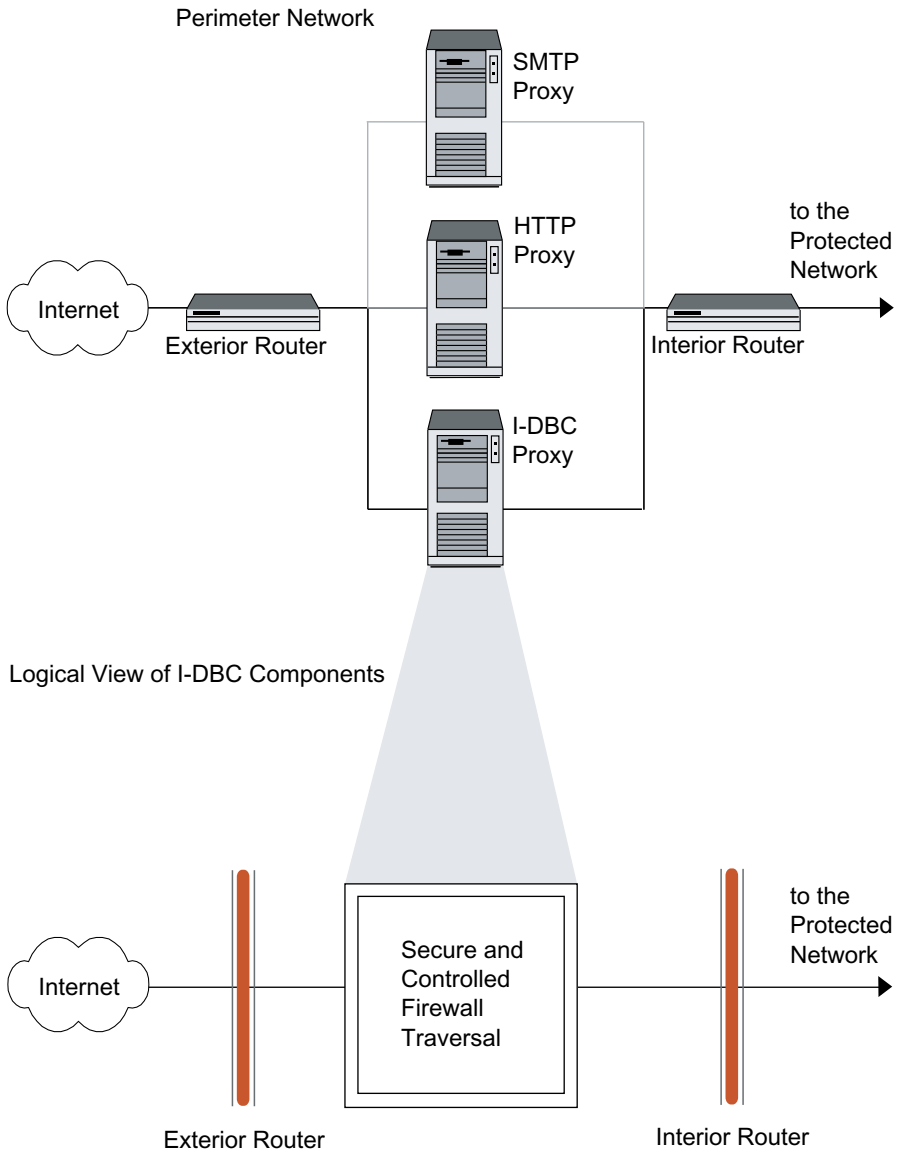
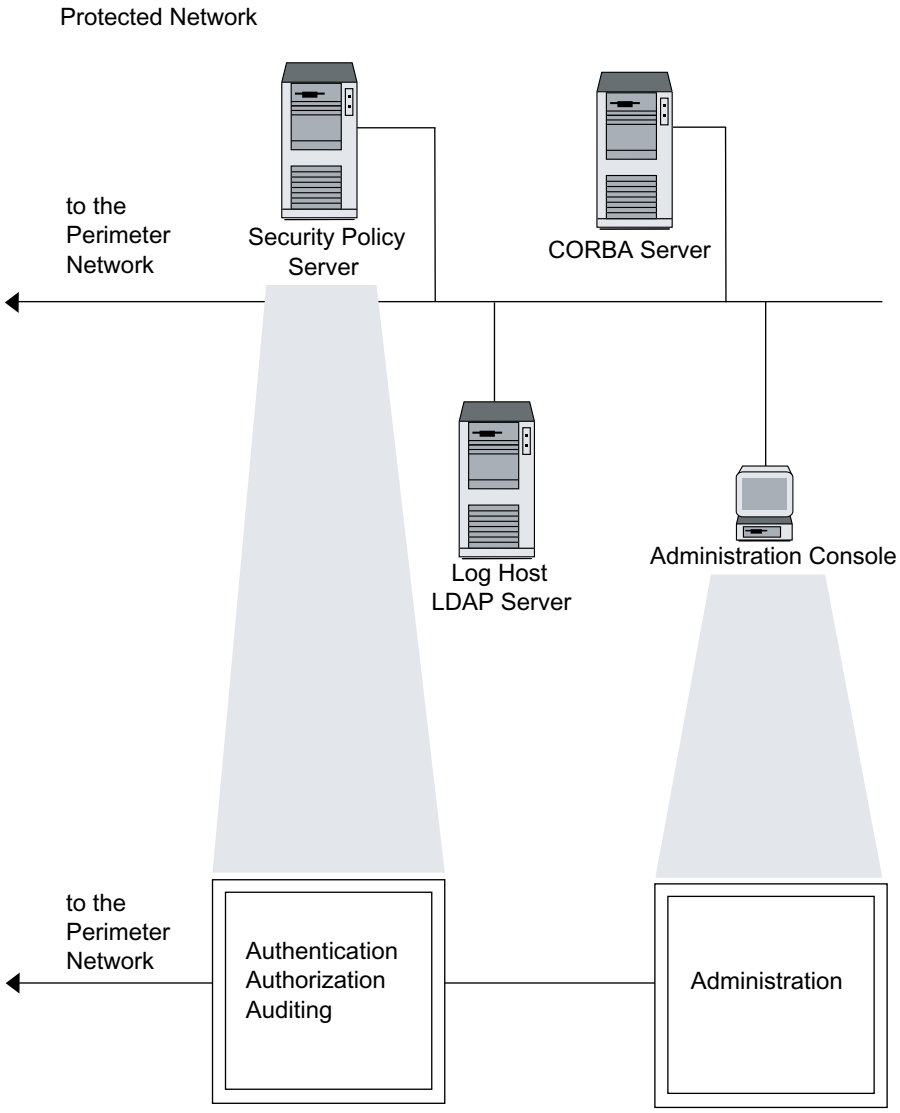


Figure 2 I-DBC Components in a typical deployment Scenario



I-DBC Components in Detail

The I-DBC System comprises three components: The I-DBC Proxy, the Security Policy Server, and the Administration Console. I-DBC components can be distributed onto multiple hosts, located in different subnets. To provide a high level of security these subnets can be protected by packet filtering routers. In environments with less stringent security requirements a single node can host all I-DBC components.

The following subsections briefly describe the function of each I-DBC component. A detailed description of the I-DBC Proxy is given in "The I-DBC Proxy in Detail". The Security Policy Server and the Administration Console are explained more thoroughly in "The Security Policy Server".

The I-DBC Proxy

The I-DBC Proxy is running on the I-DBC Proxy host located between the public domain and the protected network. As you see in Figure 3, the I-DBC Proxy consists of two software components: The I-DBC Proxy Manager and a number of I-DBC Proxies.

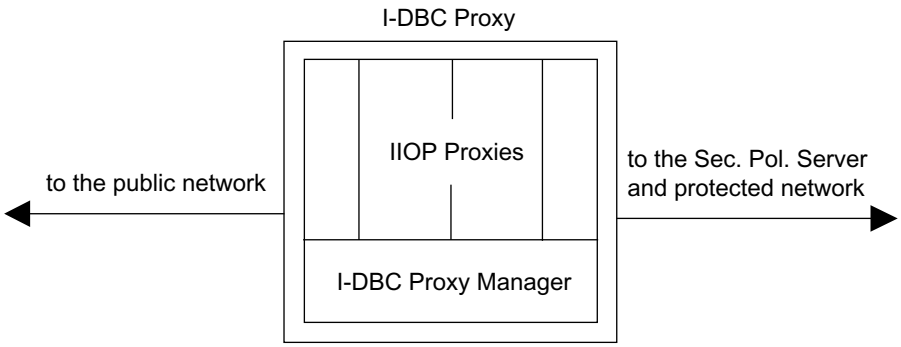


Figure 3 Proxies handle the IIOP traffic

The I-DBC Proxy Manager monitors and controls the operation of the I-DBC Proxies. It also maintains a connection to the Security Policy Server. The I-DBC Proxies are the front line workers, they handle all of the IIOP traffic. Whenever one of the proxies needs to access information not available on the I-DBC Proxy host, e.g., control information or a resolved name, it sends a request to the I-DBC Proxy Manager. The I-DBC Proxy Manager contacts the Security Policy Server to obtain the requested information.

Communication between the I-DBC Proxy and the SPS

The I-DBC Proxy is connected via a dedicated communication link with the Security Policy Server. This link is simply a pair of TCP connections, one in each direction, addressing a known host and port. Considering these requirements, two different configurations are possible:

1. The I-DBC Proxy is connected via a dedicated management and control subnet with the Security Policy Server.
2. The I-DBC Proxy uses two SSL-protected TCP connections on the internal part of the perimeter network. The interior packet filtering router must be configured to allow the two TCP connections.

The Security Policy Server

The Security Policy Server is a centralized server that supports user authentication and authorization, fine-grained access control, flexibly configurable security audit, logging and alarm reporting. The I-DBC Proxy delegates authentication and authorization decisions to the Security Policy Server which provides the necessary security intelligence. The Security Policy Server also interacts with the security information storage server, e.g., the enterprise's LDAP server, to save and retrieve the persistent security policy information. The Security Policy Server provides a management interface, which is used by the Administration Console.

The Administration Console

The Administration Console is an administrative tool that can be run remotely. It can communicate with the Security Policy Server using plain IIOP or IIOP over SSL.

The Administration Console allows for handy configuration of the I-DBC System offering a graphical user interface for set up and maintenance. The Administration Console is explained in detail in “Configuration of DBC components”.

Role-based access control

The Administration Console also allows the definition of complete and detailed security policies. The security model supports roles and resources as defined in the EJB specification and user groups as found in most organizations. The flexible solution with users, groups, roles, and resources as the key elements allows the implementation of a huge variety of authorization models. With the Administration Console’s graphical user interface you can define complex and fine-grained security policies in a structured way. This is explained in detail in the chapter “Security Policies”.

The I-DBC Proxy in Detail

The I-DBC Proxy itself is the core of the Domain Boundary Controller architecture. The I-DBC Proxy is responsible for accepting or initiating SSL-encrypted transport-layer connections across the public Internet from or to peer domains. Currently I-DBC supports SSL v2 and v3, and TLS v1.0, v1.1, v1.2 and v1.3.

Only authenticated peers can connect

Depending on the security policy, the I-DBC Proxy ensures that only authenticated and explicitly entitled peers can connect. Peers can be authenticated via public key certificates (using SSL/TLS), RSA SecurID or User ID/password mechanisms. An additional option is to map source IP addresses to users.

Only authorized requests enter the protected domain

The I-DBC Proxy also passes IIOP traffic in a controlled way through the existing firewall installation (the perimeter network in most installations) and its packet filters. Since all IIOP traffic is guaranteed to flow through that single pipe, the I-DBC Proxy is well suited to perform fine-grained access control on object and operation level according to the access control policy stored and maintained on the Security Policy Server.

The IIOP Proxies

The IIOP Proxies connect to the servers on behalf of the clients, they handle the workload of the complete IIOP traffic between clients and servers. A

client in the public domain connects to the Initial Contact Point (ICP) which corresponds to an initial IOR residing at the I-DBC Proxy host (see Figure 4).

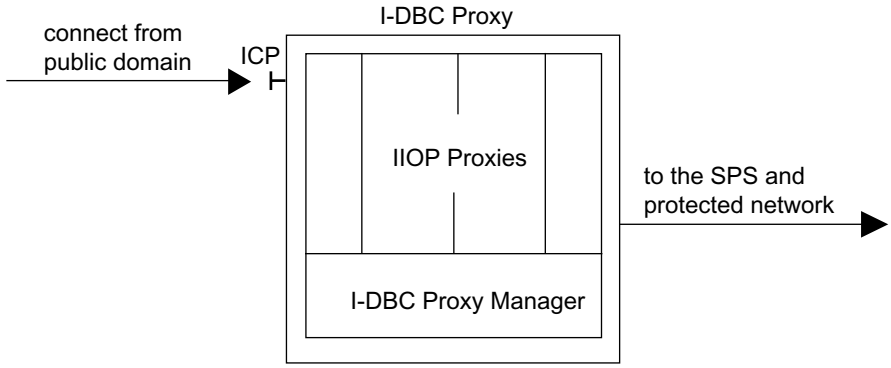


Figure 4 Initial Contact Point at the IIOP Proxy Server

From the perspective of the application located in the public domain, the ICP is the CORBA object which the application initially addresses. At transport layer an ICP corresponds to an IIOP listener socket. Consequently at least one port on the I-DBC Proxy host has to be opened for selected hosts, i.e., those Internet hosts on which the clients run.

The I-DBC Proxy Manager chooses which of the Proxies shall handle an incoming connection. The I-DBC Proxy's tasks comprise connection management, accepting, verifying, and forwarding of IIOP messages, enforcement of access control rules, and, hence its name, proxification of transmitted IORs (IIOP Proxy and *proxy process* mean the same thing in this document).

IOR Proxification

When a client wants to use a service offered by a CORBA server, it needs an Interoperable Object Reference (IOR). If the service can be contacted via TCP/IP, the initial IOR for that service contains – among other things – a TCP address[1] to denote the contact point at which the object can be reached. When an IOR is created by the server's ORB, the TCP address will be the address of the service as known to the ORB.

Replacing the address information in an IOR

Sometimes the address contained in the IOR cannot be used to contact the CORBA server. In this case, the IOR has to be proxified, i.e., the TCP address in the IOR is replaced with another TCP address - the one of the IIOP Proxy, so that clients will contact the I-DBC Proxy instead of the real server. Proxification of IORs is necessary when:

- A firewall blocks direct access to the server.
- The address contained in the IOR is unreachable from the Internet, i.e., the server runs on a non-routed IP address (e.g., 192.168.*.*).
- The traffic is redirected through an I-DBC Proxy to perform access control.

After converting the original IOR into a proxified IOR, the address in the proxified IOR is reachable from the public network. Instead of providing the client with the original IOR, as done for clients which can connect directly to

1. A TCP address consists of a host name or IP address together with a port number.

the server, the *proxified IOR* will be exported to the client application in the public domain.

Proxification of IORs for Initial Contact Points (ICPs) is an administrative task, because it defines which services in the protected domain will be initially reachable from the public domain. When configuring the I-DBC system, you can use the Administration Console to export an IOR to the public domain (see "IOR Proxification").

Passing IORs as parameters

Another situation requiring IOR proxification is the passing of IORs as parameters in CORBA calls. The I-DBC Proxy automatically detects passing IORs and proxifies them accordingly in both directions.

Exporting an IOR Through the I-DBC Proxy

When exporting an IOR through the I-DBC Proxy, two things happen:

- The I-DBC Proxy saves the original IOR.
- A proxified IOR is created.

The *original IOR* will be used by the I-DBC Proxy to contact the server on behalf of a client. Also, clients which can connect directly to the servers, without intermediate firewalls or Network Address Translation, can use the *original IOR* if access control is not desired.

Generated IORs

The I-DBC supports configurations with or without Network Address Translation (NAT) on the interior and/or the exterior boundary, in short, every combination of NAT routers. If you never heard of NAT before, please read section "Basic NAT Properties" on page 16 first. If you do not employ NAT, you can skip this section. This section describes the *intermediate IORs* generated by the I-DBC when NAT is active on the exterior and interior router.

Intermediate IORs are only of relevance to when configuring the firewall rules on your NAT Routers because the addresses contained in the intermediate IORs have to be made accessible from the respective networks.

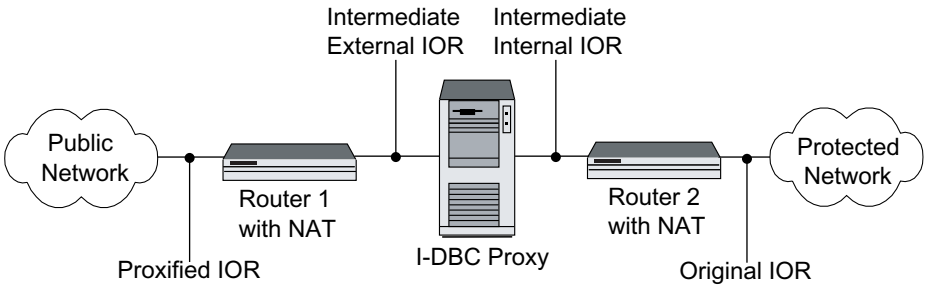


Figure 5 Different types of IORs in a typical I-DBC Scenario

Figure 5 shows the different types of IORs in a typical I-DBC Scenario. The figure will be explained from right to left, i.e., from the *original IOR* as supplied by the CORBA server to the *proxified IOR* as exported by the Administration Console.

Any client connected directly to the protected network can use the *original IOR* to connect to the server. When a router with NAT is employed between the I-DBC Proxy host and the protected network (Router 2), the I-DBC Proxy must use the *intermediate internal IOR* to reach the server in the

protected network. It should be clear that the firewall rules on Router 2 must allow access from the I-DBC Proxy to all TCP addresses contained in the *original IORs* or their translated counterparts respectively as specified by the NAT rules.

Intermediate internal and external IOR

The address contained in the *intermediate external IOR* is used by the I-DBC Proxy to bind TCP sockets to addresses. On these sockets, the I-DBC Proxies on the I-DBC Proxy host will listen for connect requests issued by clients in the public domain. If another router employs NAT on this side of the I-DBC Proxy, the addresses used in the *intermediate external IOR* must again be mapped to match the NAT rules on Router 1. This final IOR is called *proxified IOR*. This *proxified IOR* is the IOR which must be made available to the clients in the public network, because only this IOR will contain the information needed for a connection to the I-DBC Proxy.

If Router 1 does not have NAT configured, the *proxified IOR* will be identical to the *intermediate external IOR*. If Router 2 does not employ NAT, the *intermediate internal IOR* will be identical to the *original IOR*. Again, the two *intermediate IORs* are never exported outside of the I-DBC Proxy.

Proxifying IORs in CORBA Calls

While the previous section described the exporting of Interoperable Object References from the protected domain for use in the public domain, the opposite case is also a basic requirement. IORs can be passed back from a client to a server, e.g., for callback purposes. If the IOR passed is addressing a service in the public domain, the above description can just be mirrored, i.e., public and protected domain appear swapped. Another quite common case is passing back an already exported IOR addressing a server in the protected domain. The I-DBC Proxy converts the IOR back to its original counterpart while it passes from the public to the protected domain, because the proxified variant would not be valid in the protected domain. This is called de-proxification.

Basic NAT Properties

This section describes the properties of Network Address Translation (NAT) that are important for understanding the Micro Focus I-DBC.

IP (Version 4) addresses are a scarce resource, making it expensive for corporations to get an official IP address for every host on their protected network. Besides, an official IP address is not desirable for every host, because it would make the host directly addressable from within the Internet. NAT offers a solution to this problem by mapping a number of hosts to a smaller number of official IP addresses. It is possible to give an entire corporate network full access to the Internet by using just a single official IP address.

The common use of NAT is merely one-way, allowing a corporate host with a private IP address access to any Internet host. A firewall usually restricts the reachable port range. In this scenario, no host located within the corporate network can be reached directly from the outside. Any connection to Internet hosts must be initiated by a corporate host.

If a host of the corporate network is to offer services to be usable from the Internet, such as a Web Server or a CORBA Application Server, it must be possible for clients located in the Internet to connect to this service. Therefore, the private server needs an official IP address. One solution would be to put the private server directly on the Internet, but doing so

would expose the host to attacks. Again, NAT offers a solution. By mapping one of the official IP addresses to the server host's private address at the network border, any Internet user will gain full access to the server. This scenario is depicted in Figure 6. Clients located in the Internet contact the NAT Router (e.g., the service HTTP at port 80). The NAT Router translates the official IP address and port number (e.g. 62.159.77.194:80) into the server's private address and port number (e.g.192.168.1.1:80)

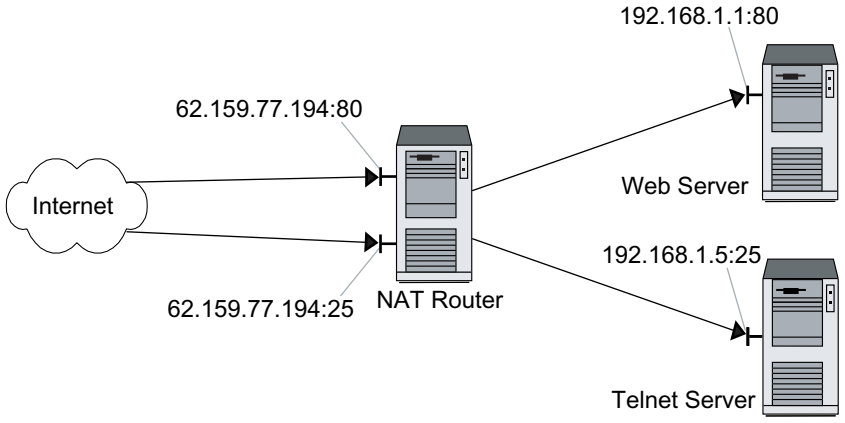


Figure 6 Network Address Translation

To save valuable official IP addresses and to protect non-exported services on the server host, it is wise not to map a complete official IP address, but a certain IP address / TCP port combination to a particular host / TCP port combination on the private server host. Using this technique, any TCP port of a particular range on an official IP address can be mapped onto a different private host, making the most efficient use of official IP addresses.

NAT mappings are asymmetric most of the time, especially in the Internet access case. To the user in the protected network every server on the Internet appears to have its official address. On the other hand, these servers only see the official address of the NAT router. For the servers it looks like a client on the NAT router is connecting (see Figure 7).

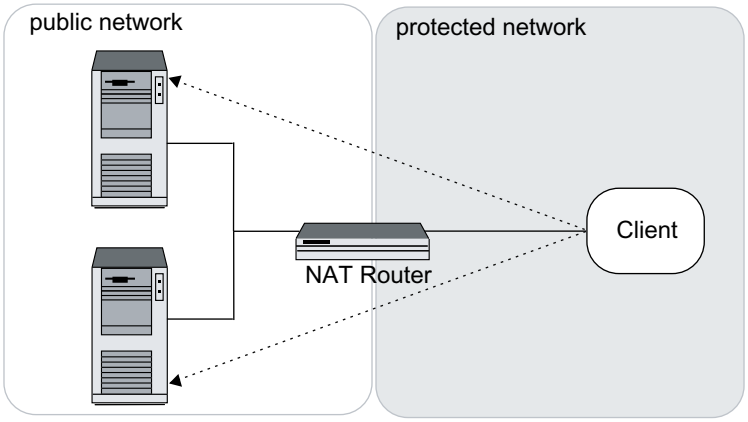


Figure 7 Asymmetric NAT Mapping: The client is located in the protected network

Swapping client and server locations the scenario looks different. A service on a server with a private address might be mapped to a particular port on

an official IP address. This server can be reached from outside the protected network using the official IP address (see Figure 8).

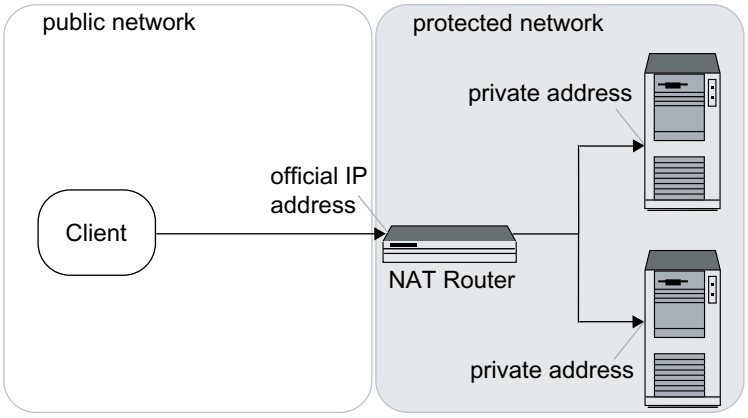


Figure 8 Asymmetric NAT Mapping: The client is located in the public network

Session-based Access Control

The I-DBC can perform access control at different granularities. The coarsest such granularity is that of a user session or access session. The advantage of relying on session-based access control is that it isolates individual clients by restricting the reachability of objects and requires only limited administrative effort to set up. To restrict the actions of a client on a resource that it can reach, however, requires additional access control rules.

An access session is established when a client first tries to connect to the target through the I-DBC. The context created by the I-DBC for this session includes all objects accessed by the client during that session. Any object reference (IOR) that the client retrieves during this session is effectively turned into a capability for just that client and recorded by the I-DBC. No other client will be able to use these IORs as they are restricted to the session in which they were created. When the session terminates, these IORs are no longer usable, so they must be regarded as transient by applications.

When no additional authorization rules are applied, the concept of session-based access control restricts a particular client's accesses to the objects that are usable in this session, and precludes the client from forwarding these to other clients in different sessions. Because a session records the object references, the client can use all objects that are reachable from his initial IOR, i.e., the transitive closure of reachable objects.

Creation of Access Sessions

Every time a client connects to the I-DBC Proxy a new access session for this user is created unless there is already a session for that client. The client's identity is first determined by his IP address. Later, the client can be promoted to another identity, for example, after a successful SSL client authentication.

Replication of access session state

When running multiple IIOP Proxies per host with replication, the IIOP Proxies that accept new connections and establish sessions are chosen by the operating system. The state generated within this access session, i.e.,

the IORs proxified, will be replicated amongst all IIOP Proxies so that it is available for other connections that are mapped to the same access session to different IIOP Proxies.

The policy for accepting connections can be set globally. Depending on the security requirements that are relevant in a concrete scenario, it might be necessary to allow or reject connection requests when only the peer's IP address is known. Acceptable connections can be restricted to specific source IP addresses or IP addresses from certain networks. In setups with stricter security requirements, authentication of peers can be done using SSL client authentication.

Access Session Hierarchies

Every client is first assigned an access session based on his connection's source IP address. For every additional authentication mechanism that is used to authenticate the client, a separate access session is created. These are logically arranged in a hierarchy with the IP-based access session at the top. Information in access sessions (i.e., object references) is available to other access sessions that are located below them in the hierarchy so that the visible object references for a client always include the ones that were available to it with the previous identity, if any. A more detailed description on access session hierarchies and how to configure the user management so that different clients share the same access session can be found in section "Number of Access Sessions and Access Session Hierarchy" on page 186.

Termination of Access Sessions

The IIOP protocol used by CORBA and EJB applications has no notion of access sessions. Hence, there is no facility within the protocol pertaining to access sessions, neither for explicitly opening sessions nor for closing them. Nevertheless, an access session must eventually end so that resources can be freed. The I-DBC uses timeouts to decide whether to terminate an access session. If the last client for an access session has disconnected, the access session is held active for a configurable number of seconds (default is 2), before it is terminated and any associated resources are reclaimed.

If the client keeps such a transient IOR for a specific time (e.g., in a servlet that has a 30 minute life time) and might reuse it within this time frame, the access session timeout of the I-DBC should be prolonged to a value slightly higher than that of the client notion of the IOR's lifetime.

The Security Policy Server

The Security Policy Server controls the security enforcement modules, e.g., the IIOP proxies by providing the necessary security intelligence and communicating all security decisions. The Security Policy Server is responsible for a number of different functions:

- User Authentication
- Access Control
- Event Forwarding
- Configuration Management
- Security Management
- Monitoring

All sensitive data is held by the Security Policy Server to prevent a potential attacker on the I-DBC Proxy host from getting access to security related

information, such as private keys, passwords, log files or even access control lists. The separation between I-DBC Proxy host and Security Policy Server implies that for a number of actions, the processes on the I-DBC Proxy host will have to contact the Security Policy Server to fulfill the request.

Besides the I-DBC Proxy host as the main client of the Security Policy Server, there is another client: the Administration Console, a graphical user interface for the configuration of the system, and the maintenance of the access control rules. This application can be run on any host in the protected network with access to the Security Policy Server. For special requirements, it is possible to direct the management link through the I-DBC Proxy to make the Security Policy Server accessible from within the public domain.

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 only detected after the fact, it is important to have written 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 defined in the I-DBC, see chapter “Audit Policy”. The list of auditable events that the I-DBC Proxy can generate can be found in the appendix “Audit Events”.

Client Authentication

When a client connects to the I-DBC Proxy, the I-DBC Proxy has to determine its identity first. This is called authentication. The peer’s identity is needed for the basic decision whether the client is allowed to connect at all. Later, when the client is connected, access control on different server objects and their operations will be based on the user’s identity as well. The access session concept also requires client authentication.

Clients can authenticate using several different methods, each of them with its specific strengths and weaknesses. The possible authentication methods are based on:

- SSL Certificate
- RSA/ACE SecurID
- User ID / Password
- IP address

How trusted are the authentication methods?

A very important property of each authentication method is the answer to the question: “How sure can we be, that the clients using this specific

authentication method really are who they appear to be?". The so-called *authentication level* describes this property.

The two strongest authentication methods with the highest authentication level the I-DBC Proxy offers are authentication via an SSL certificate and RSA/ACE SecurID.

A user ID/password scheme is a weaker authentication method. It protects against an unauthorized person using an authorized host to connect from, but it does not prevent the loss of or disclosed passwords.

Authentication via IP address is the weakest authentication mode with the lowest ALA, because the client's IP address might not be trustworthy. An attacker could be using IP spoofing or could allocate a previously unused address of a trusted subnet. But the client's IP address is the only information on which we can always rely. If the IP address is all we have, the I-DBC Proxy can be set into "access session for IP address" mode and use the IP address as the only discriminating information.

To conclude this section we will discuss some basic security concepts that will help you understand how the I-DBC Proxy is made secure against unwanted access.

Basic Security Concepts

Authenticating a client as described in the previous section is an important part of the basic security the Micro Focus I-DBC provides. But knowing who you are talking to is not enough. Depending on the required privacy for your communication, you must choose a suitable transport protocol and its properties to protect against a plethora of potential attacks.

Plain TCP was not designed with security requirements in mind, therefore it offers protection only against technically induced data corruption. If you need protection against intentional interception, modification or disruption of transmitted data, you'll need a different protocol.

The Secure Sockets Layer (SSL) protocol was designed to add security properties to TCP. It uses TCP and improves it by transmitting the payload and control information in encrypted form. SSL achieves a high level of security by utilizing different encryption methods and certificates in combination with public key cryptography.

Keys and Certificates

When the I-DBC talks SSL, it can use a certificate to authenticate itself to the client and the server respectively. A certificate is basically the public key cryptographically signed by a certification authority. The private keys of the entity to be authenticated and of the certification authority must be protected against disclosure.

For security reasons private keys are not stored on the I-DBC Proxy host. If an attacker succeeds in compromising the I-DBC Proxy host, he will not be able to get access to the private keys. When private keys are needed by the I-DBC Proxy host, it requests them from the Security Policy Server via a secure connection.

OCSP – the On-line Certificate Status Protocol (RFC 2560) can be activated to get information about the validity of a certificate in the moment of use.

Authentication with RSA SecurID

The I-DBC can be operated with the RSA Authentication Manager (formerly known as ACE Server), RSA's solution to enterprise-wide security. The RSA Authentication Manager provides two-factor authentication technology, where one factor is something that is known (a password or PIN) and the other factor is something that is possessed (such as a code provided by an authenticator). For more information on RSA, SecurID and the RSA Authentication Manager, visit https://www.rsa.com.

Scalability and High Availability

This section describes the architectural concepts for the deployment of the I-DBC in scenarios with high demands on service scalability and availability. For more details on configuring I-DBCs for these requirements see "High Availability and Scalability" on page 3 of the ***Deployment Guide.***

Scalability

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

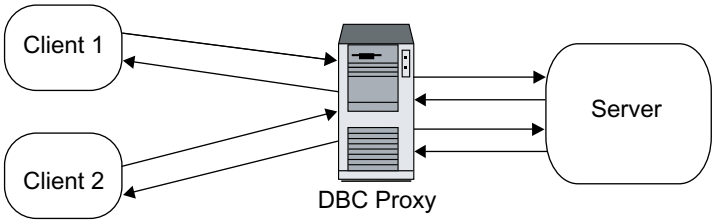


Figure 9 Single I-DBC Proxy, high load

In case of a very large number of clients, or clients with high throughput requirements, the system load of a single I-DBC Proxy may reach a point where the processing speed starts decreasing. Hardware upgrades will remedy the situation, but systems designed for high performance are usually very expensive. An alternative approach is the use of multiple machines in conjunction with a *Traffic Redirector* (see Figure 10).

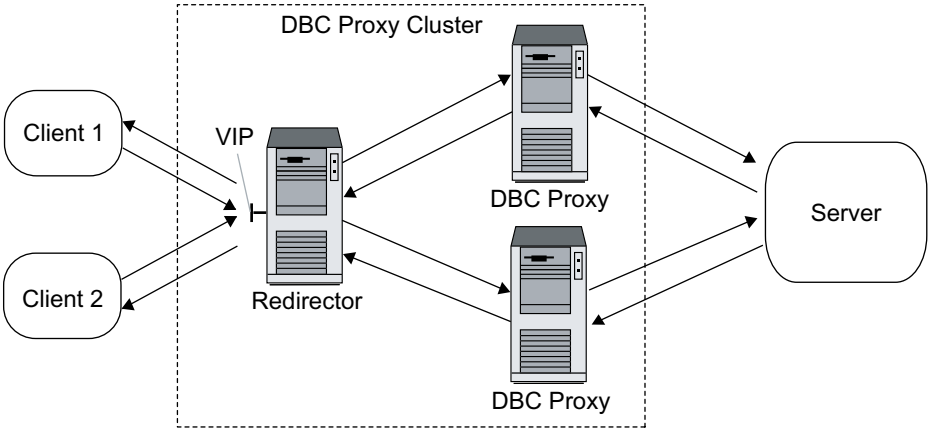


Figure 10 Multiple I-DBC Proxies, balanced load

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 servers. In our case, the servers are individual I-DBC Proxies. The load of

message processing is reduced on the individual I-DBC Proxy machines in the cluster, allowing the deployment of less expensive hardware.

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

High Availability

Another commonly required feature is high availability. In case the I-DBC becomes unavailable the CORBA services behind it can no longer be reached by clients, so downtime of the I-DBC is as costly as downtime of the CORBA services. The architecture described in the previous chapter, 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 notice this fact, and 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 failed machine are affected at all. For the latter, any existing sessions are aborted, but they can immediately be re-establish (now on another cluster machine). To clients it appears as if the failed I-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 I-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.

Flow Control

The I-DBC can be configured to perform flow control. The main goals of traffic flow control are to

- ensure the moderate usage of system resources (especially memory),
- improve behavior against flooding attacks,
- improve the throughput fairness for applications.

Basically, the network read rate at a client side connection is coupled to the write rate at the corresponding server side connection.

Features of the I-DBC Enterprise Edition

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

- **Linear Scalability, High Availability**: The I-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 I-DBCs as well as WS-DBCs.
- **Policy Versioning and Rollback**: The I-DBC internally versions policy and configuration data and supports rollbacks to previous versions in case of administrator errors.
- **Support for Multiple, Concurrent Administrators and Role-based Administration Rights**: The I-DBC is designed for enterprise deployment and fully supports concurrent administrator access, which is controlled by role-based definition of administrator permissions.

Part II

Configuration

In this part

This part is aimed at system administrators who wish to configure the I-DBC system. It includes the following chapters:

Configuration of DBC components

This chapter gives an introduction to the Administration 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 Administration Console as described in the previous chapters. The subsequent chapters describe the panels of the Administration Console in detail.

Administration Console – Introduction

The Administration Console provides a graphical user interface for configuring and managing the IIOP DBC. This guide explains how to configure different DBC components, including

- single I-DBC Proxies and Proxy Clusters (“DBC Proxy Cluster Configuration”),
- single Security Policy Servers and a cluster of Security Policy Servers (“Security Policy Server (Cluster)”),
- Audit Policies (“Audit Policy”),
- Security Policies (“Security Policies”).

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.

Quick Start: Typical Configuration Steps

The Administration 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.

- To start DBC components, enter the following commands:
  - Security Policy Server (as `root`): `/etc/init.d/xdn_sps start`
  - I-DBC Proxy (as `root`): `/etc/init.d/xdn_idbc start`
  - Administration Console:
    - Linux: `<INSTALLDIR>/AdminConsole/bin/AdminConsole`
    - Windows: Click on the **Start** menu of the task bar and choose:

      Programs > Administration Console > Administration Console
- Define Security Policies, e.g., access control rules for resources on the “Resources” panel (for details see “Security Policies”).
- Define audit policies by selecting events for which notifications should be written to an audit log (see “Audit Policy”).
- Configure initial IORs (on the “IOR Proxification” panel). Export proxified IORs to a file and make the proxified IOR available to the client application.

- 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).

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.

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 Administration 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 Administration 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 the section "Write Configuration – Properties").

The Administration 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 Administration 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 by changing to `<INSTALLDIR>/sps/bin`, and typing:

```
./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”).

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.

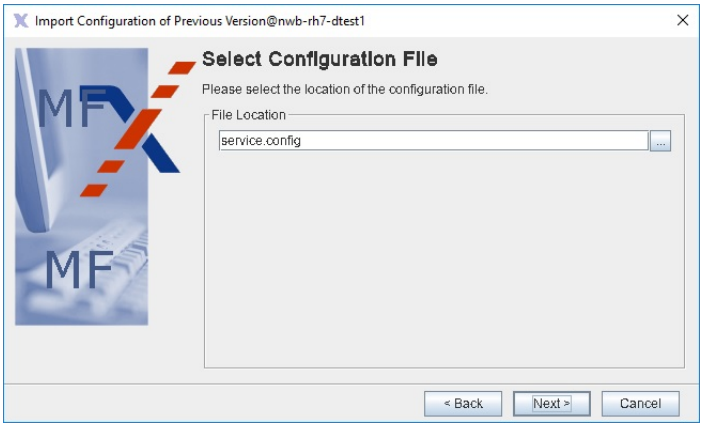


Figure 11 Import Previous Configuration – Select Configuration File

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

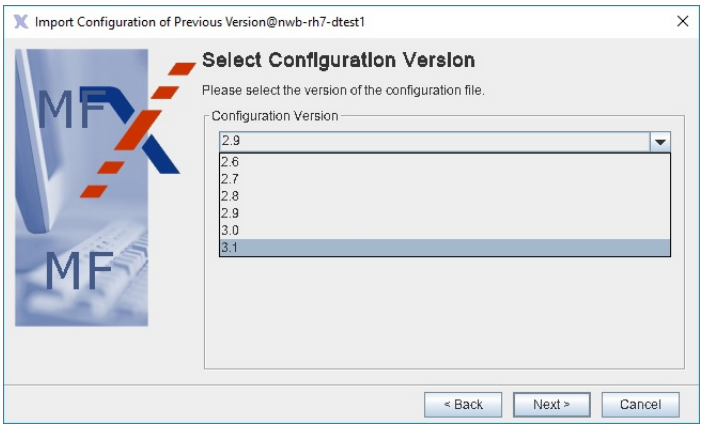


Figure 12 Import Configuration – Select Configuration Version

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

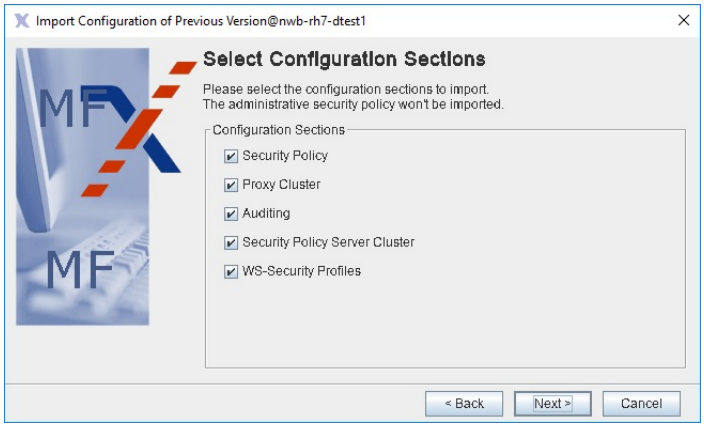


Figure 13 Import Previous Configuration – Select Configuration Sections

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 135.

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.

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` directory 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).

First Start

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

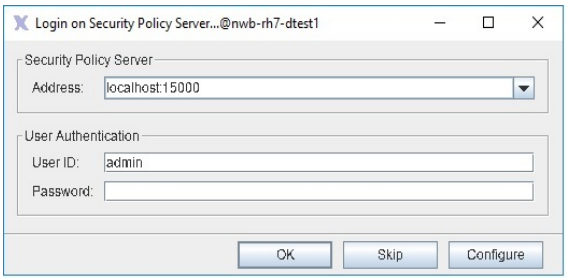


Figure 14 LoginPanel of the Administration Console

Server Address

In the “Address” field you enter the address of the SPS The 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:15000`, or

`192.168.1.90:15000`.

The Administration Console remembers 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 “Preferences”.

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 135). You

should change the admin user's password (see "Changing the Admin User's Password" on page 123 for details).

KEEPING THE DEFAULT PASSWORD IS A SEVERE SECURITY RISK!

Working Offline

Pressing the "Skip" button will start the Administration 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 Administration Console will quit if you click the ✕ in the upper right corner.

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

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

Preferences

Pressing the **Configure** button on the "Login" panel will bring up the "Preferences" panel. The following 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 Administration 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.

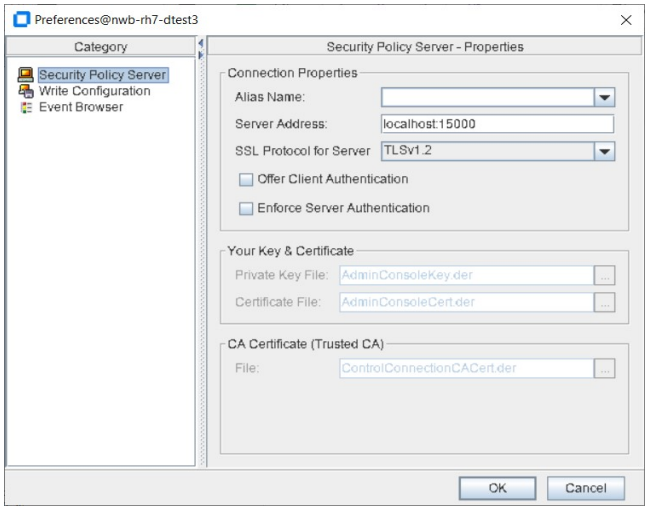


Figure 15 Server Preferences

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 Server, SSL is used. Some SSL properties can be configured. The SSL Protocol for Server preference is set to TLSv1.2 by default. TLSv1.3 is also supported. 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 Administration Console will be generated. The installation of these keys is described in chapter "Installing Keys and Certificates". 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 Administration Console will then connect to the 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 Administration Console logs audit events from the Security Policy Serveras well as local events from the Administration Console. These events are displayed at the bottom of the Administration Console window (explained in detail in "Audit Event Browser"). 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 Administration Console will be displayed in the event browser. Administration 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.

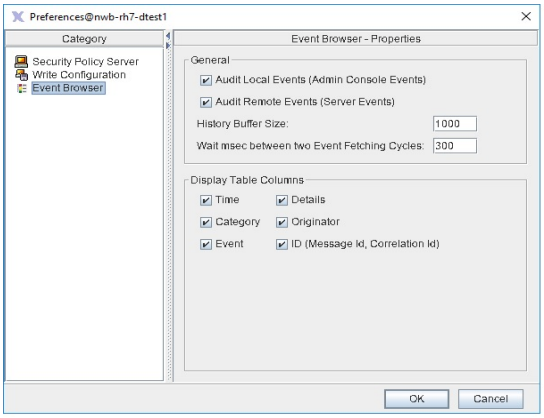


Figure 16 Event Browser Preferences

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).

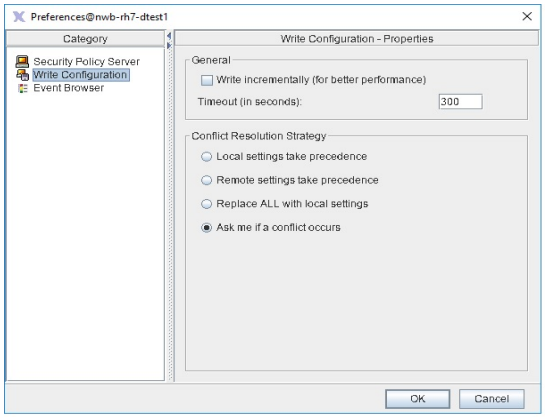


Figure 17 Write Configuration Preferences

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 Administration Console prompts for the passphrase that protects the private key file (the passphrase is `blahblah` by default).

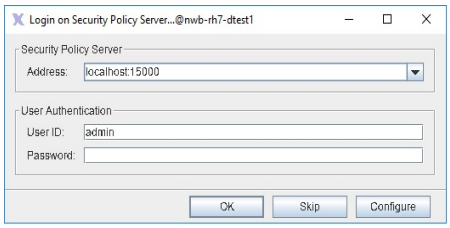


Figure 18 Passphrase Prompt

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

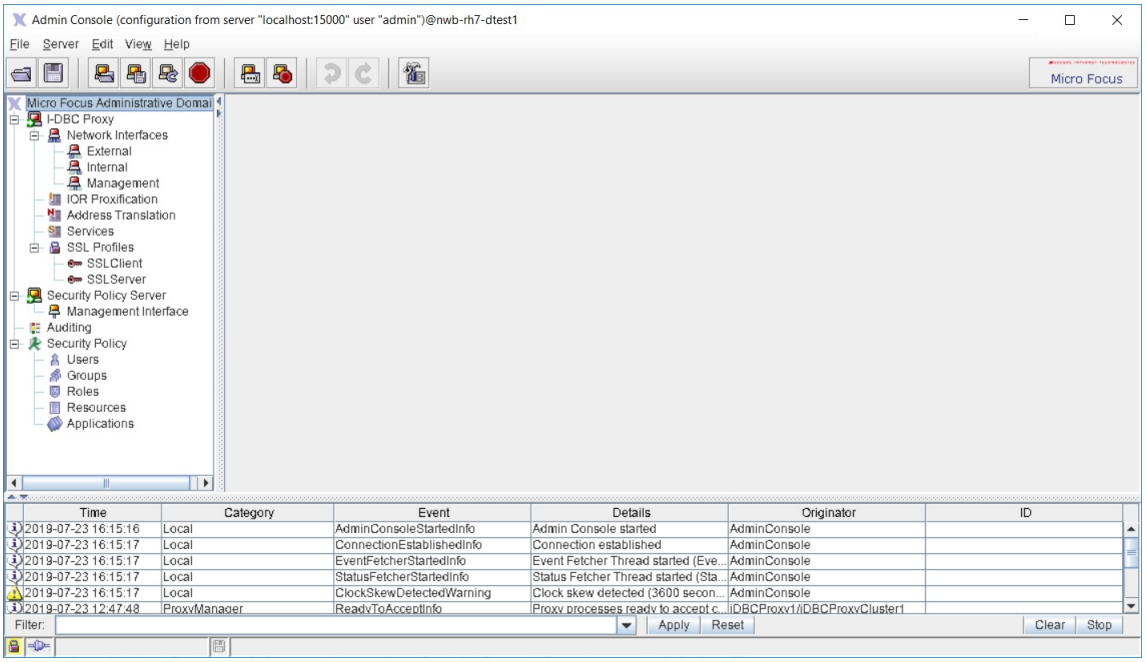


Figure 19 Screenshot of the Administration Console

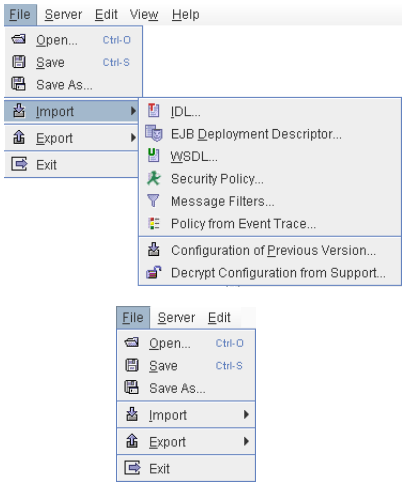
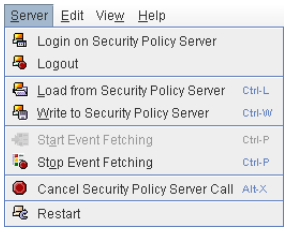
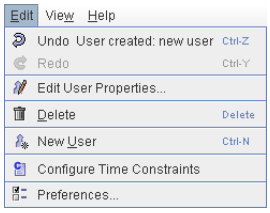
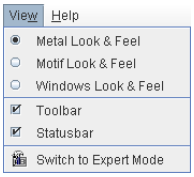
The Administration Console provides a graphical user interface for configuring and managing different CORBA products, i.e., IIOP DBC and the Web Services DBC. This guide will only explain the configuration of the I-DBC. However, you may see menu items in the Administration Console that apply to other products.

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**).

Administration Console Menus

	The Administration Console can store and retrieve configuration data from files (**Open …**, **Save …, Save As …**). The **Import**/**Export** menus offer to several options for importing and exporting configurations or certain parts of a configuration, for example security policy resources may be imported from IDL files (I-DBC) or WSDL files (WS-DBC). The security policy may be exported in XACML format.
	The Server Menu comprises actions performed on the Security Policy Server, like login, logout,and loading and writing of configuration data. Additionally, the event fetcher can be started or stopped and a call to the Security Policy Servercan be cancelled. Selecting **Restart** will restart the currently selected DBC or SPS in the navigation tree.
	The Edit Menu is context sensitive and will adapt depending on the selected DBC component in the navigation tree. This example shows the Edit Menu when a single user in the security policy is selected. The **Preferences** item is always available in the Edit Menu and allows you to set up the Security Policy Server connection, configure the Event Browser, and the settings that apply when writing a configuration to the server.
	In the View Menu 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.

General Organization of the Administration Console

When starting the Administration Console and logging on to an SPS, the current configuration is loaded from the Security Policy Serverautomatically.

Standard mode and expert mode

The Administration 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 Administration Console. The expert mode is explained in detail in the chapter “Expert Mode”.

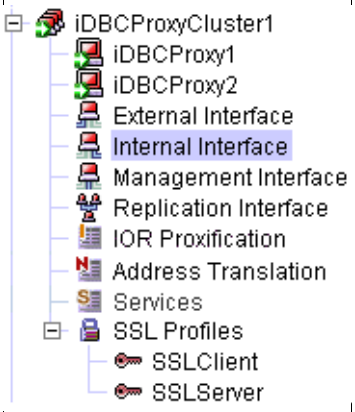
Usually, you will only work in the standard mode. Using the expert mode is not recommended as it is easily possible to render a DBC setup unusable. The expert mode should only be used when advised by technical support.

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

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

When clicking the ⊞ next to the **DBC Proxy Cluster** main node in the navigation tree several subnodes appear (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 Serverwill connect (selected at install time).

Replication Interface

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

IOR Proxification

Configures references to CORBA objects that will be accessible through the I-DBC.

Address Translation

Defines address mappings for **outgoing connections** from the I-DBC Proxies to CORBA servers.

Services

Configure services like DBC Monitoring.

SSL profiles

SSL profiles for communication endpoints.**Security Policy Server Cluster**

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

Audit Event Browser

The audit event browser is displayed at the bottom of the Administration Console (main) window. Every audit event that is enabled according to policy (see "Audit Policy") 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 = ProxyManager" in the "Filter" text field, all audited events in the Category *ProxyManager* 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.

Create a filter from a selected event property

Filters can be created from a selected event property. To do so double-click on an event in the event browser and select a rom in the Event Details panel. Then choose **Create Filter From Selection** from the context menu. A filter will be created from the selection and applied to the events displayed in the event browser.

Additional functionality can be reached via Server/Edit menu or via the context menu (right-click into the event 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").

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"), the Admin Console displays a dialog panel and you can choose between several options on how to proceed.

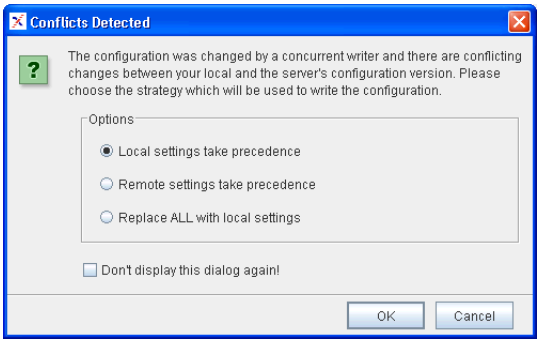


Figure 20 Conflicts Detected

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".

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 Administration 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).

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.

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.

DBC Proxy Cluster Configuration

The DBC Architecture supports multiple clusters of DBC Proxies to ensure High Availability and Scalability (see "High Availability and Scalability" in 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.)

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.

I-DBC Proxy Cluster – General

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

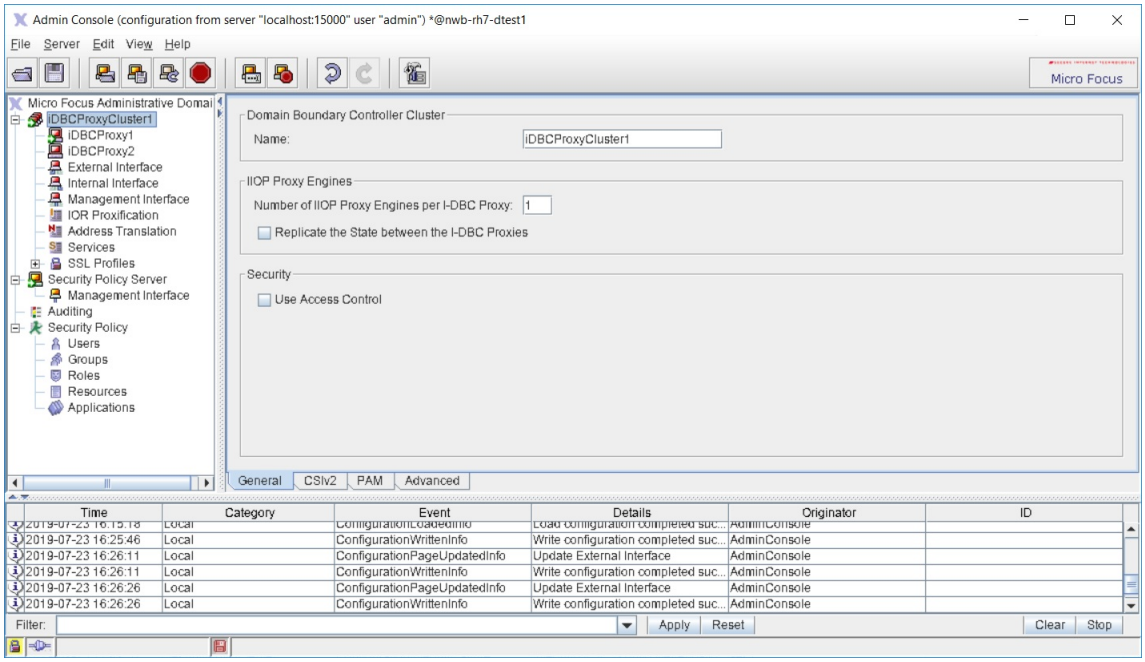


Figure 21 I-DBC Proxy Cluster - General Properties

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 the section “Replication” in the ***Deployment Guide***.

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. “Security Policies”).

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.

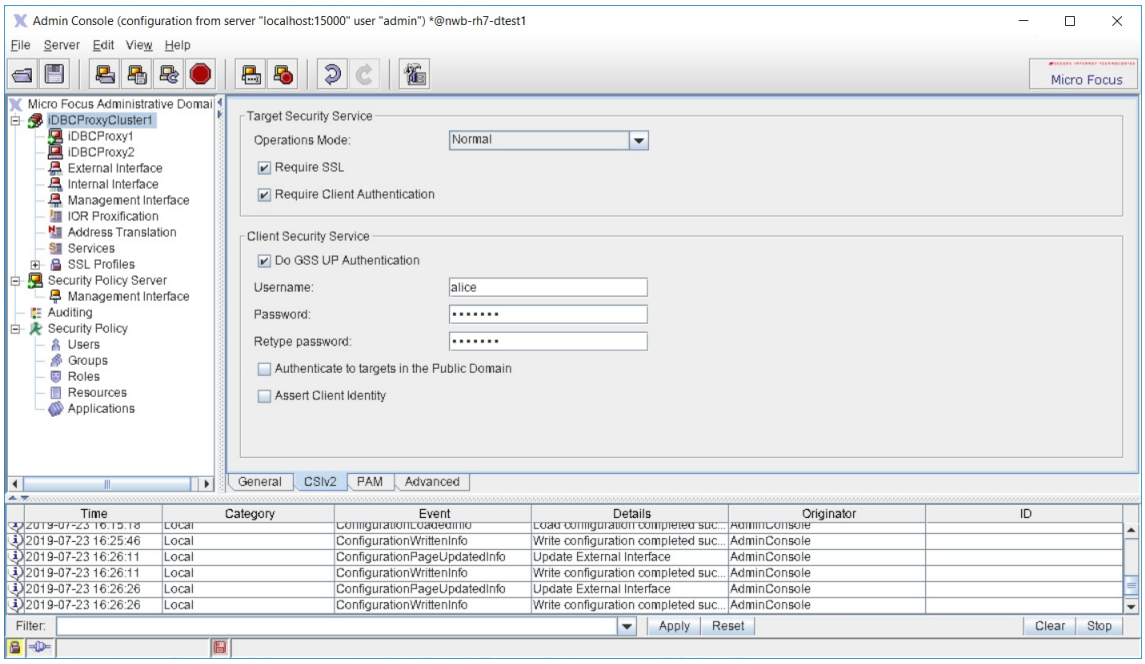


Figure 22 I-DBC Proxy Cluster - CSIv2

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 only)".

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.

I-DBC Proxy Cluster - PAM

On the "PAM" tab 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 how to do that). An advantage of activating PAM is that you can use your existing user database instead of having to re-define 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`. 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").
- **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.

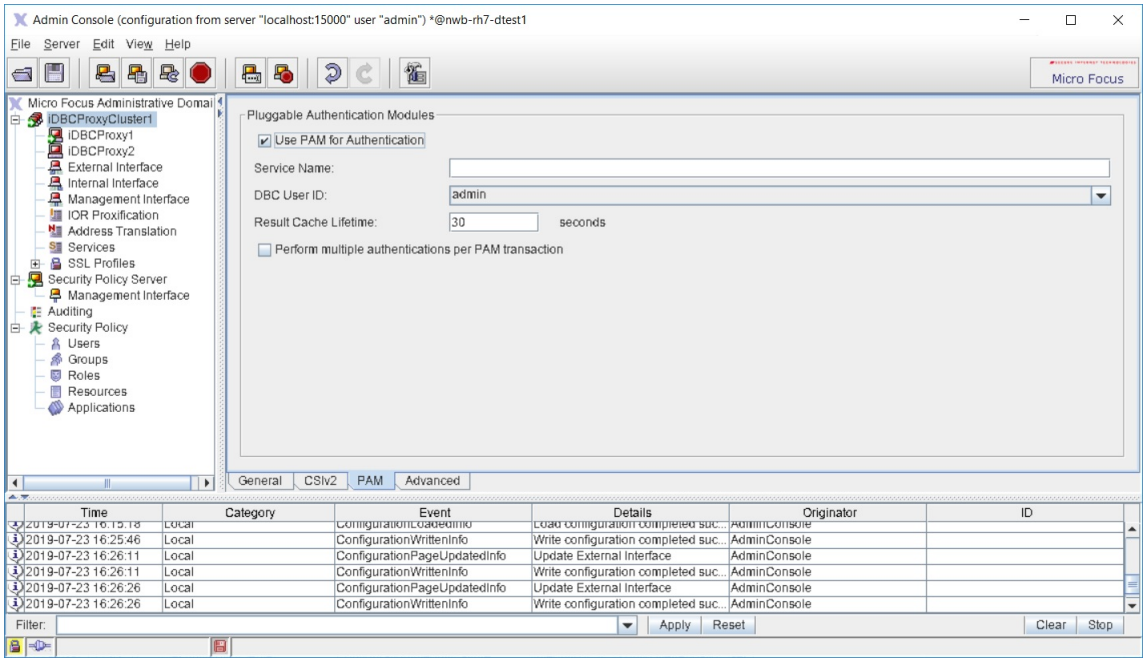


Figure 23 I-DBC Proxy Cluster - Pluggable Authentication Modules

I-DBC Proxy Cluster – Advanced

The "Advanced" tab 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).

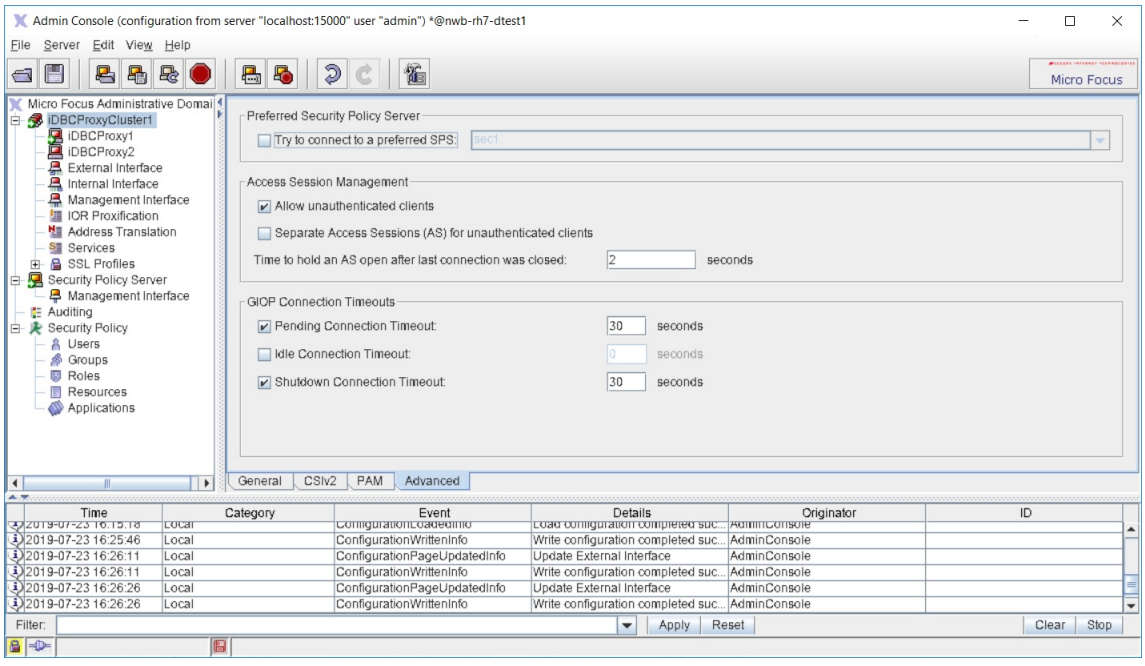


Figure 24 I-DBC Proxy Cluster - Advanced Properties

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”.
- *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”.
- *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 “Firewall Configuration – TCP Connection Timeouts”.

Shutdown connection timeout

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.

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.

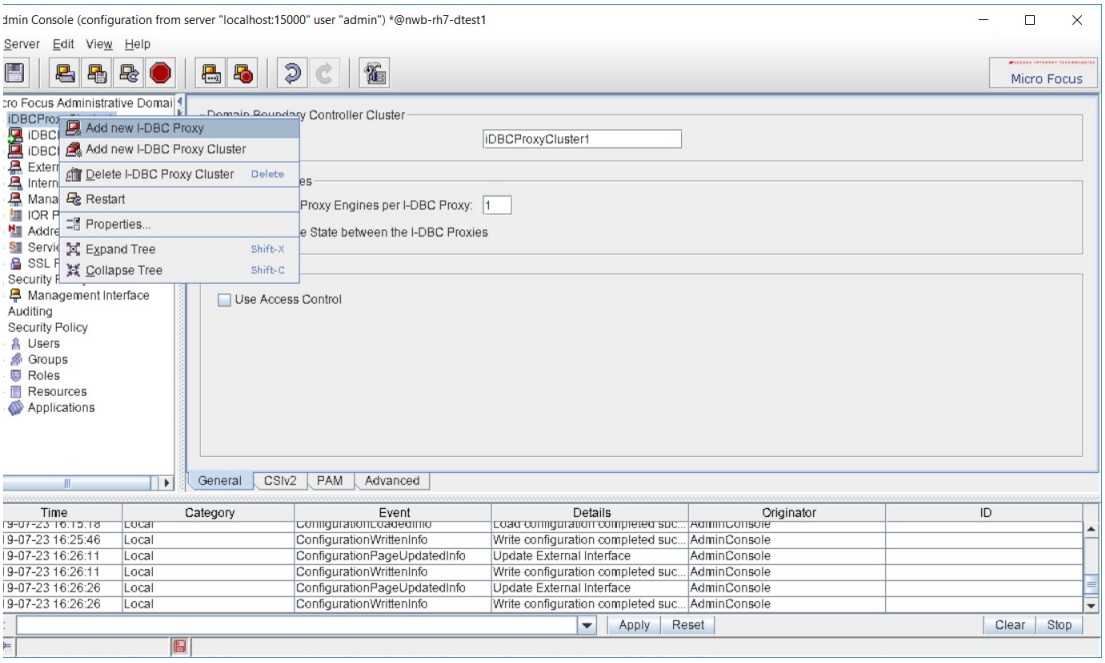


Figure 25 DBC Proxy Properties

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.

DBC Proxy

In the upper part of the "DBC Proxy" panel you can set the name of the DBC Proxy. Names have to be unique for each DBC installation.

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.

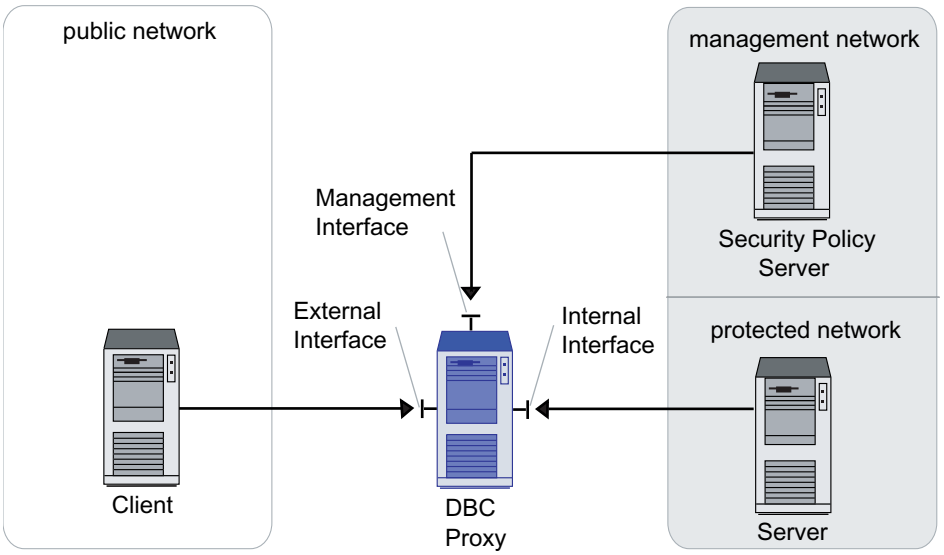


Figure 26 DBC Proxy Network Interfaces

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.

		Logical Interfaces			
		External Interface	**Internal Interface**	**Mgmt. Interface**	**Replication Interface**
Physical Interfaces	Interface A	4	4	4	4
	Interface A	4			
	Interface B		4	4	4
	Interface A	4	4		
	Interface B			4	4
	Interface A	4			
	Interface B		4		
	Interface C			4	4
	Interface A	4			
	Interface B		4		
	Interface C			4	
	Interface D				4

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"). 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.

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.

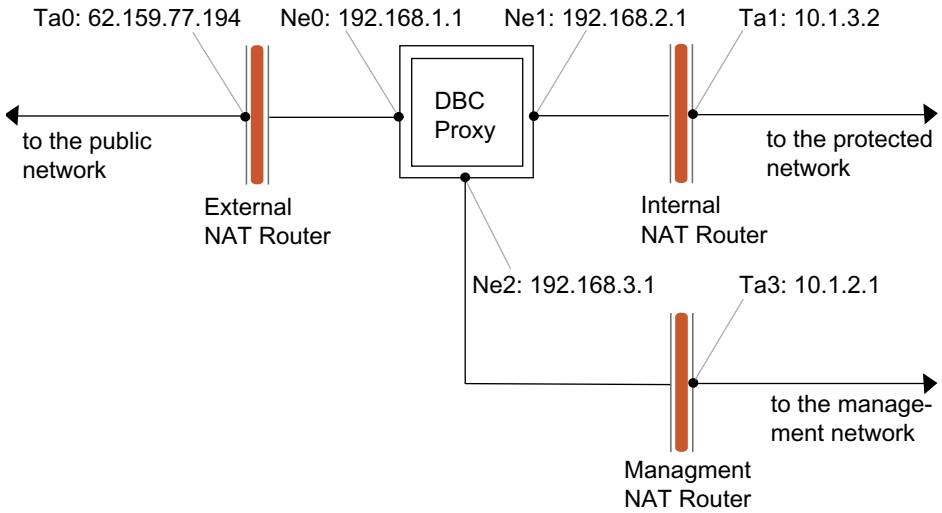


Figure 27 DBC Proxy Network Interfaces

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 Administration Console.

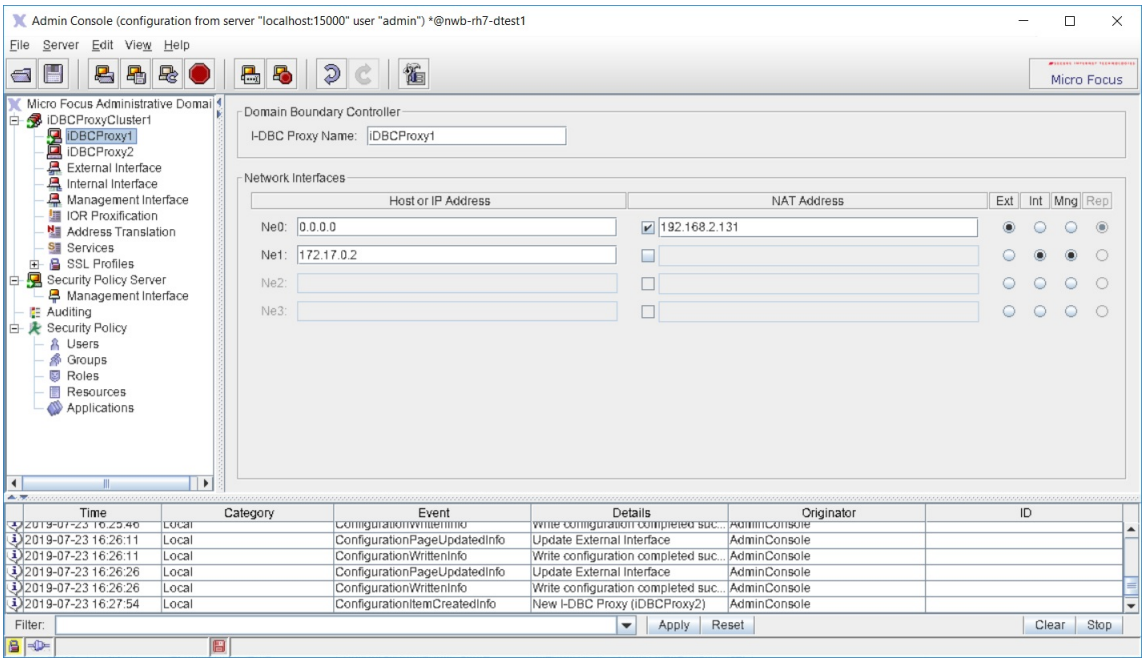


Figure 28 Example for DBC Proxy network interfaces

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”).

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 interface panes are identical. Therefore the following section only explains the settings for External Listeners in detail.

External Interface

Inbound and outbound connections **from and to the public domain** are configured on the "External Interface" panel (see screenshot below).

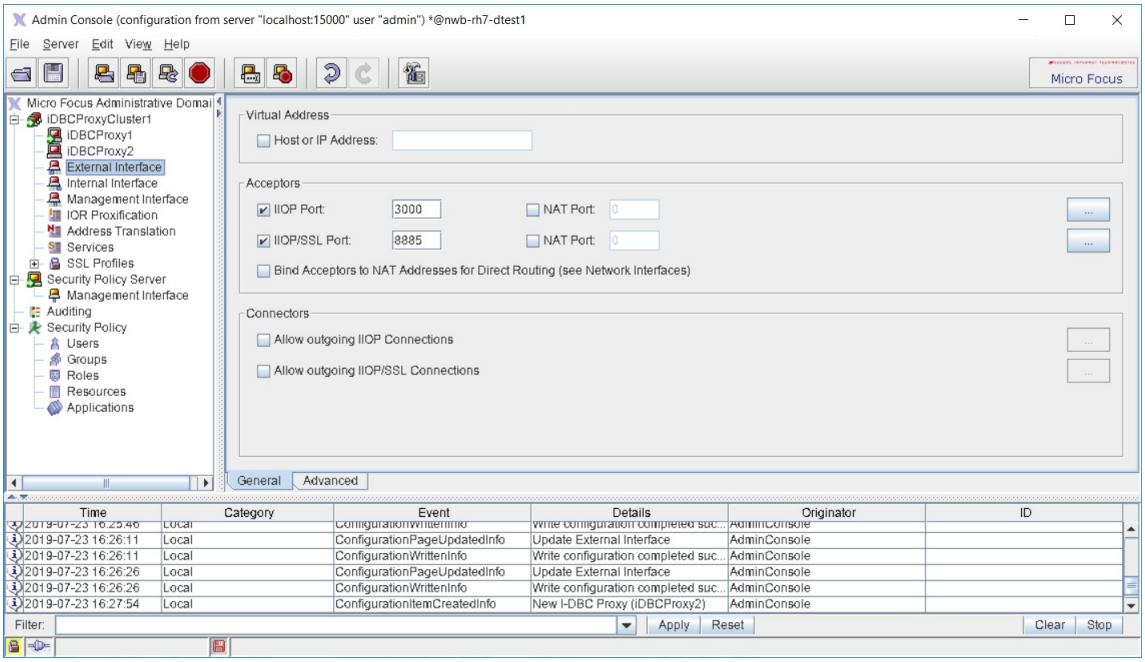


Figure 29 External Interface - General

The following can be configured:

- The Virtual Address of the DBC Proxy in the cluster, cf. "Virtual Address" (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".

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). 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.

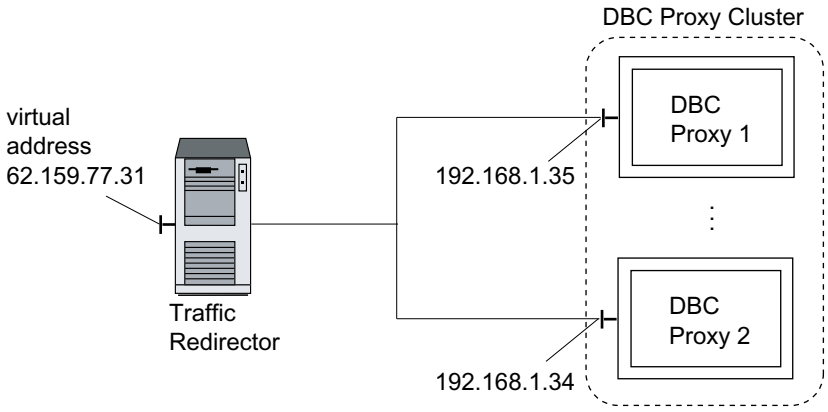


Figure 30 External Interfaces and Virtual Address of DBC Proxies in a cluster

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.

Acceptors

There are two types of Acceptors for incoming connections:

- *IIOP Acceptor*: Accepts incoming connections for IIOP over plain TCP.
- *IIOP/SSL Acceptor*: Accepts incoming connections for IIOP 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" dialog provides additional configuration settings for the TCP transport. To open the dialog click the "..." button next to the Acceptor Port.

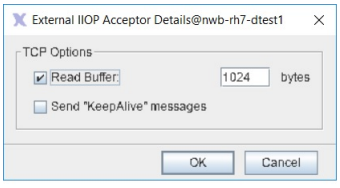


Figure 31 Acceptor Details: TCP Options

The following configuration options are provided:

TCP Options

- *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*

SSL Acceptor Details

The "IIOP/SSL 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).

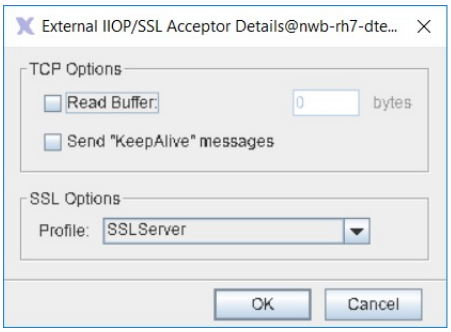


Figure 32 SSL Acceptor Details

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 Profiles".

NAT Port Mappings for the I-DBC Proxy

As discussed in section "NAT Addresses for I-DBC Proxy Interfaces" 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.

Connectors

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

- *IIOP Connector*: Sets up outgoing connections for IIOP over plain TCP.
- *IIOP/SSL Connector*: Sets up outgoing connections for IIOP 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.

External I-DBC Interface - Advanced

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

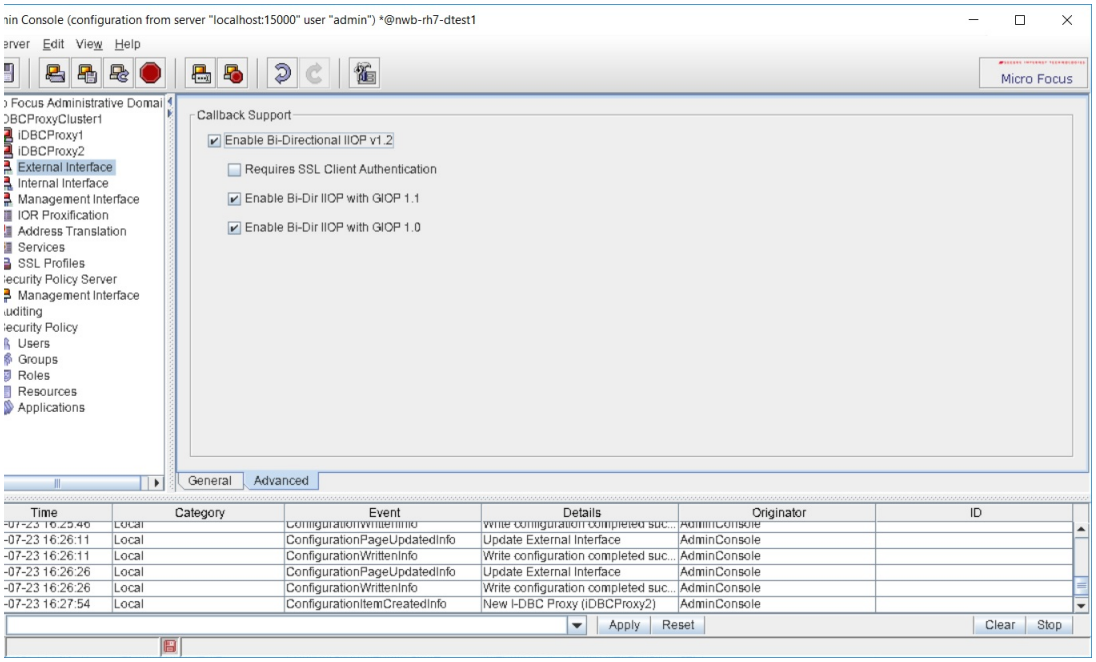


Figure 33 External Interface - Advanced

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.

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.

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 and NAT port for the Management Interface

- *Port*: Choose a port number (default is 14000).

- *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".

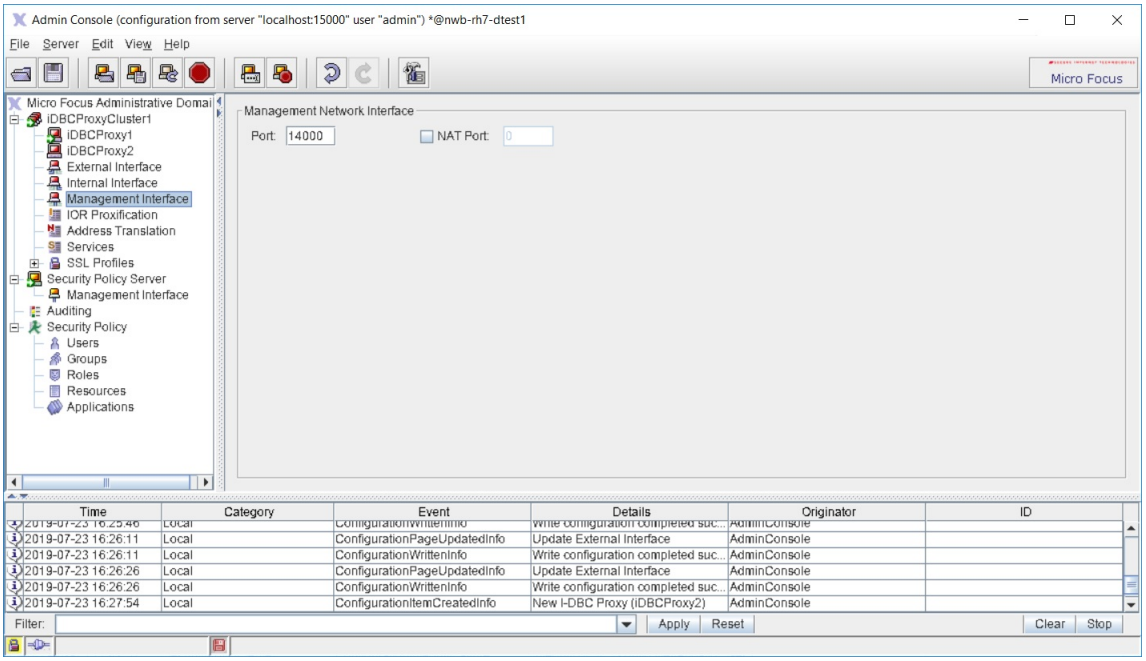


Figure 34 Management Network Interface

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 "Replication" in the ***Deployment Guide***.

Services

Configure services such as DBC Monitoring, CORBA Naming Service, Flow Control, and Connection Limiting.

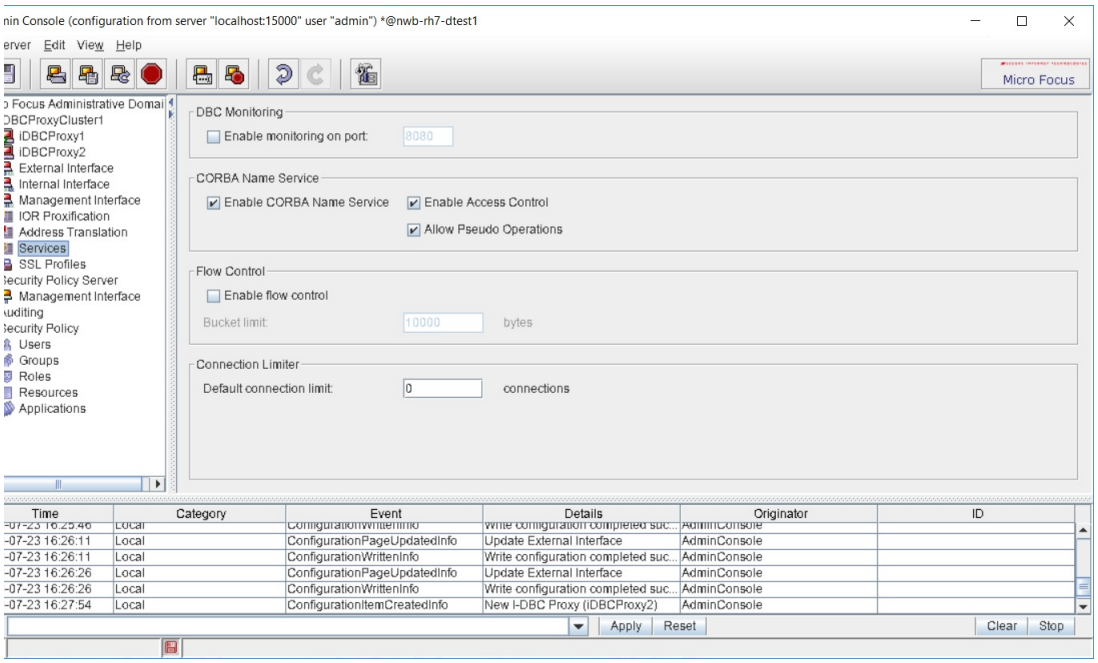


Figure 35 Services

DBC Monitoring

If you want to monitor the Proxy, for instance 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 of such tools are:

- RedHat Piranha
- Cisco CSS
- Dispatcher from other vendors

CORBA Name Service

Enable the CORBA Name Service. This service is a DBC name service functioning basically like a standard Name Service. If “Enable Access Control” is selected, 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.

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 (for example 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.

IOR Proxification

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

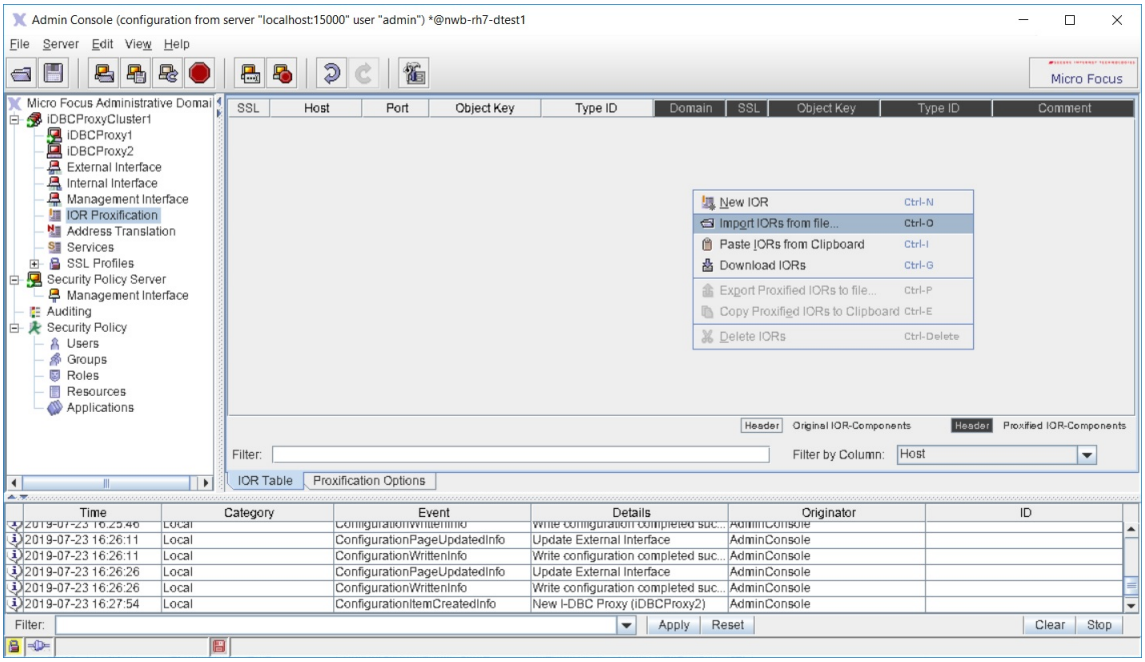


Figure 36 Initial IOR Table

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".

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

The table below shows two different original IORs from a typical 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 Administration 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.

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”). 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.

Proxification Options

Configure advanced proxification options on this tab.

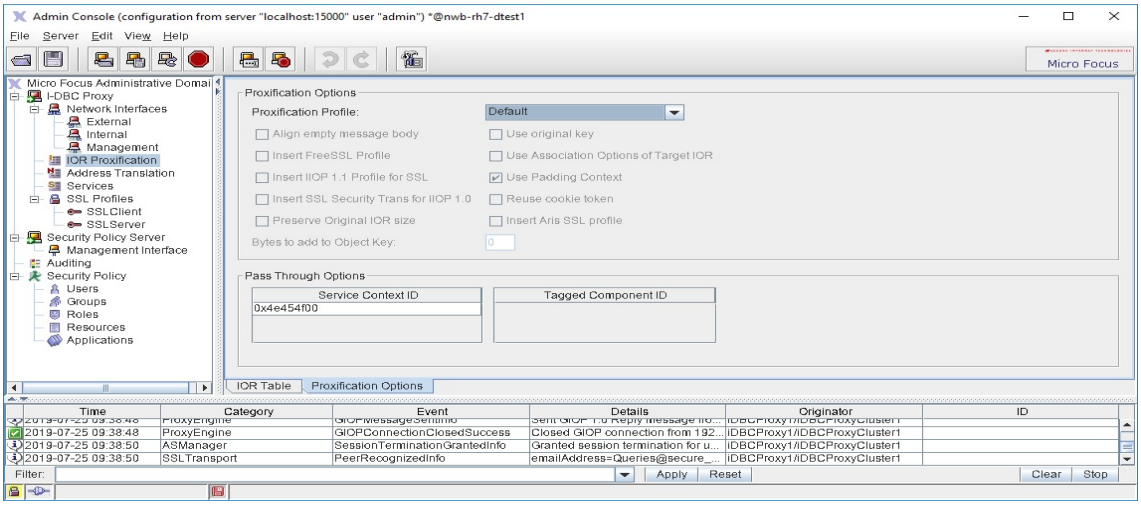


Figure 37 Proxification Options

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)

VisiBroker Smart Agent Relay

You can select the option to Recognize VisiBroker Smart Agent Relay in expert mode, where it is known as `visiOSAgentPerPOA` and is located in `configs.iDBCProxyCluster1.shared.proxy.proxificationOptions`. See the chapter "Expert Mode" for details on using the expert mode.

For more information on the Smart Agent Relay (osarelay), see the section "*The VisiBroker Smart Agent Relay*" in the ***Micro Focus CORBA Add-on for Cloud, Containers & Virtual Environments Installation and Configuration Guide***.

Pass Through Options

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

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 38).

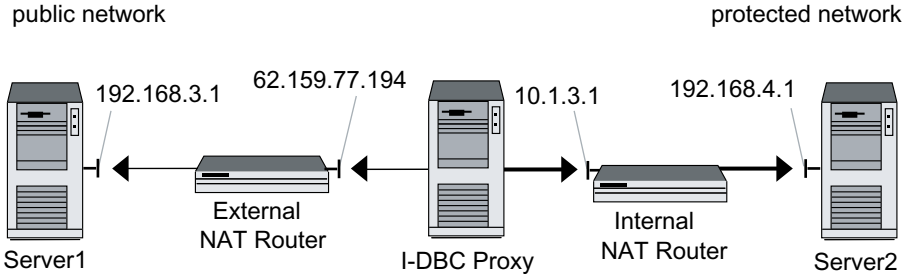


Figure 38 Address Mappings

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 38). 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).

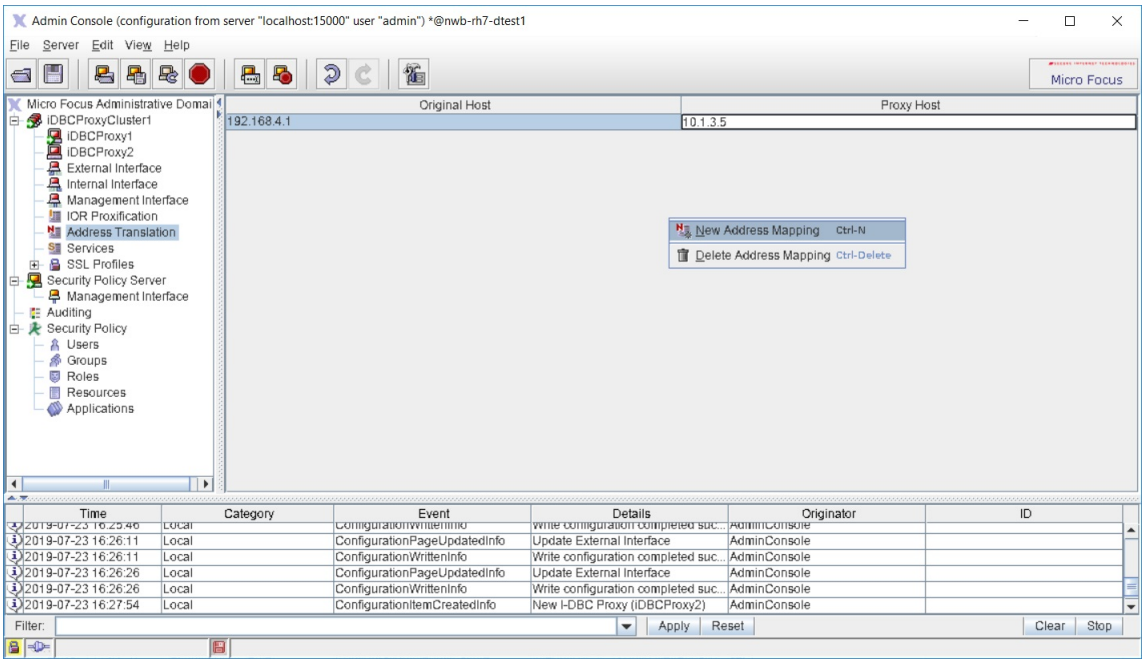


Figure 39 Configuring Address Mappings

Address Translation: Defining and Deleting Address Mappings

Note that if you do not 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.

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 Administration Console prompts you to delete the affected IORs. Correspondingly, if you delete an IOR from the Initial IOR table the Administration Console will prompt you to delete all unused address mappings.

To configure the address mapping table for the scenario depicted in Figure 38 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.

Do not forget to specify mappings for dynamically generated IORs, i.e., for every CORBA server behind the NAT router.

SSL Profiles

This chapter describes SSL Profiles. SSL Profiles define the keys and certificates for an SSL connection. 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”.

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

For convenience, when adding a new cluster to the configuration, the Administration 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”.

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.

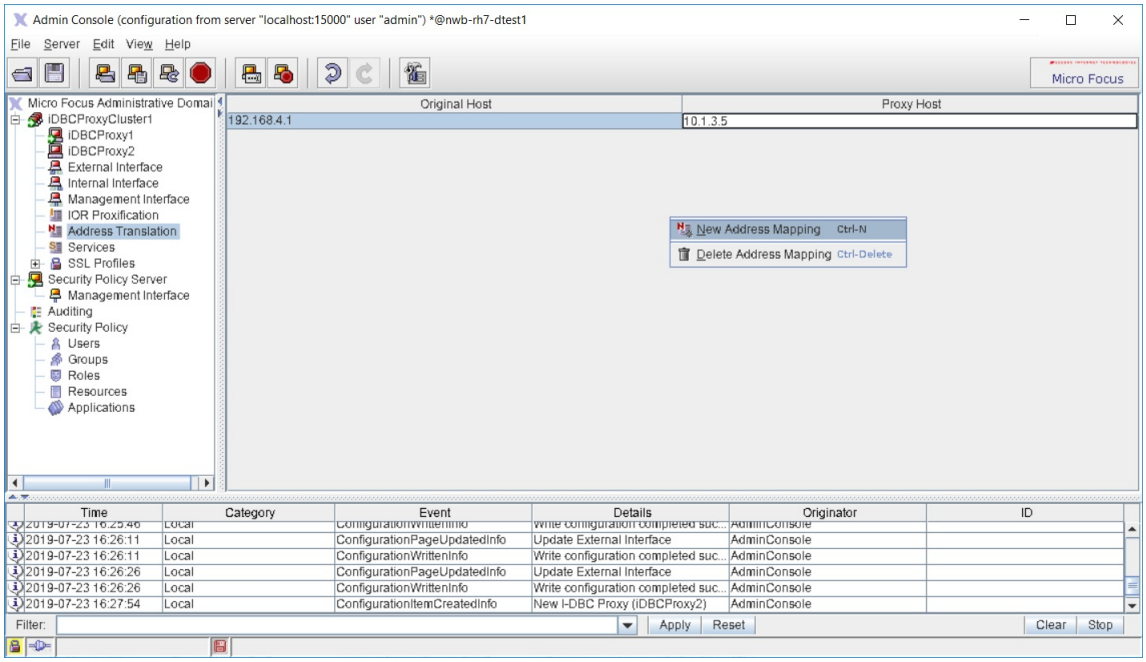


Figure 40 SSL Profile – Protocol

SSL Version

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

- **SSLv3 Only**: This will only permit SSL3 to be used. We do not advise using this as it is only for legacy deployments which cannot be upgraded to later versions.
- **SSLv3 and higher**: Support SSLv3, and also TLSv1.0, TLSv1.1,TLSv1.2 and TLSv1.3.
- **TLSv1.0 and higher**: Support TLSv1.0, TLSv1.1, TLSv1.2 and TLSv1.3. For the SSLClient Profile (which applies when the DBC is in the client role) it is best practice to use this option or one of the following options.
- **TLSv1.1 and higher**: Support TLSv1.1, TLSv1.2 and TLSv1.3.
- **TLSv1.2 and higher**: Support TLSv1.2 and TLSv1.3.
- **TLSv1.3 and higher**: Support TLSv1.3
- **V23/TLS**: This will allow all protocols. This option is not recommended as it can cause security vulnerabilities. It is included for legacy purposes.

It should be noted that in many of the cases the setting will allow the value that is being set and higher protocols. This means that the highest protocol, TLS 1.3, will always be enabled. Disabling TLS 1.3 is actually performed by updating the allowed Ciphersuites (see below).

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:TLSv1.3**: Allows TLSv1 ciphers and TLSv1.3 ciphers.
- **TLSv1.2:TLSv1.3**: Allows TLSV1.2 ciphers and TLSv1.3 ciphers.
- **TLSv1.3**: Allows TLSv1.3 ciphers.
- **TLSv1:!TLSv1.3**: Allows TLSv1 ciphers but not TLSv1.3 ciphers (use this to disable TLS 1.3)
- **TLSv1.2:TLSv1.3**: Allows TLSv1.2 ciphers but not TLSv1.3 ciphers (use this to disable TLS 1.3)
- **TSLv1**: Allows TLSv1 and above.
- **SSLv3**: Disallows SSLv2 ciphers.

The following values still exist for legacy purposes but we would recommend using one of the above values.

- **DEFAULT**: This corresponds to ALL:!aNULL:!eNULL, i.e. all available ciphers.
- **TLSv1:!EXPORT**: Like TLSv1 but disallows 40 bit ciphers.
- **TLSv1:!EXPORT:!aNULL:!eNULL**: Additionally disallows ciphers offering no authentication and ciphers offering 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).

The appendix "SSL Ciphers" 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.

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".

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.

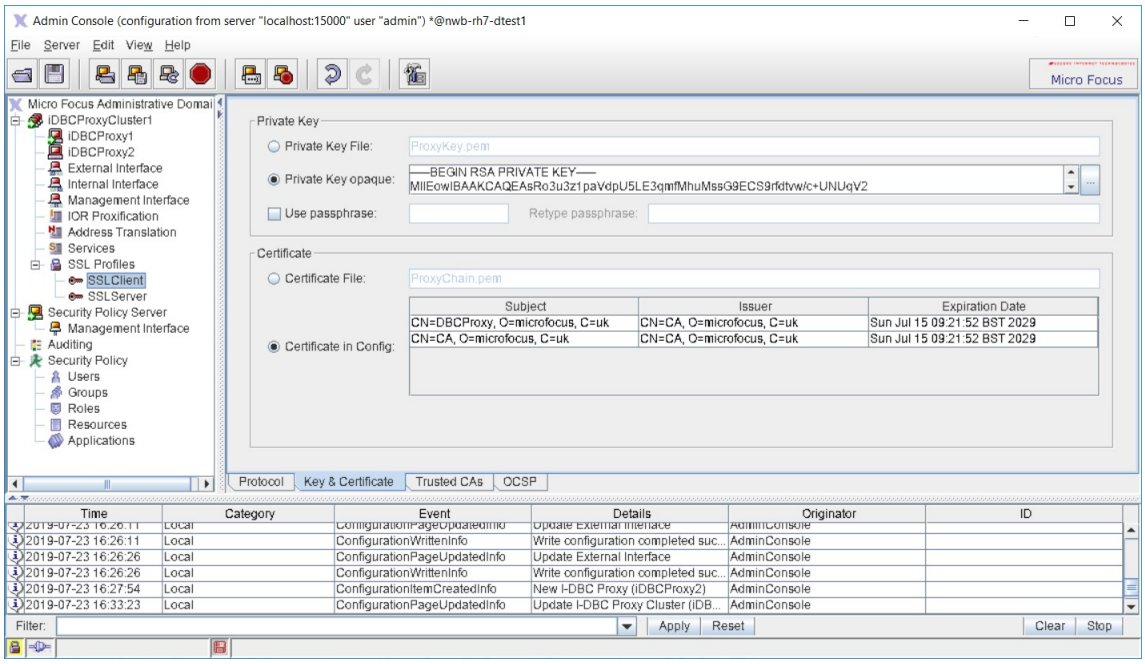


Figure 41 SSL Profile – Key & Certificate

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 on 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

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.

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).

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".

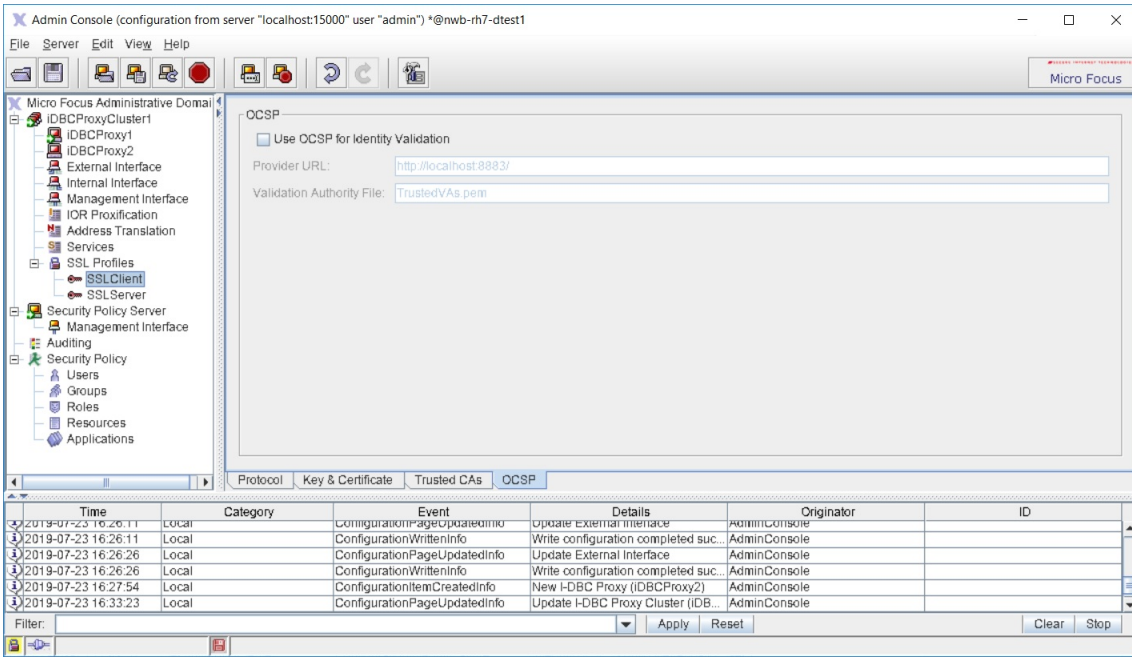


Figure 42 SSL Profile – OCSP

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).

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.

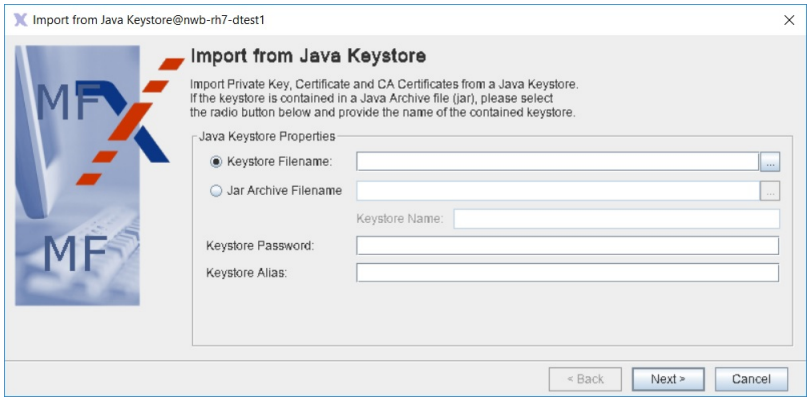


Figure 43 Import from Keystore: Java Keystore Properties

On the first panel (cf. Figure 43), 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.

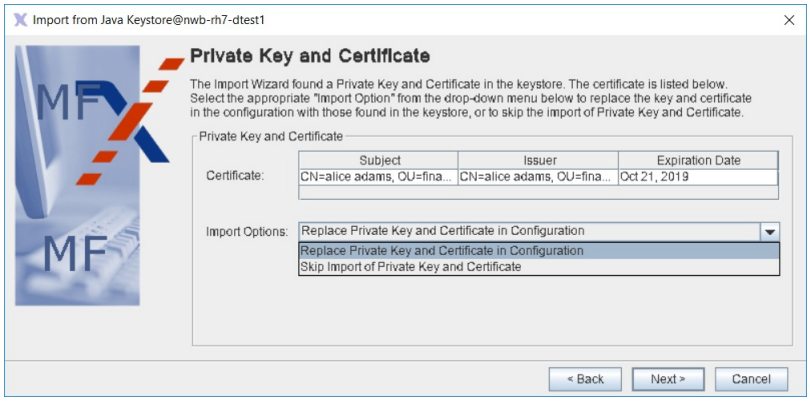


Figure 44 Import from Keystore: Private Key and Certificate

The next panel (cf. Figure 44) 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.

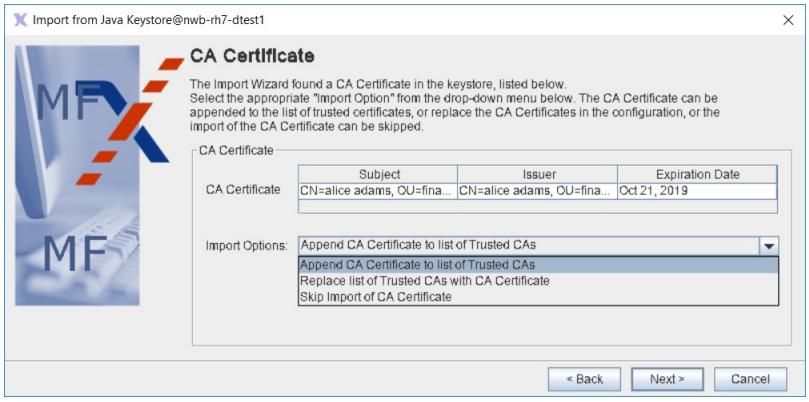


Figure 45 Import from Keystore: CA Certificate

The next panel (cf. Figure 45) 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.

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, Micro Focus's DBC for CORBA, and the WS-DBC is the Web Services Domain Boundary Controller, Micro Focus's DBC for Web Services. The appearance of the SPS configuration panel in the Administration Console is the same for both products.

Single SPS and SPS Cluster

The DBC architecture supports High Availability and Scalability (see "High Availability and Scalability" in 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 Administration 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.

Security Policy Server Cluster Properties

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

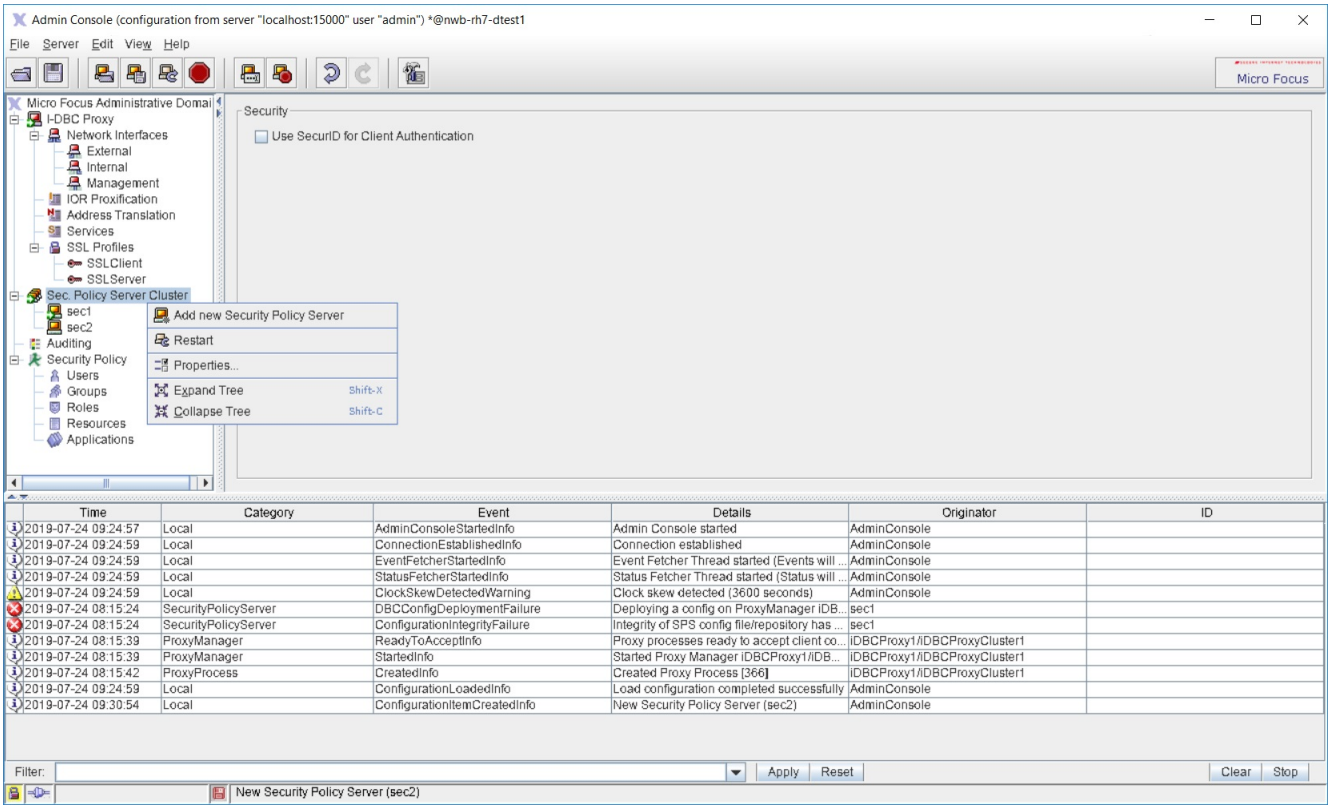


Figure 46 Security Policy Server Cluster

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

SecurID for Client Authentication only applies to the **I-DBC**.

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. If this box is checked the DBC authenticator on the SPS is activated (cf. "I-DBC Authentication" in 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.

Security Policy Server

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

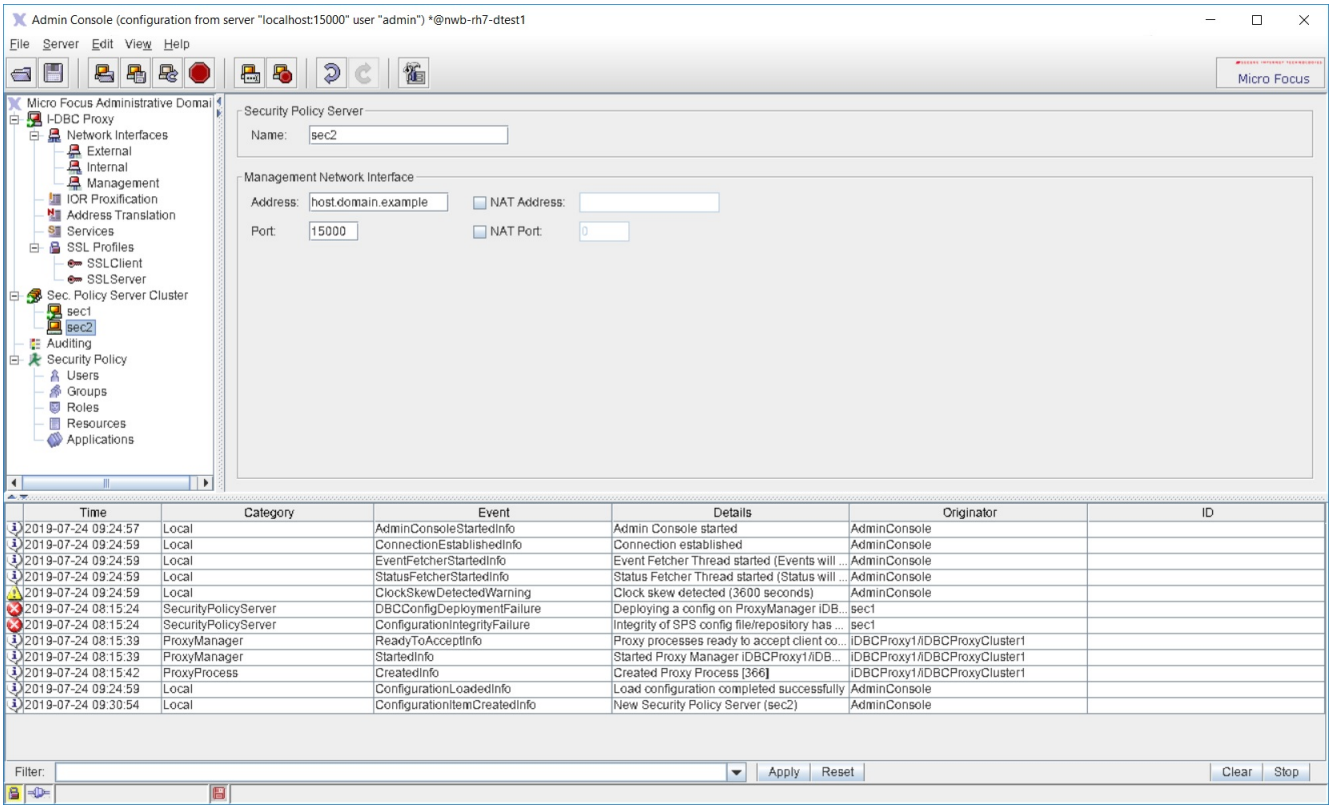


Figure 47 Security Policy Server Properties

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.

Security Policy Server Name

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.

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 Administration 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 Administration 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 Administration 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 Administration 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.

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 48 displays the situation.

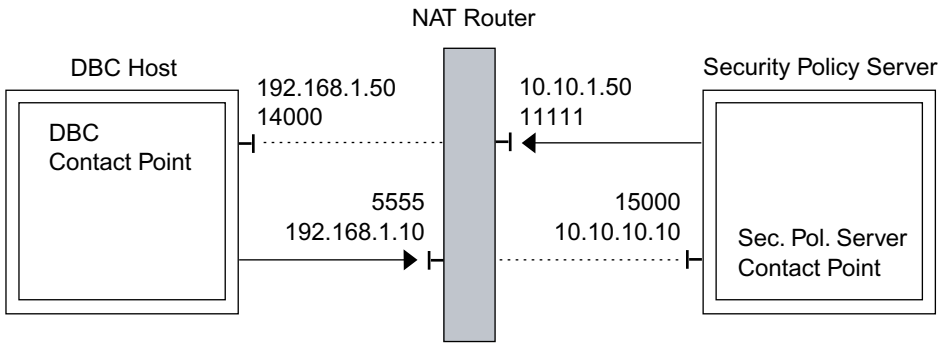


Figure 48 NAT Router between the DBC Proxy host and the SPS host

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.

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.

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 available separately.

Audit Events

Event notifications include a name denoting the event, a time stamp, and the originator. Events related to IIOP messages 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.

Event Flow

Events occur in different system components: in the Proxy, the Security Policy Server, and the Administration 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.

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.

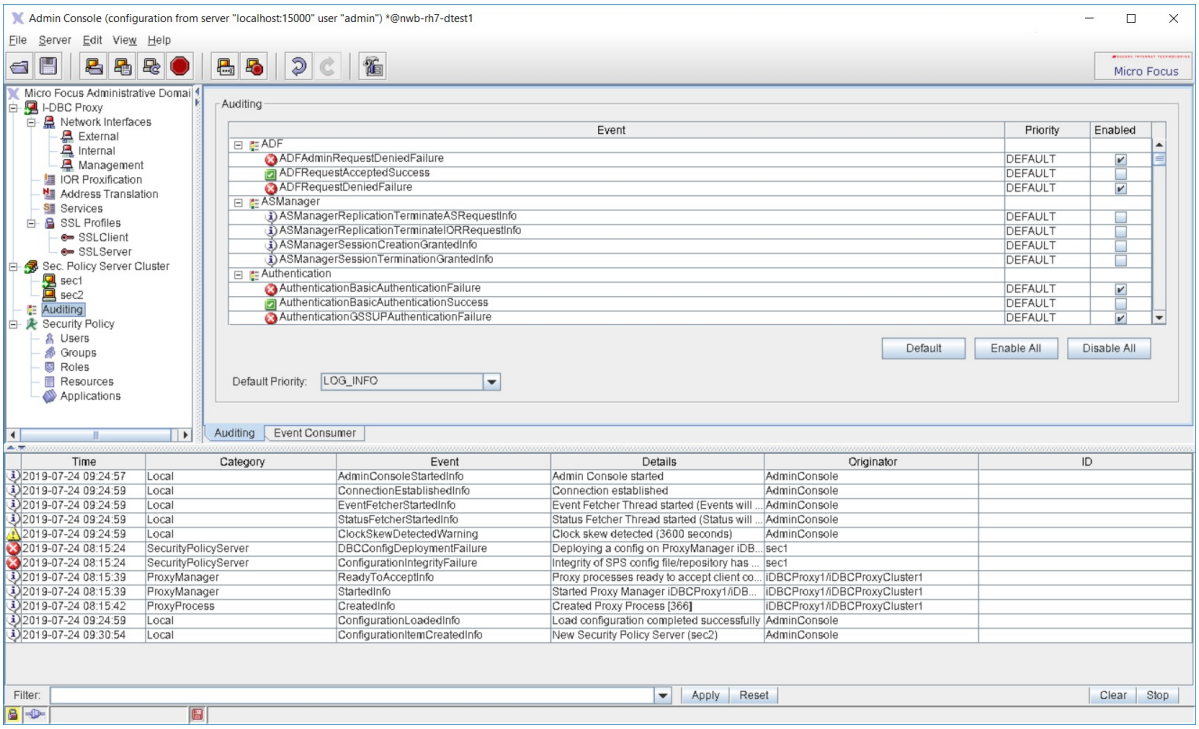


Figure 49 Auditing – Configuring Audit Events

Events are sorted by their category and displayed hierarchically. Click on the ⊞ next to the event category to see all events of the category. 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 the appendix “Audit Events”):

- Proxy Events: Operational status: started, resource limits reached, etc.
- Connection-related Events:
  - GIOP connection: 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.
- IIOP message processing:
  - IOR processing: new original IOR proxified/deproxified
  - Parameter Checking: success, failure

Event Consumer

On this tab you can define Event Consumer details.

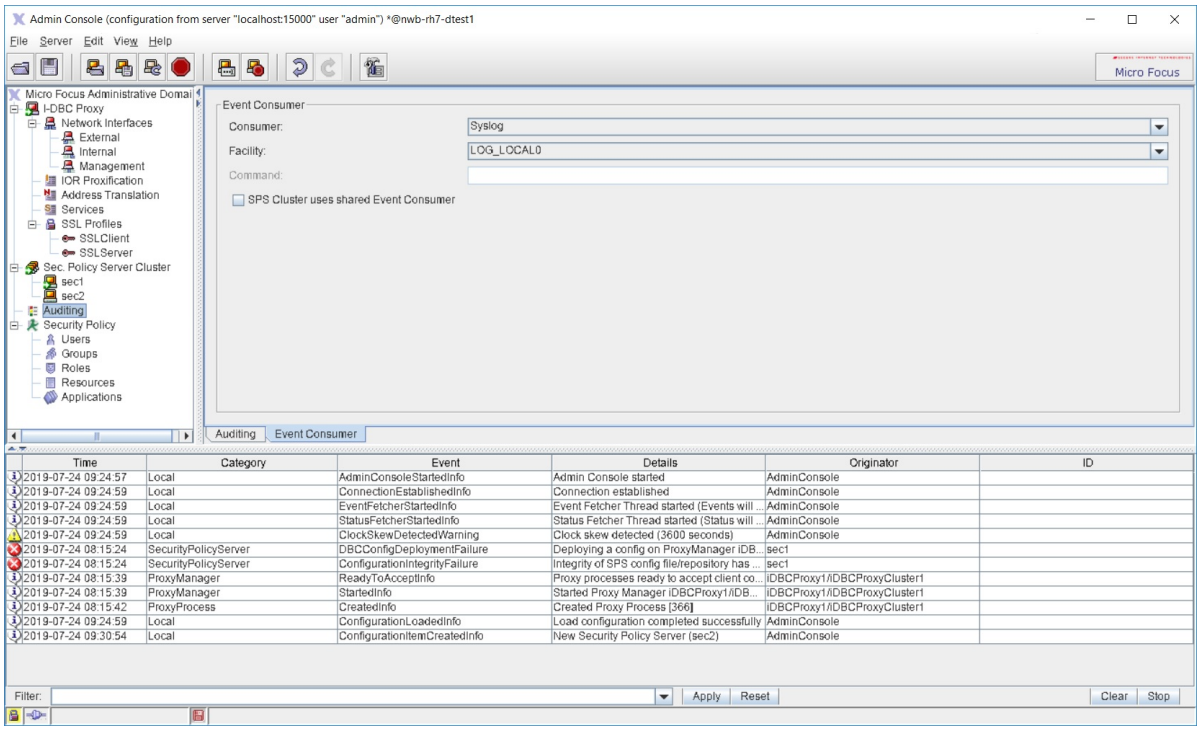


Figure 50 Auditing – Configuring the Event Consumer

external command

Currently audit events can be logged via `syslog` or an external command. The external command can be any UNIX command, for example:

- `cat >>/tmp/event.log`
- `/usr/bin/tee /tmp/events.log 1>&2`

`syslog`

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:

```
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).

shared event consumer

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.

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.

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.

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.

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.

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").
- 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.

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".

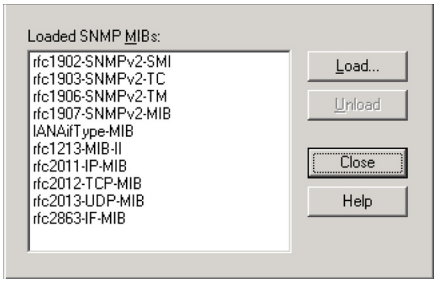


Figure 51 HP OpenView NNM "Load/Unload MIBs: SNMP" dialog

- The "Load/Unload MIBs: SNMP" dialog will appear (see Figure 51). 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.

Expert Mode

Not all configuration properties are accessible in the default appearance of the Administration 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.

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 Audit Policy is stored in the dictionary `logging`. 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. For more detailed information about configuration keys used in the DBC contact Micro Focus SupportLine (http://supportline.microfocus.com/).

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 conveniently edit the raw configuration data with the Administration Console in expert mode instead of using a text editor on the configuration file.

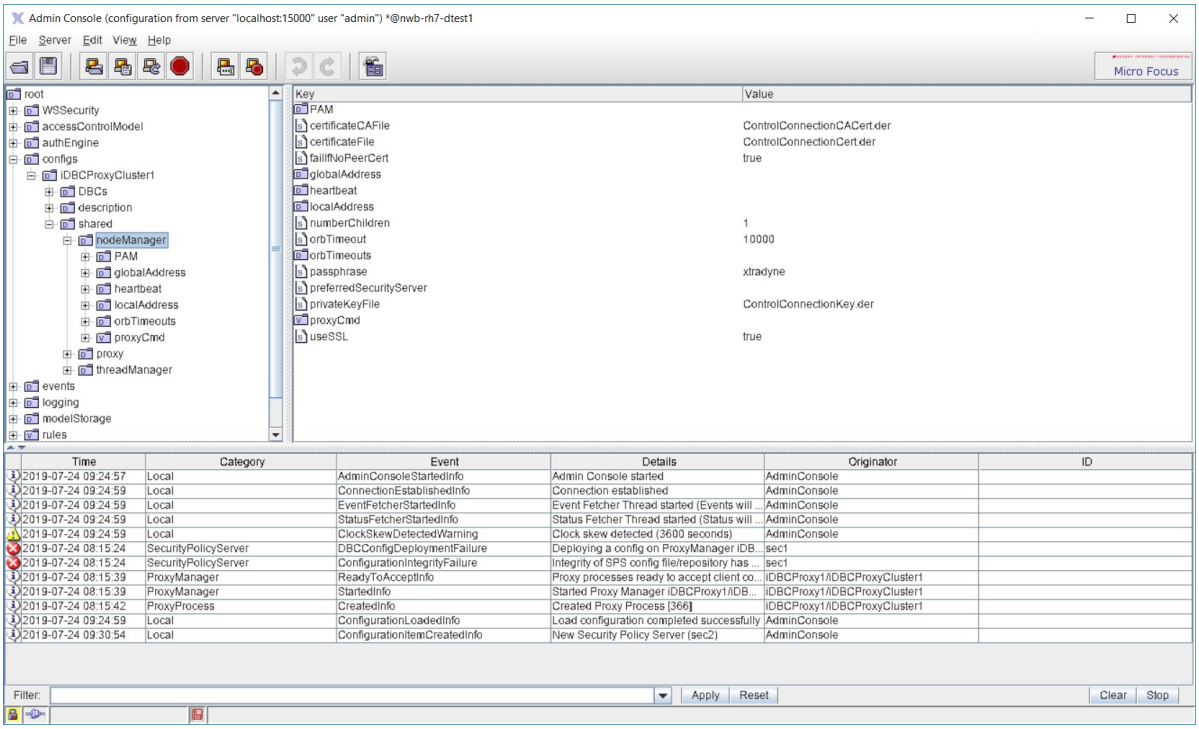


Figure 52 Expert Mode

Note that the direct manipulation of configuration data is only recommended for experienced users.

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 values, you have to verify yourself that they are correct.

When switching from the expert mode back to the standard mode the Administration Console checks the existence of some mandatory keys in the configuration. This guarantees the working of the Administration Console, but not the correct operation of a DBC installation with such a configuration.

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.

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.

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.

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 Administration 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 Administration 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.

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.

For a quick reference on how to install keys and certificates, please proceed to section "Replacing Keys and Certificates".

Trust Establishment

To establish mutual trust between DBC components (the DBC Proxy, the Security Policy Server, and the Administration 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.

Certificates and Certification Authorities

For security reasons, the DBC has two different kinds of communication links:

- 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 Administration Console. Keys and certificates used on these connections are signed by another Certification Authority called *ControlConnectionCA*.

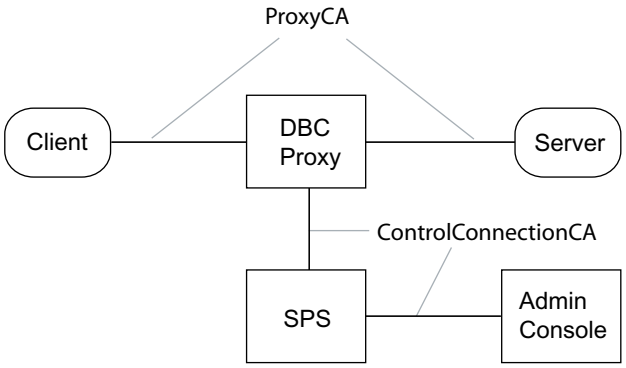


Figure 53 Certification Authorities in the DBC installation

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 Administration 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 1, "Default Keys and Certificates for application connections" Table 2, "Default Keys and Certificates for Control Connection".

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 Administration 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 Administration Console.

Trust Store for the Control Connections

For the control connections (between the DBC Proxy, the SPS, and the Administration 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 connections, the file `ControlConnectionCACert.der` has to be replaced by a new CA certificate in DER encoding.

Application Connections

Clients 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 clients for valid 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 54).

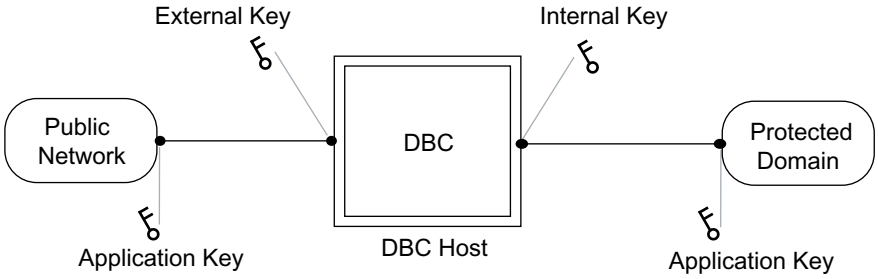


Figure 54 Keys used on application connections

These keys are signed by the *ProxyCA* (cf. "Trust Stores and Trusted CAs").

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. "SSL Profiles") 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 Administration Console (see "CA Certificates (Trusted CAs)").

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.

Table 1 Default Keys and Certificates for application connections

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 Administration Console)	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

All these keys are passphrase-protected with the passphrase `blahblah`.

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 Administration Console and is explained in detail in "Replacing External and Internal Proxy Keys".

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 certificates). By default, the DBC uses different trust stores for different kinds of SSL communication:

- The trust store for application connections is kept in the configuration and editable with the Administration Console (on how to do this, please refer to section "CA Certificates (Trusted CAs)"). Alternatively, it is located in a file in the directory `sps/adm` (filename configurable with the Administration Console).
- `sps/adm/TrustedLDAPCAs.pem`: Trust store for LDAP/SSL Profiles.
- `sps/adm/TrustedVAs.pem`: Trust store for validation authority files (for OCSP).

You can append a certificate to any of these trust stores with a text editor or by typing, for example:

```
cat myCACert.pem >> TrustedCAs.pem
```

If the certificate you want to add is not in PEM format, see the section "Changing the Certificate Encoding Format".

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).

Integrating the DBC with Applications

To allow external software to authenticate to 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 directory 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)".

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`

Control and Administration Connections

The DBC relies on 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 Administration Console is called Administration Connection (see).

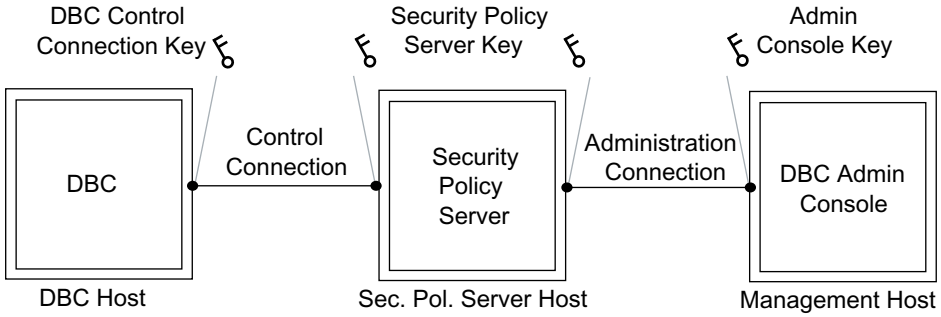


Figure 55 Keys used on the Control and Administration Connection

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 DBC Proxy uses the DBC Control Connection Key. The Administration Console uses the Administration 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").

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

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

Key	File name	Host
Public key certificate of the Administration Console	`bin/AdminConsoleCert.der`	Admin Host
Private key of the Administration Console	`bin/AdminConsoleKey.der`	Admin Host

Table 2 Default Keys and Certificates for Control Connection

All these keys are passphrase-protected with the passphrase `blahblah`.

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 Administration 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/ControlConnectionCert.der
```

The certificate will be printed in a readable form including the validity dates that look something like this:

```
Validity
   Not Before: Apr 10 10:40:32 2019 GMT
   Not After : Apr  7 10:40:32 2029 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.

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.

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 Administration Console

The Internal and External Proxy Keys and trusted certificates can be configured with the Administration Console. Both kinds of keys are selected using *SSL Profiles,* which were explained in detail in chapter "SSL Profiles".

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 Administration 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

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 54:

- 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, see "Checking Permissions for Key Files".

To make the DBC Proxy recognize the new keys, start the Administration 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 (and default) setting is "TLSv1.0 or higher". For more details, see "SSL Version".

- *Ciphersuite*: Choose the set of ciphers to be used. The recommended (and default) setting is "TLSv1" For details see "Ciphersuite".
- *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*: This method is 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). It is vital 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.

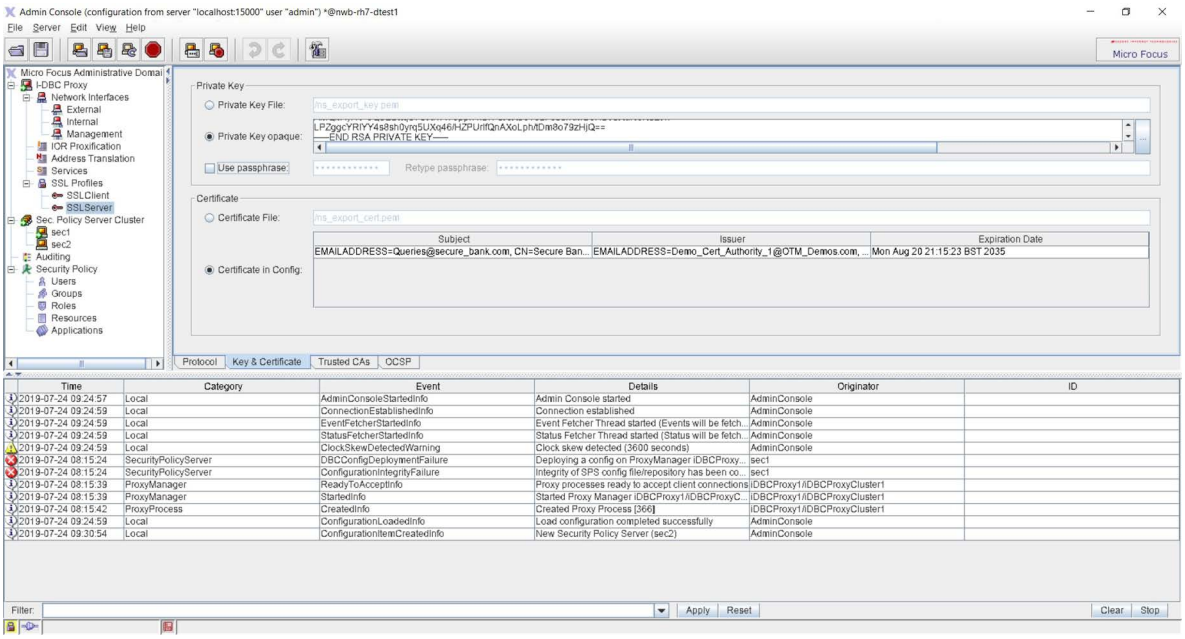


Figure 56 Configuring an SSL Profile

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 Administration 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 Administration 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.

Checking Permissions for Key Files

You should check if permissions for keys are set correctly. Only the DBC installation user (`corba` 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:/opt/microfocus/idbc/adm > ls -la *.der
-rw------ 1 corba corba 592 Jul 9 17:07 ControlConnectionCACert.der
-rw------ 1 corba corba 654 Jul 9 17:07 ProxyCert.der
-rw------ 1 corba corba 677 Jul 9 17:07 ProxyKey.de
```

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 set as above, use

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

Keys must be owned by the DBC installation user (`corba` by default). If this is not the case type, for example (as `root`):

```
chown corba *.der *.pem
```

Replacing Control and Administration Connection Keys

It is not necessary to replace the Control Connection and the Administration Client Key, which is used by the DBC Proxy and the Administration Console respectively, to establish trust with the Security Policy Server. These keys are generated at install time.

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)".

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 the "Preface" for contact information).

Encrypting the DBC Configuration File for Support

When you are asked to send the DBC configuration file to the Micro Focus support team you may use the **File > Export > Encrypt for support...** facility of the Administration 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 Micro Focus it may be re-imported with the Administration Console using the **File > Import > Decrypt from support...** function.

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. 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". DBC audit events are listed in the appendix "Audit Events". Events can be viewed with the Administration Console's event browser, see "Audit Event Browser".
- **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 into the file `dbc.log` on the DBC Proxy. A list of DBC error messages can be found in "DBC and SPS Error Messages".
  - CORBA system exceptions: A list of CORBA system exceptions and minor codes received by service clients can be found in "CORBA System Exceptions and Minor Codes".

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 and the Security Policy Server are up and running. Their status can be determined with the scripts `xdn_sps` and `xdn_idbc` with the option `status` (you must be root to execute this):

```
/etc/init.d/xdn_sps status
/etc/init.d/xdn_idbc status
```

The scripts output "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 display "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 `<INSTALLDIR>/idbc/adm/dbc.log` on the DBC Proxy host).

Useful Status Scripts

For more information about the status of the SPS and the I-DBC Proxy,respectively such as 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>/sps/bin/checksps.sh
```

Example

```
~:/opt/microfocus/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 the 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 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.

Scripts Do Not Work

Some scripts included in the DBC installation such as `proxyconfig.sh`, `checkproxy.sh` and `checksps.sh` try to switch to the user `corba` before actually running. This will only work if the home directory of this user exists. At install time the user `corba` is created if it doesn't exist yet. The home directory of this user is the installation directory (defaults to `/usr/corba/` on Linux). When uninstalling the user `corba` will not be deleted. If you intend to upgrade to a newer DBC version and do not want to use the default directory for installation, please make sure to delete the user `corba` before re-installing. Otherwise the user `corba` will have the wrong (possibly non existing) home directory.

Checking Permissions for Key Files

After installing your own keys you should check if permissions for keys are set correctly Only the DBC installation user (`corba` 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:/opt/microfocus/idbc/adm > ls -la *.der
-rw------ 1 corba corba 592 Jul 9 17:07 ControlConnectionCACert.der
-rw------ 1 corba corba 654 Jul 9 17:07 ProxyCert.der
-rw------ 1 corba corba 677 Jul 9 17:07 ProxyKey.de
```

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 set as above, use

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

Keys must be owned by the DBC installation user (`corba` by default). If this is not the case type, for example (as `root`):

```
chown corba *.der *.pem
```

When to Restart the DBC Proxy / SPS

If changes to the configuration you made with the Administration Console do not seem to take effect, the DBC Proxy or the Security Policy Server may have to be restarted. Note that the Administration Console will generally indicate when a restart is required. For a summary of when a restart is required, please refer also to “When to Restart the DBC Proxy / Security Policy Server”.

Frequently Encountered Problems

Access Denied

If client access to a resource is denied the client will receive the CORBA System Exception “NO_PERMISSION”. 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 “ADFRequestDeniedFailure”).

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”. The following sections list the most common events indicating problems with SSL and what can be done about them. Please refer also to “Audit Events”.

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 Administration Console, go to the “Security Policy – Users” panel and check that the user is included in the list (see section “User Properties – General” 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 Administration 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" 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" 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, for example:

```
printcert.sh ../adm/>
```

Alternatively, you can use a tool like `openssl`, for example:

```
openssl x509 -in <certificate_file> -inform DER -text
```

Callback Configuration

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. 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. 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.

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 I-DBC Proxy. Copy the license file to the directory `<INSTALLDIR>/idbc/adm/license.txt`.
- 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.

Administration Console

Startup

Before starting the Administration 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
```

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 (see "Checking the Validity of Keys and Certificates"). If you installed your own Administration Console keys and certificates, please check whether you configured everything as described in chapter "Installing Keys and Certificates". Switch off authentication and try to connect with SSL

encryption only (Configure this on the Administration 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).

- Management port has changed:

  Changing the SPS Management Network Interface Port to, e.g., 16005 enforces a restart of the SPS. Subsequently the Administration Console will try an automatic relogin. This may fail when the host configured in the login panel differs from the host configured on the SPS management panel (e.g. when a name is given in the login panel and an IP-address on the SPS Management Interface panel). Please make sure that the SPS address given on the login panel is exactly the same as the one configured on the SPS Management Interface panel.

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 Administration Console share one listener to contact the Security Policy Server. There are two ways to fix this error:

- turn off SSL (this is not recommended)
- reconfigure the Security Policy Server and the DBC Proxy so that they use SSL for their control connection.

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".

Miscellaneous

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 Administration 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 Administration 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
```

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`

Replace the first line:

```
#!/bin/sh
```

with

```
#!/usr/bin/bash --posix
```

My CORBA Application Doesn't Run With the 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 Administration 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 in "The Application Doesn't React At All").

No Listener Available for IOR Proxification

When importing an IOR with the Administration 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 Administration 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 Administration 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 a typical example application would look like this:

```
:~/Sample; printIOR server.prox.ref
Read from file server.prox.ior:
IOR:
Byte order: (1) Little Endian
TypeId: IDL:ItemManagement/Storage:1.0
Number of Tagged Profiles: 3
[1]     IIOP Profile (0x0)
  Byte order: (1) Little Endian
  Version: 1.2
  Host: 192.168.2.130
  Port: 8884
  ObjectKey:
    48 byte(s)
    49 74 65 6D 4D 61 6E 61 67 65 6D 65 6E 74 49 6D
       ItemManagementIm
    70 6C 2F 49 74 65 6D 4D 61 6E 61 67 65 6D 65 6E pl
       /ItemManagemen
    74 50 4F 41 2F 53 74 6F 72 61 67 65 49 6D 70 6C tPOA
       /StorageImpl
  Number of Tagged Components: 3
  [1]   SSL_SEC_TRANS Component (componentId=0x14)
    Byte order: (1) Little Endian
    Target supports: 127 - NoProtection & Integrity &
Confidentiality & DetectReplay & DetectMisordering &
EstablishTrustInTarget & EstablishTrustInClient
    Target requires: 0 - NoRestriction
    Port: 8885
```

Part III

Managing Security Policies and Protecting Applications

In this part

The I-DBC respects three different kinds of security policies: authentication policies, protection policies and access control policies. The first chapter in this part explains how security policies are defined using the Administration Console, with special emphasis on access control concepts. The second chapter describes the rules for defining parameter filters for the I-DBC

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 Administration Console to define security policies. Additional details are explained along with an example in the following chapters.

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.

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 57 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 57). 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 57) 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.

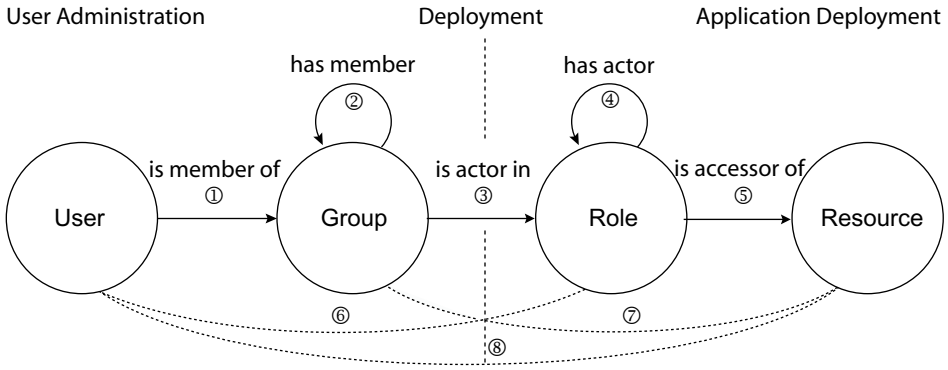


Figure 57 Components of the access control policy

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 Administration 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 57), but we recommend that they are used only sparingly!

Defining Security Policies

The following sections describe how to use the Administration 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 Administration 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 Administration Console to browse the policy.

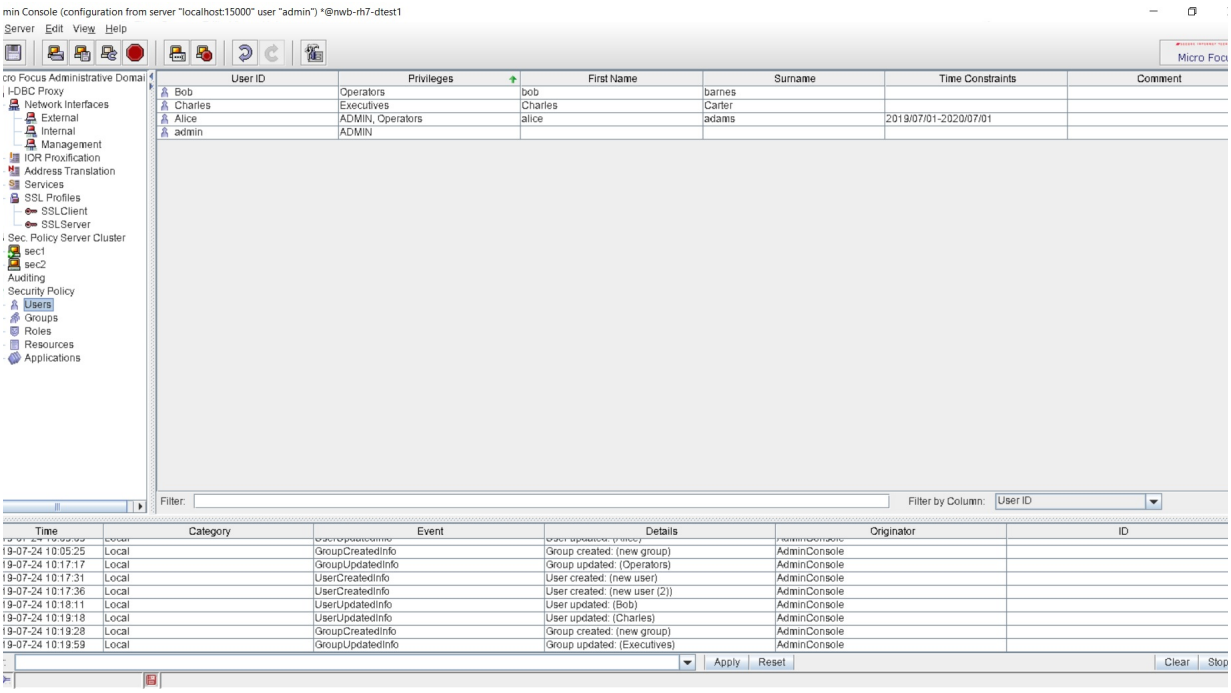


Figure 58 Screenshot of the Administration Console – Example User List

For I-DBC security policies the Administration 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”.

Security Policy

The “Security Policy” panel defines where the Security Policy Server stores security policies, and the properties of the chosen storage.

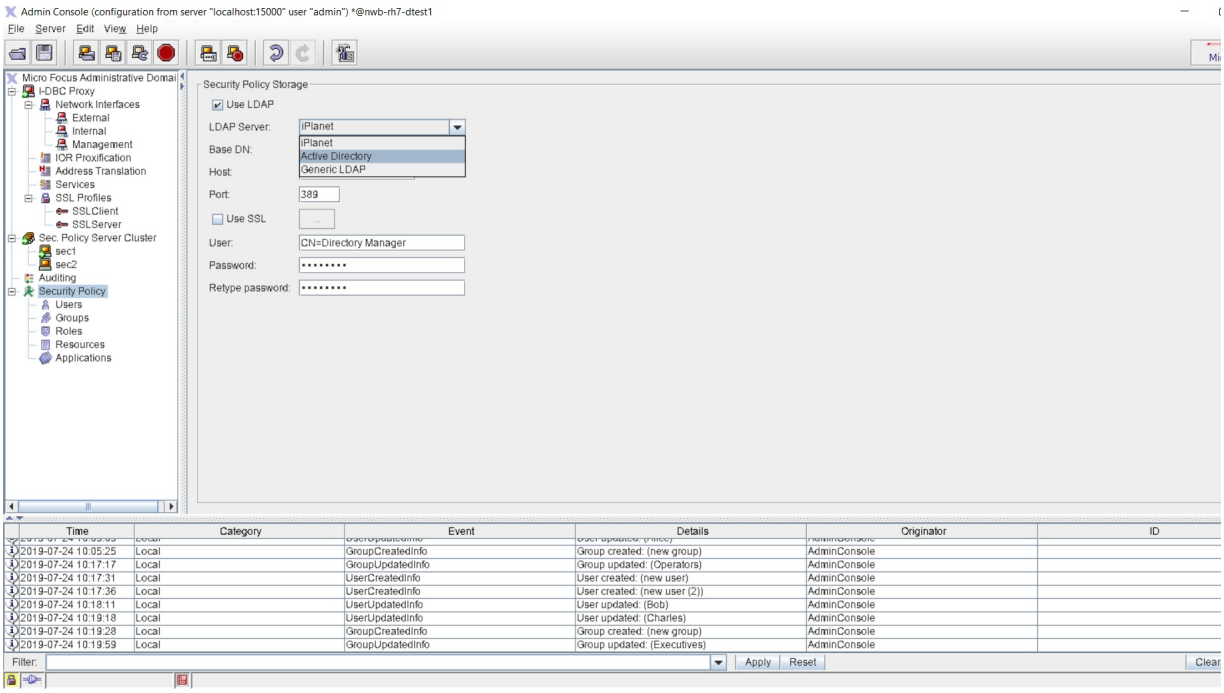


Figure 59 Security Policy: General Properties

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.

LDAP Server: Prerequisites

Before configuring the LDAP Server’s properties with the Administration Console, there are some configuration steps that need to be carried out.

DBC Base DN

obtain a node in your organization’s LDAP structure

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.

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 Administration 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.

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 Profiles"). 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.

General Navigation

In addition to the general "Security Policy" panel, the Administration 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.

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: "=", ".", ":", " ".

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 tabs (see Figure 60): The General tab defines the personal data of a user. The Authentication Methods tab defines by which means the user must authenticate to the system. The Privileges tab defines the user's relationship with other entities, for example the user's memberships in certain groups. The Constraints tab defines constraints for this user, for example an activation and expiration date.

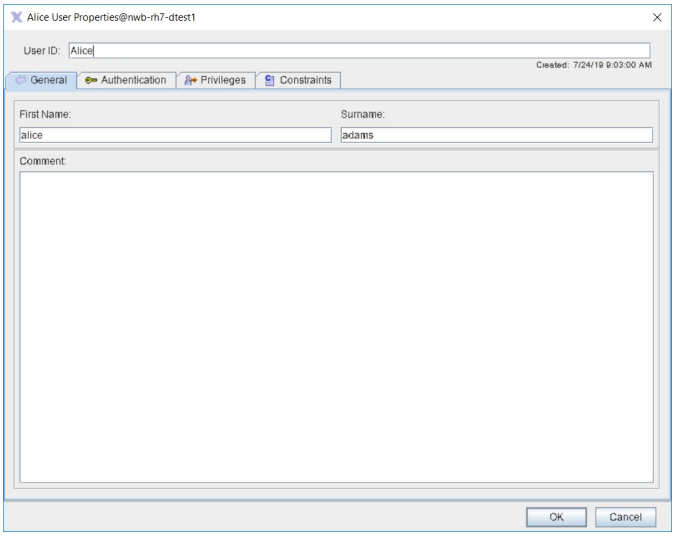


Figure 60 General User Properties

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.

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 authentication, SSL authentication, and – when operating a WS-DBC – authentication via XML Digital Signatures and SAML assertions. On the other hand resources can be configured to require a certain authentication level (e.g. authentication via SSL) to allow an access request, see "Resource Properties – Authentication".

To configure the 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 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".

Authentication via SAML Assertion

This authentication method is only available when operating a **WS-DBC** Proxy.

The I-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 I-DBC simply extracts and verifies the SAML assertion. The I-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 "SSL Profiles").

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.

X.509 Certificate Authentication

X.509 is a standard for digital certificates used for example by SSL. Certificates include among other things information about the identity of the certificate holder (i.e. the user). When choosing this authentication method the user, i.e., the client application, has to authenticate by presenting a certificate. Usually, this implies using SSL – or in case of the WS-DBC XML Digital Signatures. Using SSL is recommended anyway because it provides integrity and confidentiality protection for messages in transit.

On the right side of the panel 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).

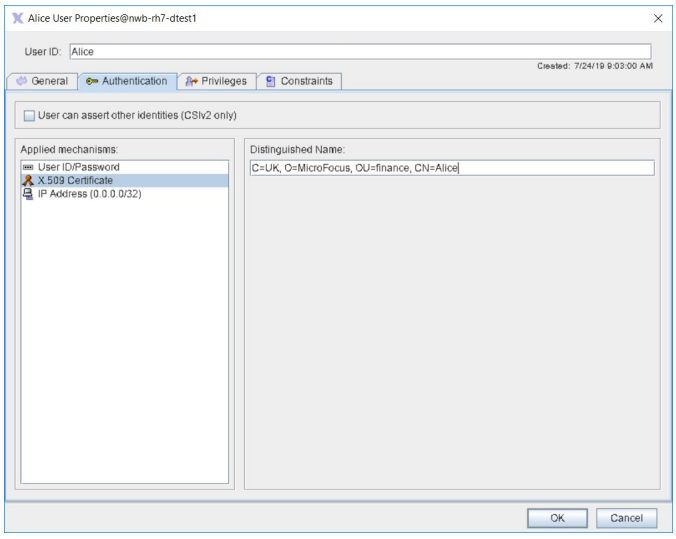


Figure 61 Authentication Methods: X.509 Certificate authentication

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

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 – Authentication”).

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 a 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 Administration Console.

Also note that the number of asterisks displayed in the Administration 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.

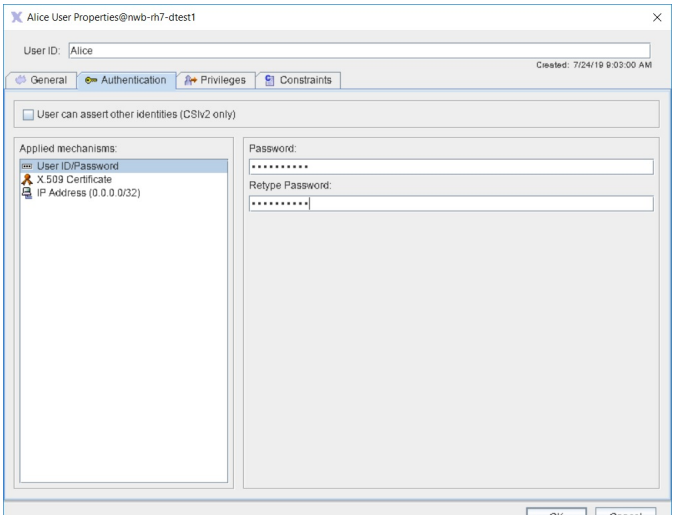


Figure 62 Authentication Methods: User ID / Password Authentication

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 135). 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.

Note:
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.

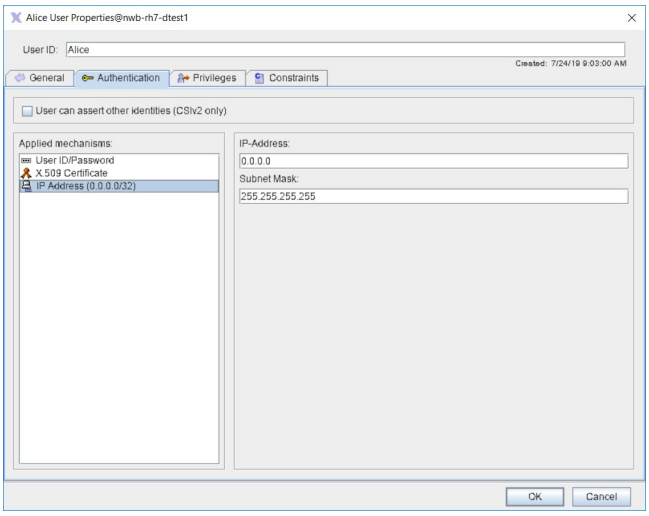


Figure 63 Authentication Methods: IP Address

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 the table below.

Table 3 policy IP address and subnet matching

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

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.

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 64).

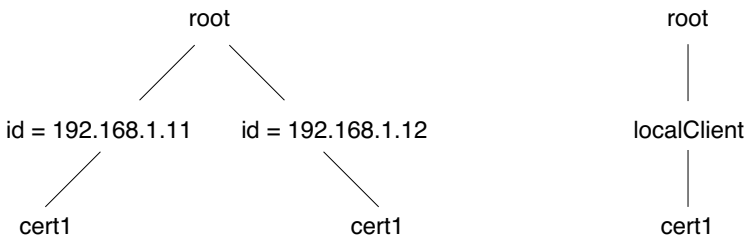


Figure 64 Access Session Hierarchy

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 64).

Example: One Access Session For Multiple Clients

You can use the Administration 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).

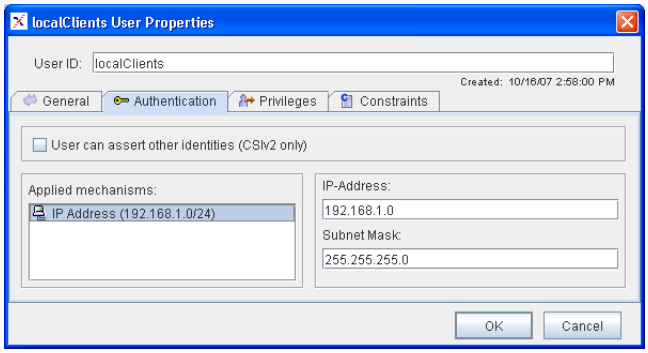


Figure 65 Configuring One Access Session for Multiple Clients

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 64).

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).

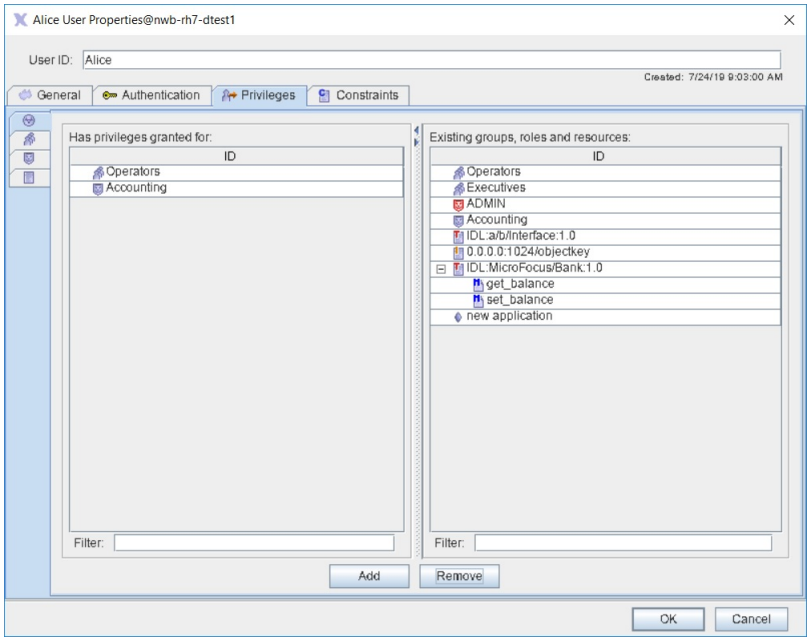


Figure 66 User Properties: Privileges

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 tab). 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 displayed 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 tabs because groups can be members of other groups, roles or resources. Additionally, roles can be members of other roles, and accessors of resources.

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 Administration 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 Administration 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.

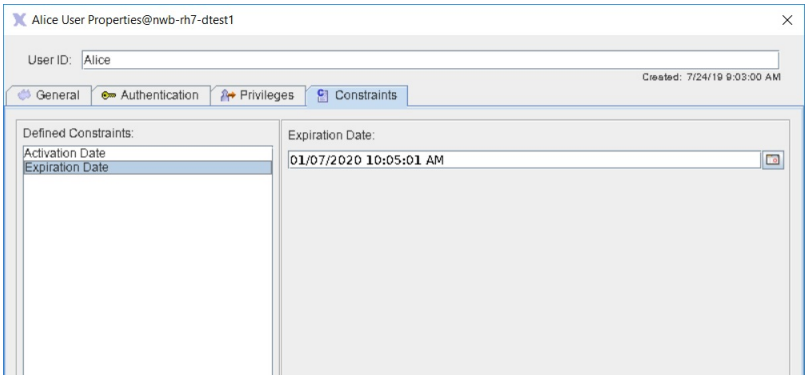


Figure 67 Define Constraints

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 screenshot below). You can choose the month and year at the top and the day which adapts to the chosen month/year below.

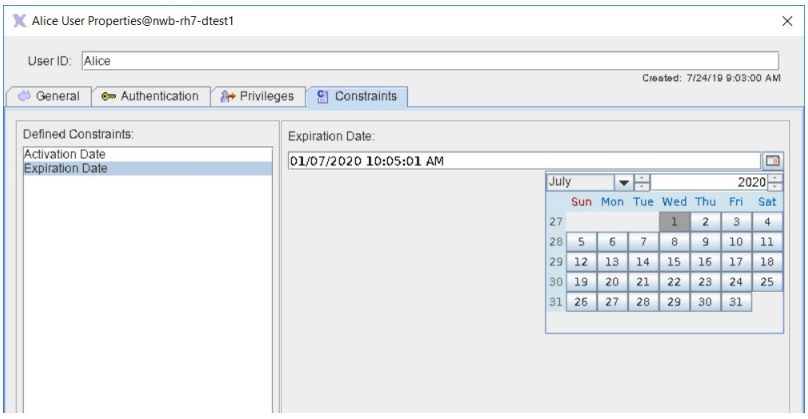


Figure 68 Choose a Date

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 68).

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.

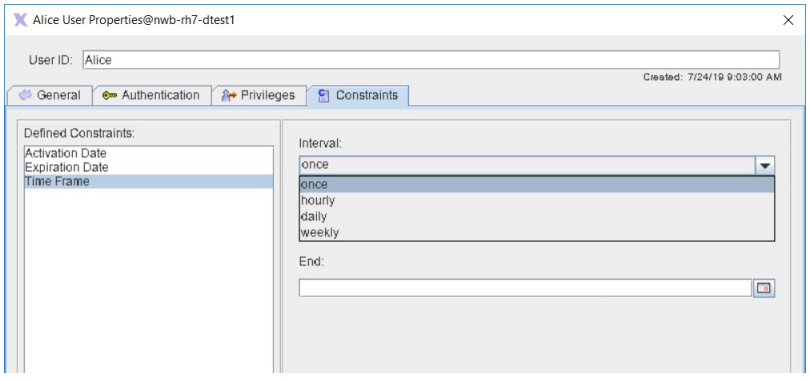


Figure 69 Define a Time Frame Constraint

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.

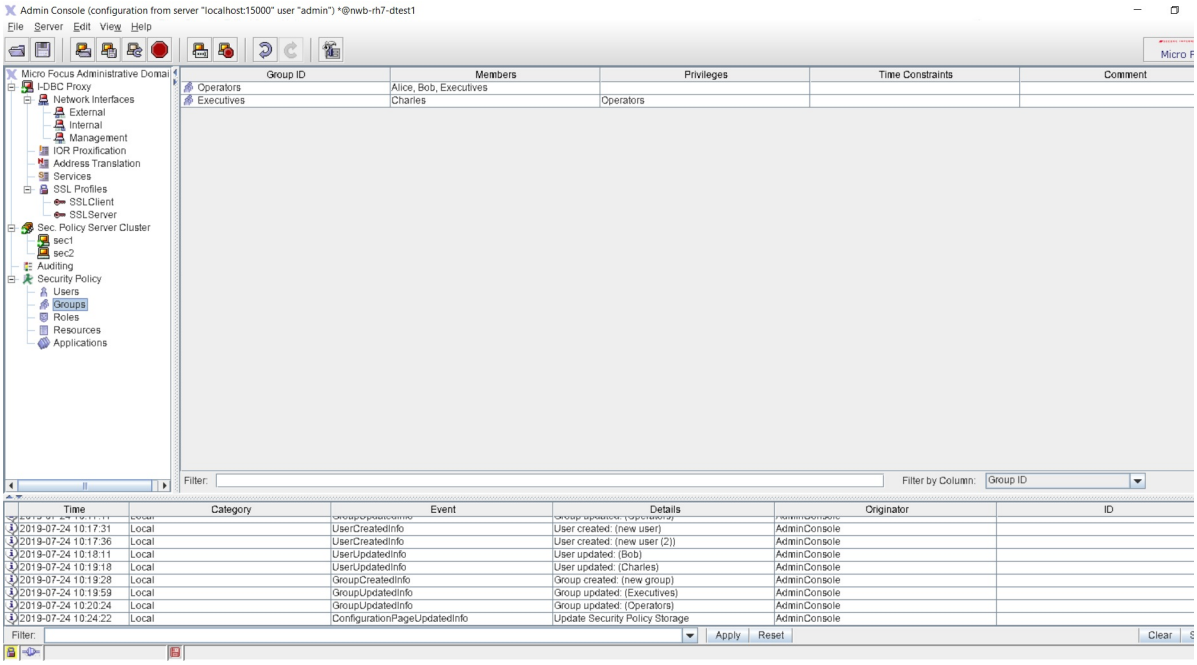


Figure 70 Administration Console – Group List

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 127.

Group Properties – General

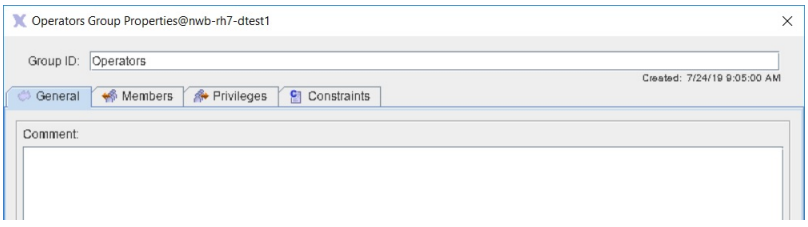


Figure 71 General Group Properties: General

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.

Group Properties – Members

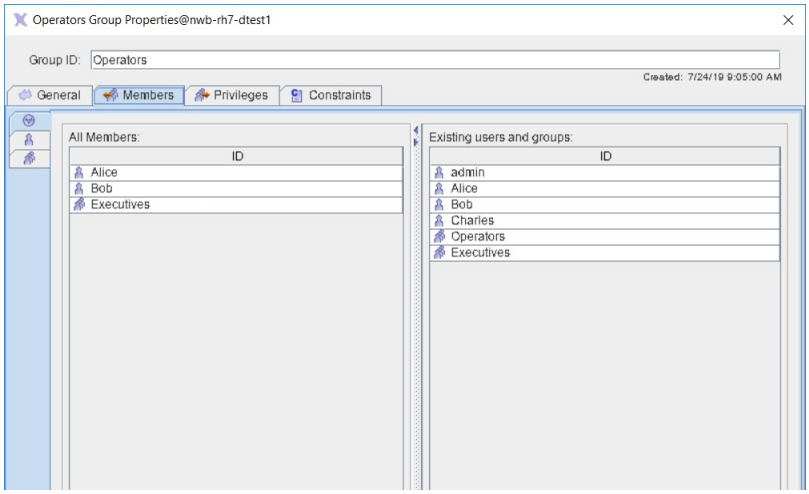


Figure 72 Group Properties: Members

The Members tab 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 tab, and *can be member* elsewhere (of other groups, roles, and resources) which is shown on the Privileges tab.

Group Properties – Privileges

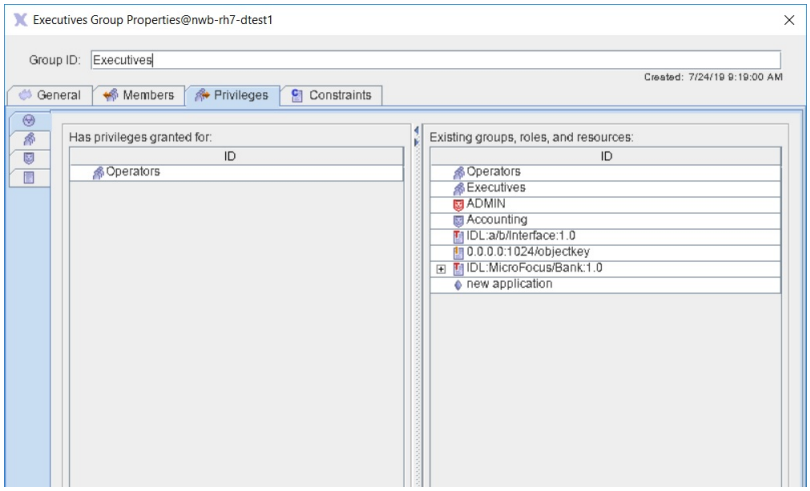


Figure 73 Group Properties: Privileges

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 tab 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.

Roles

From a configuration point of view, roles look much like groups:

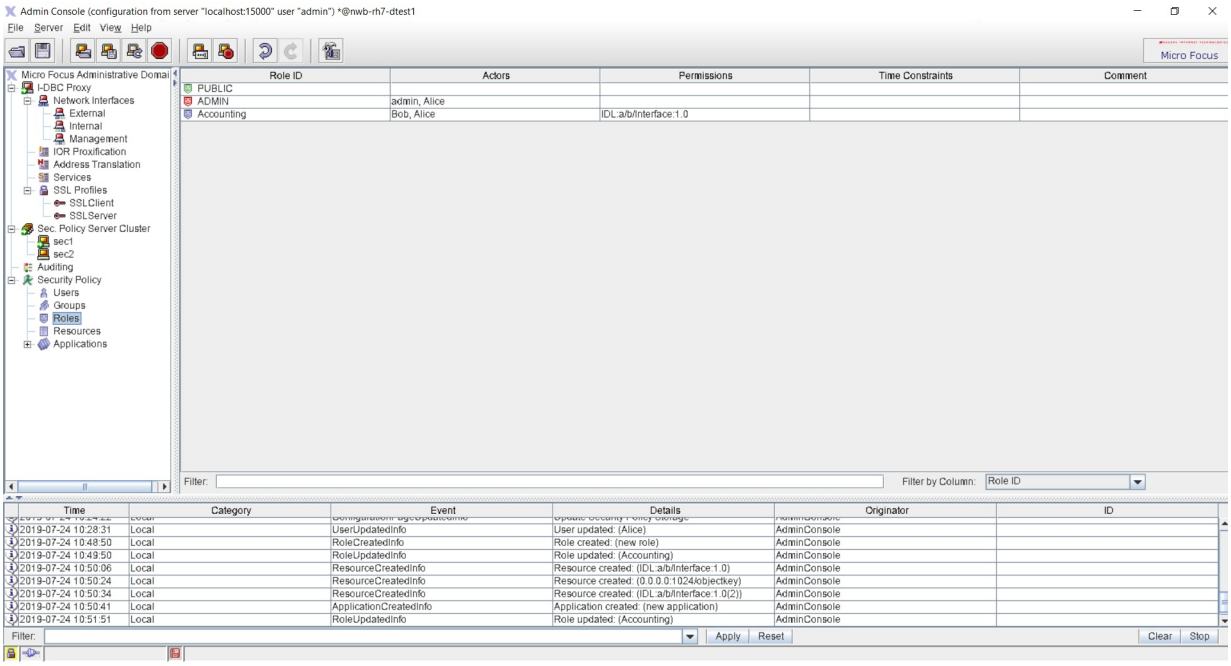


Figure 74 Administration Console – Role List

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: the PUBLIC Role"). The "ADMIN" role is designed for granting permissions for administrative tasks like starting or stopping DBC Proxies (for details see "Role Properties – Administration").

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".

Role Properties – General

You can configure roles similarly to groups, specifying a role ID and a comment.

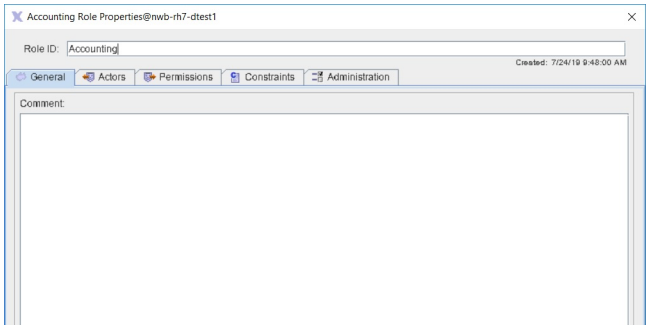


Figure 75 General Role Properties: General

Role Properties – Actors

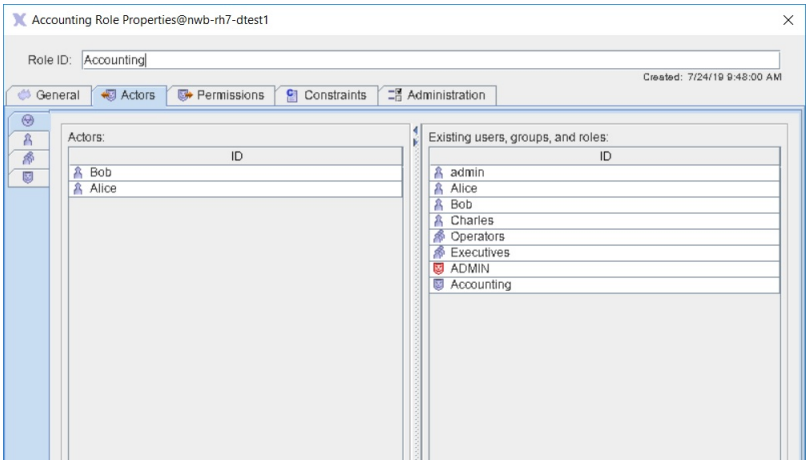


Figure 76 Role Properties: Actors

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 57).

Role Properties – Permissions

The role's Permissions tab is similar to the user's and group's privileges tab. Roles receive permissions from resources or other roles (see screenshot below).

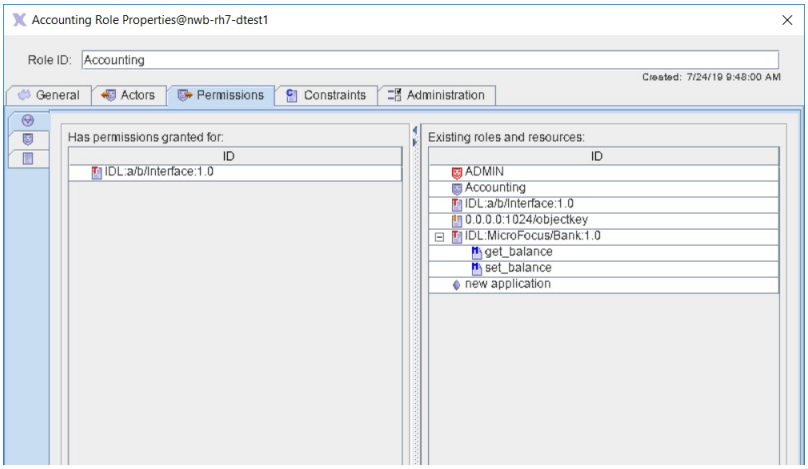


Figure 77 Role Properties: Permissions

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.

Role Properties – Administration

The "Administration" tab 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.

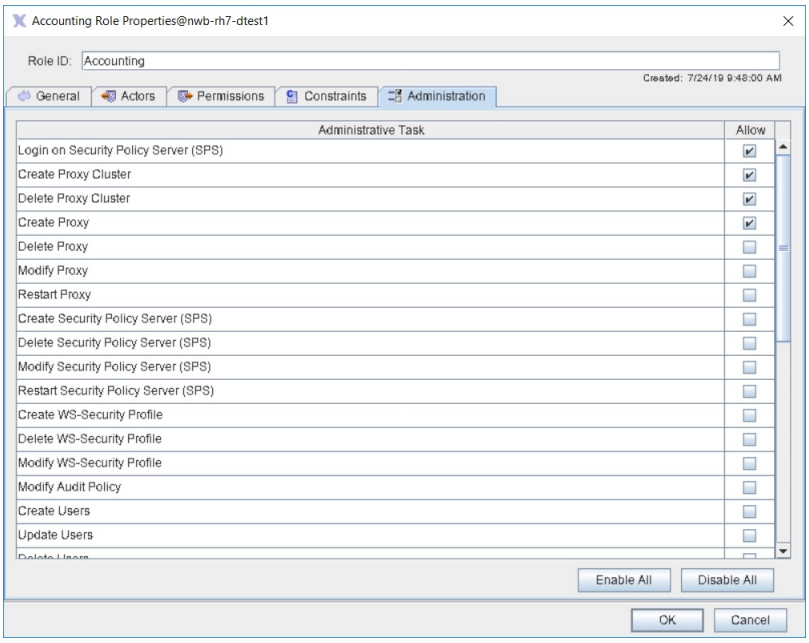


Figure 78 ADMIN Role – Granted administration rights

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).

Resources

Resources point to target services. These services are CORBA Services specified by their IOR[1].Clients only target a virtual resource in the DBC Proxy, not the actual target which is protected by the DBC. The original target IORs are only used in the access control policy. The actual target is defined by the proxified IOR; see “IOR Proxification” for details.

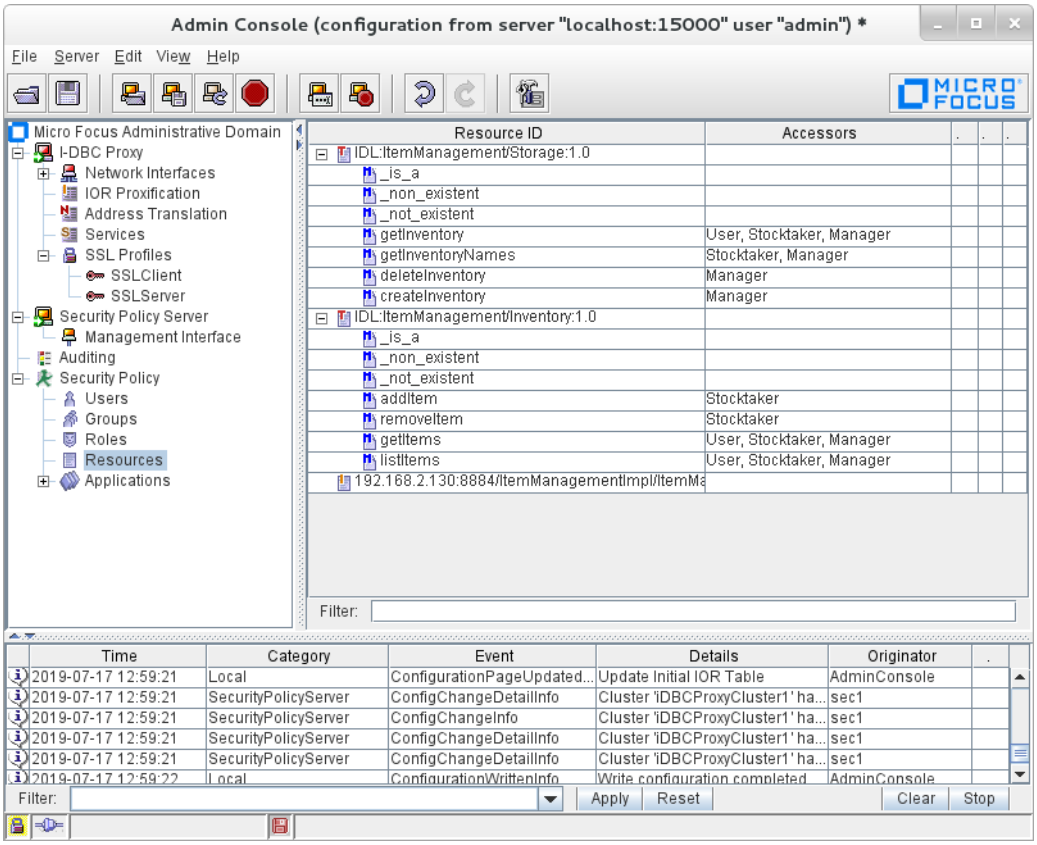


Figure 79 Administration Console: Resource List

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 IDL Type**..., or **New Corba Object**... from the context menu (right-click in the resources table).

If an EJB deployment descriptor or an IDL document describing the interface that you want to expose is available the exposure wizards may be used instead of defining the resource manually (refer to section “Exposure Wizards”). Operations offered by a resource can be viewed by clicking the ⊞ next to the resource.

1. They may as well be Web Services specified by their URL. Web Service Resources are only of interest when configuring WS-DBC’s and are not explained here.

Type-based and Instance-based Resources

There are two types of resources – type-based and instance-based resources. Type-based resources are defined by their IDL Type (usually: `IDL:<pragma prefix/module-hierarchy>/<interface>:<version>`). The first two entries in the resource pane are the **ItemManagement/Storage** and **ItemManagement/Inventory** interfaces, both with the default version "1.0". In CORBA IDL, interfaces can inherit from other interfaces, but the Administration Console does not support inheritance hierarchies. Only rules for IDL types that match exactly are applied.

Instance-based resources are defined by `<hostIP>:<port>/<objectkey>`. The host must be specified by its IP address. The object key must be given in URL encoding. In the example above the object runs on **192.168.2.130** on port **8884**, and has the object key **ItemManagementImpl/ItemManagementPOA/StorageImpl**.

Note that if both an IOR-based resource and an IDL-type-based resource are found for a run-time request, the policy for the IOR-based one takes precedence.

Properties of CORBA resources can be defined on the respective resource properties tabs, explained in the following sections.

Exposure Wizards

The Administration Console offers an exposure wizard to import IDL descriptions of a resource. Furthermore EJB Security Policies can be imported into the DBC Security Policy. The IDL exposure wizard is explained in the following section. The EJB import wizard is explained on "Importing EJB Security Policies into the Security Policy".

IDL Exposure Wizard

The IDL exposure wizard can be accessed via the menu item **File > Import** or via the context menu on the "Security Policy – Resources" panel.

On the first panel of the IDL exposure wizard you select the IDL file.

import options

As import options you can choose to:

- import operation parameters (if you want to define rules for parameter checking).
- a compatibility mode is provided for IDL files which do not strictly follow the CORBA IDL standard. Use the compatibility mode only if the default mode yields an error.

To proceed press the "Next" button.

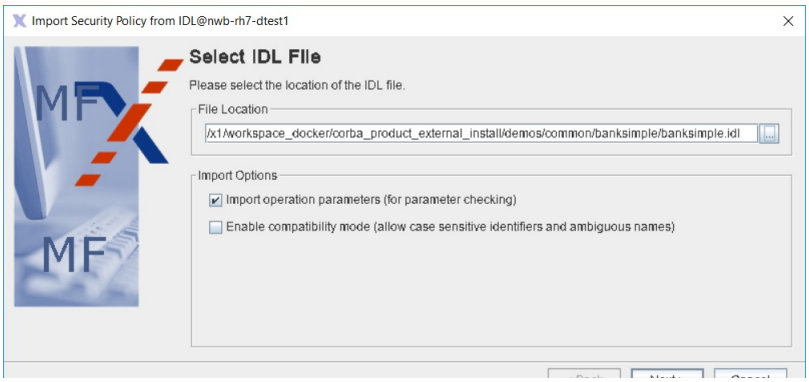


Figure 80 IDL import wizard: Select IDL File

On the next panel you can define which pseudo operations shall be added to the imported resources.

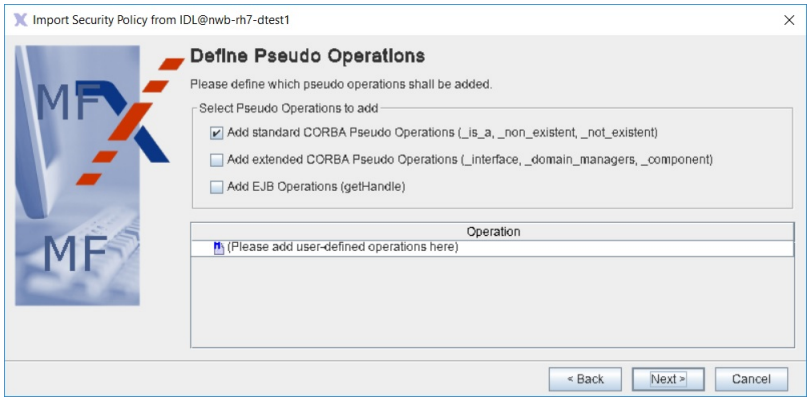


Figure 81 IDL import wizard: Define Pseudo Operations

In the next step the wizard extracts the service descriptions from the IDL document. On the following panel you can select the resources and operations that will be imported into the security policy.

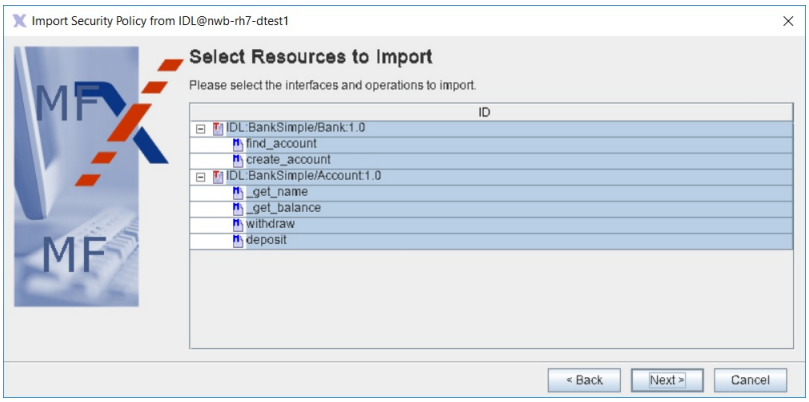


Figure 82 IDL import wizard: Select Resources to Import

conflicts while importing

If the resource already exists in the security policy the Administration 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 the

section “Applications (Application Domains)”. If you don’t want to import into an application, just press the “Next” button.

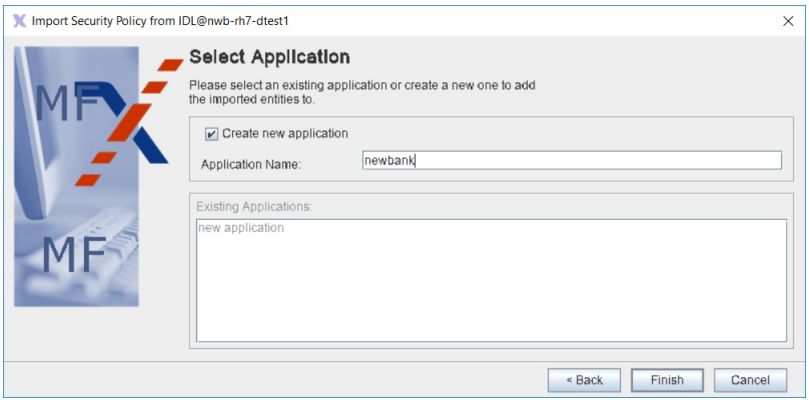


Figure 83 IDL import wizard: Select Application

The IDL import is complete now, but note that the security policy for the imported resource should be refined: By default the imported resource has no accessors, so the DBC will not allow any access (unless the resource 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 resource with its accessors as described in section “Resource Properties” and following.

Importing EJB Security Policies into the Security Policy

The EJB deployment descriptor allows the application assembler to define a security policy to be enforced by the container. This policy consists of method permissions and security roles. A method permission can be a single method, a set of methods with the same name (overloaded methods), or all methods of a specified interface. Method permissions are grouped in security roles. In addition to the user defined security roles, the application assembler can declare method permissions as “unchecked”, i.e., access is granted without restrictions.

To import EJB security policies into the DBC security policy, do the following:

- Select **File > Import EJB Deployment Description...**. This will open a dialog where an enterprise archive containing multiple EJBs or a single EJB jar file can be selected. As import options you can choose to
  - additionally import operations from Home- and Remote-Interfaces that are not explicitly declared in the deployment descriptor.
  - import the two pseudo operations “_is_a()” and “_non_existent()”.
- After clicking “Open”, the deployment descriptors of the EJBs contained in the selected archive will be parsed. This may take a few seconds to complete. After successful parsing, a new dialog pops up. Select the EJBs

that you want to import into the security policy from the available EJBs in the enterprise archive.

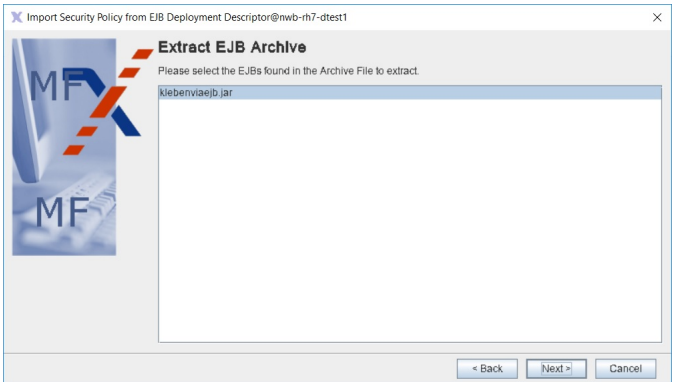


Figure 84 Extract EJB Archive

Select roles, interfaces, and methods for import into the security policy (methods correspond to operations in the security policy).

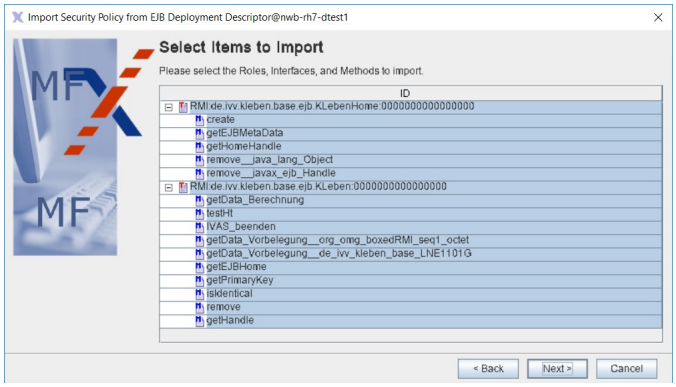


Figure 85 Select Items to Import

During import some method names may have to be changed according to the Java-To-IDL Mapping. For example, IDL doesn't support method overloading, so overloaded Java method names have to be renamed in order to generate correct IDL.

Conflicts while importing

If the role or resource that you want to import already exists in the security policy the Administration 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 role or 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)". If you don't want to import into an application, just press the "Next" button.

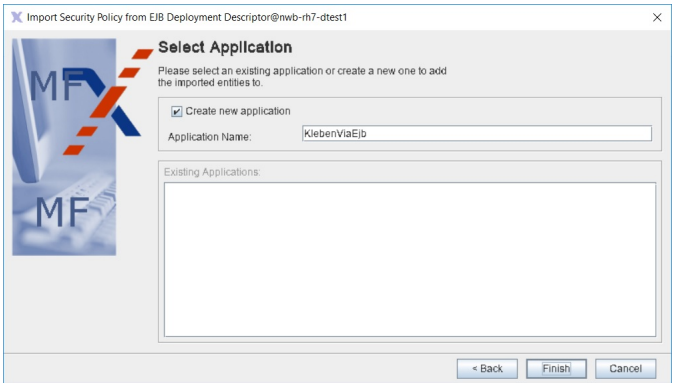


Figure 86 Select Application

The EJB import is complete now, but note that the security policy for the imported roles or resources should be refined. By default the imported resources have no accessors, so the 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.

Resource Properties

To edit a resource, double-click on it or use the menu **Edit > Edit Resource Properties** or the context menu. In any case, the "Resource Properties" panel pops up showing a tree view on the left side and configuration panels on the right side.

The table below depicts the navigation tree (left side) and explains the corresponding panels (right side).

General Transport Layer Protection Authentication Authorization Message Filters Constraints Operations	**General** General Resource Properties, i.e. a comment. **Transport Layer Protection** Defines the required protection to access the resource. **Authentication** Defines the required authentication to access the resource. **Authorization** Defines the resources Accessors. **Message Filters** Defines message filters. **Constraints** Defines time-constraints. **Operations** Defines the resource's operations.

Figure 87 Resource Properties – Tree Navigation

Resource Properties – General

The General page defines a comment.

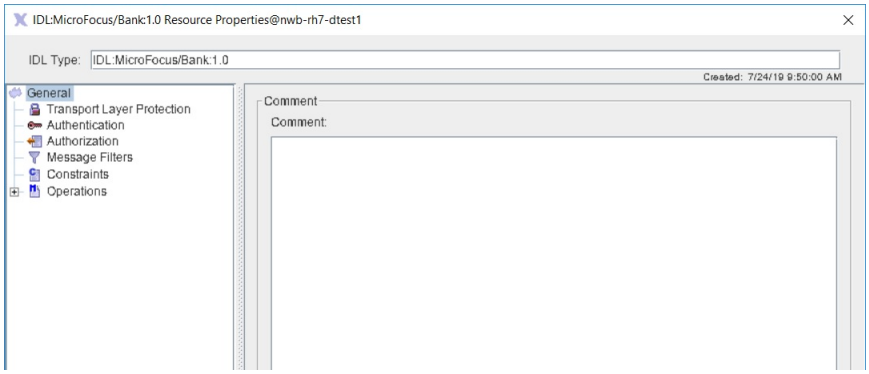


Figure 88 General Resource Properties

Resource Properties – Transport Layer Protection

The Transport Layer Protection page defines the required protection of the connection via which the resource is accessed.

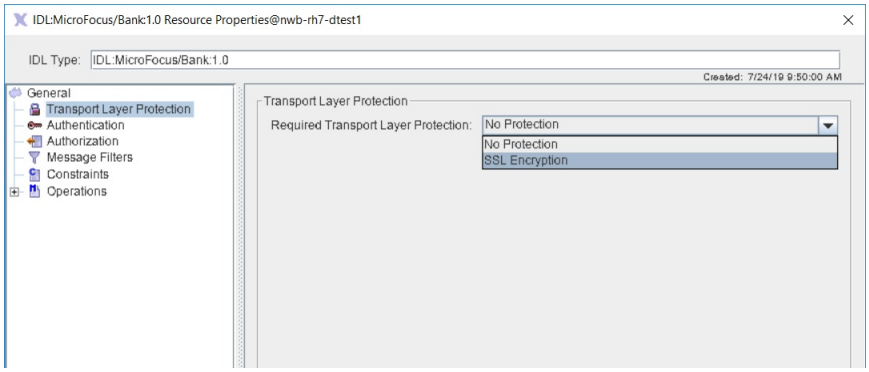


Figure 89 Resource Properties – Transport Layer Protection

Required protection

The *Required Transport Layer 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.

Resource Properties – Authentication

The Authentication page resources can define the access mode by which they allow access to specific operations.

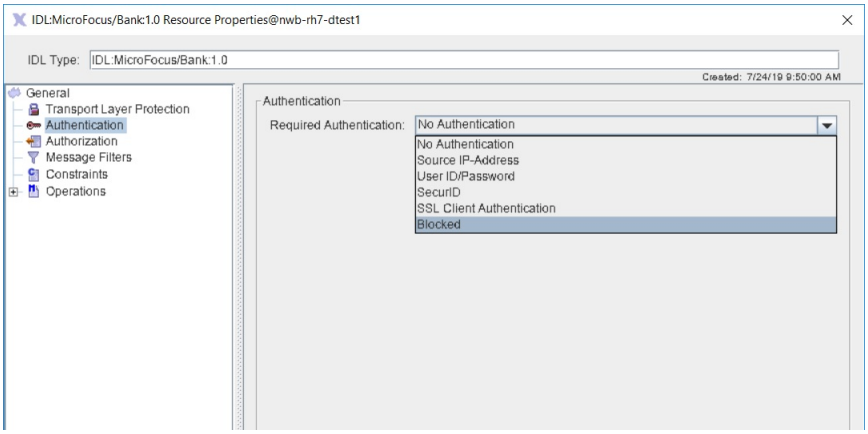


Figure 90 Resource Properties – Authentication

The *Required Authentication* level applies to all operations of a resource. The DBC supports the following authentication mechanisms (listed from the weakest to the strongest):

Required authentication

- *No Authentication*
- *Source IP Address*: authentication by IP address
- *User ID/Password*
- *SecurID*: RSA SecurID authentication
- *SSL Client Authentication* (512 bit RSA, 1024 bit DSS)
- *Blocked*: "emergency stop" to disallow any access

Note that the chosen authentication method is the minimum required method, i.e., if you allow access with at least *User ID/Password* authentication, access is also granted for clients using SSL x.509 credentials (if the user has permissions to access the resource this way). The WS-DBC will look first for authentication information for the stronger mechanism (e.g., SSL) and only consider other mechanisms if no authentication information is available for that level.

Selecting *No Authentication* means that the WS-DBC will accept unauthenticated messages, or any of the 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 only anonymous.

Resource Properties – Authorization

The previous sections explained how protection and authentication requirements for a resource are specified. On the Authorization panel 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.

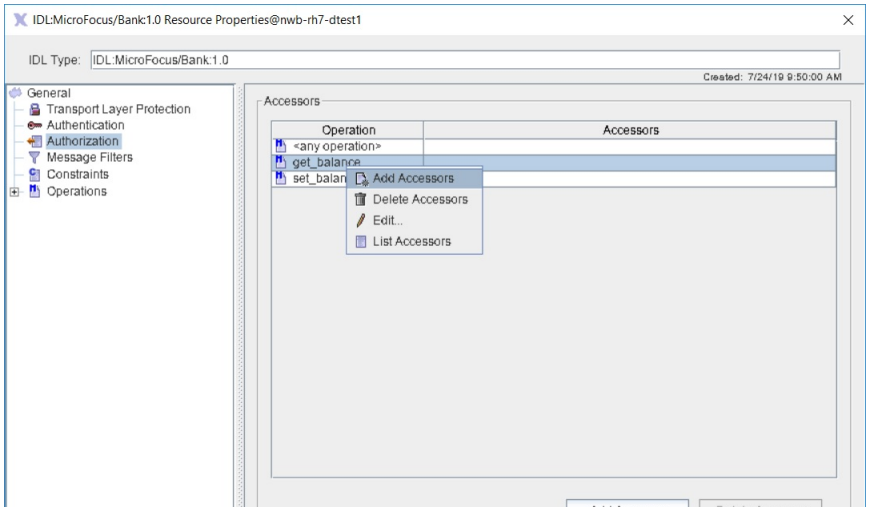


Figure 91 Resource Properties – Authorization

Configuring Accessor Lists

Adding accessors

On the Authorization panel a list of all operations defined for this resource and their corresponding accessors is displayed (see Figure 91). 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.

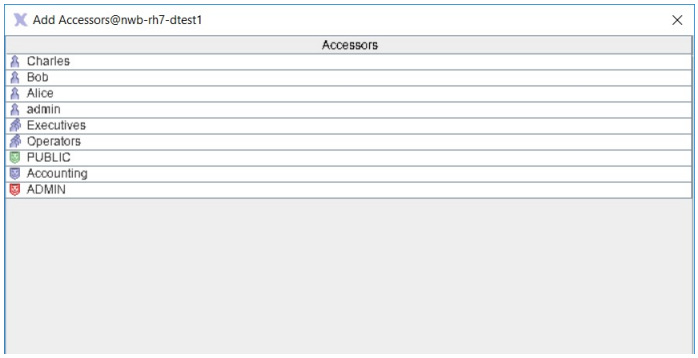


Figure 92 Resource Properties – Selecting Accessors

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 (below the "Operation" tree node). 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: the PUBLIC Role

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.

Note that adding the role "PUBLIC" to the `<any operation>` accessor list implies that all operations which are not defined in the Administration 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.

Resource Properties – Message Filters

Content inspection

The DBC can perform content inspection by validating the messages that are sent to a service. The Message Filters panel lets you define filter rules for operations. Filter terms (rules) can be combined with the boolean operators AND and OR.

Note that the Message Filters panel is disabled if the resource defines no operations. Also it is not possible to create a filter expression if the operation has no parameters. In this case, import the IDL file first (choose **File > Import IDL...**).

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 Message Filters panel. Then right-click into the filter table and choose **New Filter Group...** from the context menu (see screenshot below).

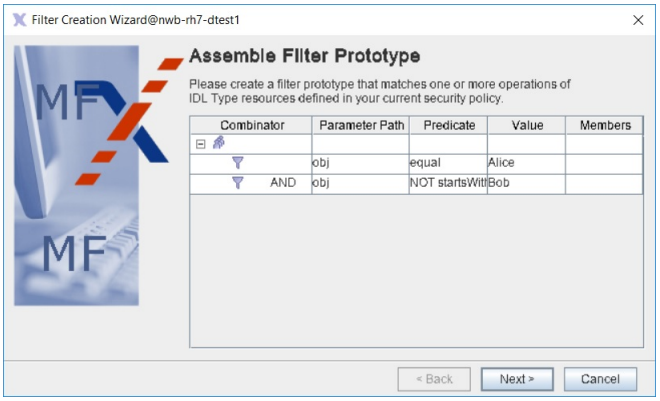


Figure 93 Resource Properties: Defining Message Filter Rules

A new entry in the filter list appears, select the group and then choose **New Filter...** from the context menu. The "Filter Properties" panel pops up.

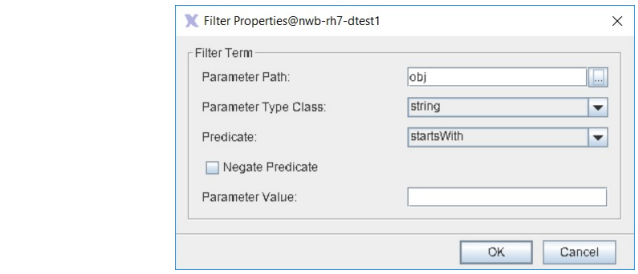


Figure 94 Resource Properties: Defining Filter Properties

You can choose the parameter path by clicking the selector next to the "Parameter Path" text field.

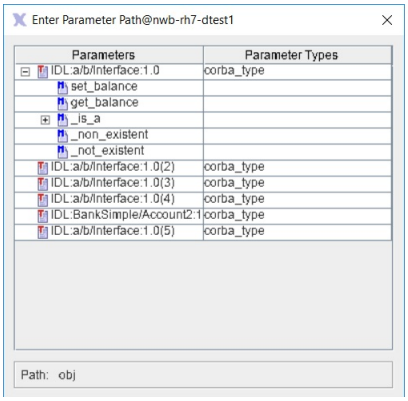


Figure 95 Resource Properties: Enter Parameter Path

After specifying the parameter path, choose a predicate from "Predicate" drop-down menu. Enter the parameter value into the text field. Each predicate can be negated by checking the "Negate Predicate" box preceding the predicate field.

Note that 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**
string	"startswith" (the string starts with...), "endswith" (the string ends with...), "contains" (the string contains), "match" (the string matches), "equal" (the string equals), "equalNocase" (the string equals, the case is not considered), "length <", "length >" (the string length is smaller/greater than), "length <=", "length >=" (the string length is smaller than/greater than or equal), "length = =" (the string length equals)
list (sequence, array)	"length <", "length >" (the list length is smaller/greater than), "length <=", "length >=" (the list length is smaller than/greater than or equal), "length = =" (the string length equals)
enum, numeric	= =, >, <, >=, <=
boolean	"is"

Note that when using the predicate "match" or "matchNocase" the value must be a regular expression. Regular expressions as used in the DBC is explained in more detail in chapter "Regular Expressions".

Combining Simple Filter Expressions

Each row of the "Filter" table shows a simple filter expression. 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. Note that AND has precedence over OR. For this reason the AND operator is indented in the filters table.

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)).

Defining Accessors for Filter Groups

You may define accessors for filter groups. The filter will only be applied when the user is an accessor of this filter group (either directly or as a member of a role or group). When the accessor list of the filter is empty the filter will always be applied.

To define accessors for a filter group select the filter group, choose **Add Accessors...** from the context menu and select accessor from the list.

Unsupported Types

Types that may appear in the IDL document but cannot be used in filter expressions:

- long long,
- Object (i.e., interfaces),
- wchar,
- wstring.

Types that must not appear in the IDL document imported with the Administration Console at all:

- resource types,
- value types,
- fixed types,
- any,
- union.
- value types

Filter Creation Wizard

With the Filter Creation Wizard you can create filter prototypes for resources that may apply to several operations in several resources. The Filter Creation Wizard can be accessed via **Edit > New Filter Prototype...**

(this function is only available when you choose “Resources” on the navigation tree in the Administration Console).

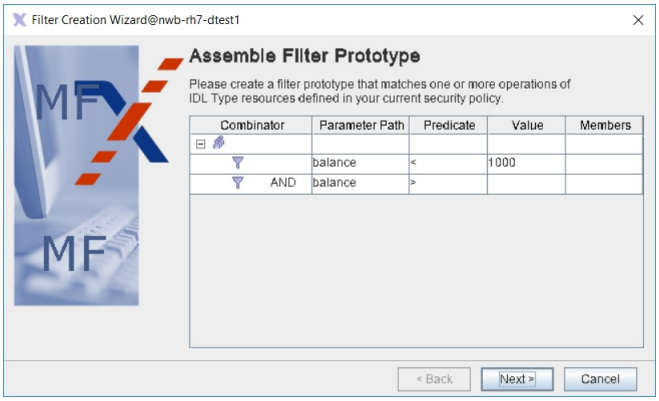


Figure 96 Assemble Filter Prototype

To create a new filter prototype, select **New Filter** in the context menu that pops up when right-clicking into the field. The “New Filter” panel is the same as in non-wizard mode. The only difference is that the parameter type is selectable and wild cards may be used in the parameter path. For example, a valid parameter path would be: “person.address.*”. When you’ve completed the definition of the filter prototype, click the “Next” button.

Select Operations to Apply

On the next panel, the wizard presents all the operations the filter prototype applies to. Select those operations you want your filter prototype to apply to.

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.

Resource Properties – Constraints

The constraints that can be defined for a resource are the same for each type of entity (user, group, role, resource), therefore, constraints are explained only once in section "User Properties – Constraints".

Resource Properties – Operations

If the resource defines operations these operations will appear as tree nodes below the "Operations" node. Select an operation in the navigation tree to see its properties on the panel that appears on the right side. You can edit the operation name in the text field and view the operation's parameters and their types in the table below.

It is not possible to insert parameters of an operation. Parameters of operations are loaded into the configuration while performing the IDL import.

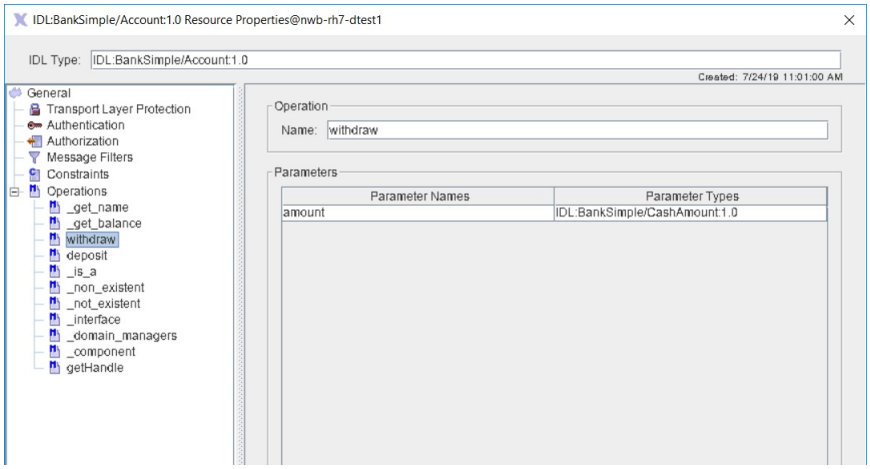


Figure 97 Resource Properties: Operations

Define New Operations

If no IDL document is available for import you can define the operations of a service here. Press the "Add Operation" button or choose "Add Operation" from the context menu that appears when right-clicking on the "Operation" node in the navigation tree. The operation will appear as sub-node of the "Operations" node. Select the new node and change the name of the operation.

Deleting Operations

You can delete operations by right-clicking on the corresponding node in the navigation tree and selecting "Delete Operation" from the context menu.

Callback Support

Callbacks and Bidirectional GIOP

Callback support is required if a CORBA server application needs to invoke operations on CORBA objects (*Callback object*) located in a client application. Callbacks present problems, because CORBA's GIOP was primarily designed as a unidirectional protocol. This means that only the originator of a transport connection may send CORBA requests on this connection. If the acceptor of the connection has to send CORBA request back to the originator it needs to establish a new connection. If firewall traversal is required this behavior implies that the firewall needs to be opened for both, inbound and outbound connections.

The problem has been recognized and an extension for bidirectional GIOP/IIOP1.2 has been included in recent versions of the OMG CORBA specification. Unfortunately, most CORBA middleware products do not support bidirectional IIOP yet, or they provide proprietary approaches only. The WS-DBC offers the following solutions:

Bidirectional IIOP for Inter-WS-DBC connections

- *Bidirectional IIOP for Inter-WS-DBC connections*: This allows for bidirectional connections between WS-DBCs. This option makes only sense if you use inter-WS-DBC communication across the firewall (cascaded I-DBCs). To activate this feature, you can check "Use bidirectional IIOP for Inter-WS-DBC Connections" on the "External Network Interfaces - Advanced" panel.
- *Bidirectional GIOP/IIOP 1.2*: This allows for bidirectional GIOP connections between client, I-DBC, and server. This feature is activated by default. Special permissions must be granted.

Define Callback Permissions in the WS-DBC

- *Callback-Permissions in the WS-DBC*: Use the internal or external acceptors to handle server callbacks (see Figure 98). Special permissions must be granted to the server, so that the server can lookup the callback IOR.

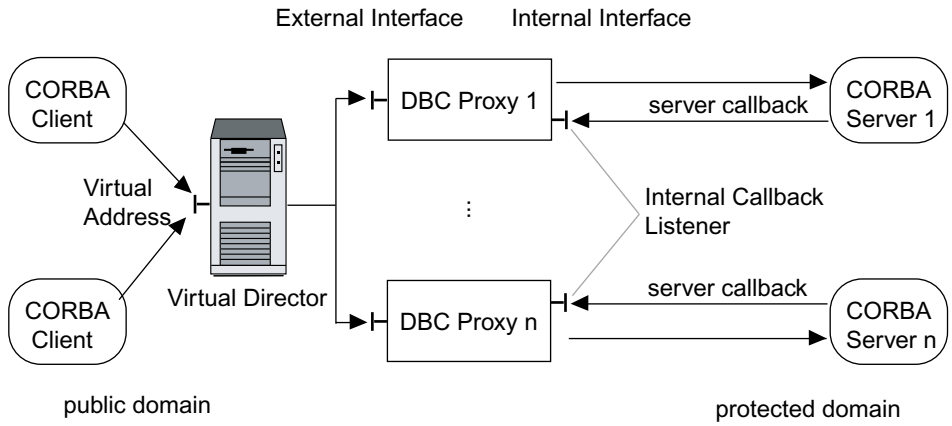


Figure 98 Client Request (from the public domain) and Server Callback

Callback Permissions in the WS-DBC

To perform a callback the server needs to know the client's (proxified) object reference. The client exports his object reference to the server

through the WS-DBC. The server will subsequently establish a new connection to the WS-DBC and try to access the object. A new access session under the server ID (the server's IP address) will be created. From this access session the server has no access to the object reference exported by the client. The required object reference is contained in the client's access session. In general, a client is allowed to access objects in its own access session only (see also section "Number of Access Sessions and Access Session Hierarchy"). To make the required IOR available to the server, you have to grant the server the permission to extend the lookup of IORs to search across all access sessions if the object reference cannot be resolved within the default access session hierarchy. Thus, enabling callbacks corresponds to allowing the lookup of proxified object keys in all access sessions.

Configuring Permissions for Callbacks

You can configure special callback permissions with the Administration Console. Go to the "Resources" panel and add a new resource. Fill in "XDN:Callback:1.0" as resource id and set the "Required Authentication" to "Source IP Address" (on the Incoming Policy tab).

As a first step you can grant "PUBLIC" access to the new resource (on the "Accessors" panel). This allows access to any IOR from any client, regardless of access session boundaries. Therefore, in a production environment you should define proper permissions allowing only your CORBA servers callback access: Add a new user, for example, with the ID "server". Add IP-based authentication for this user ("User Properties – Authentication Methods"). Configure the server's IP address (see screenshot below). If you have several servers using callback, you can use a netmask to map the servers to your "server" identity. For example the IP address 192.168.1.0 together with the netmask 255.255.255.0 implies that all servers with an address from the range 192.168.1.0 to 192.168.1.255 will be resolved to the user name "server".

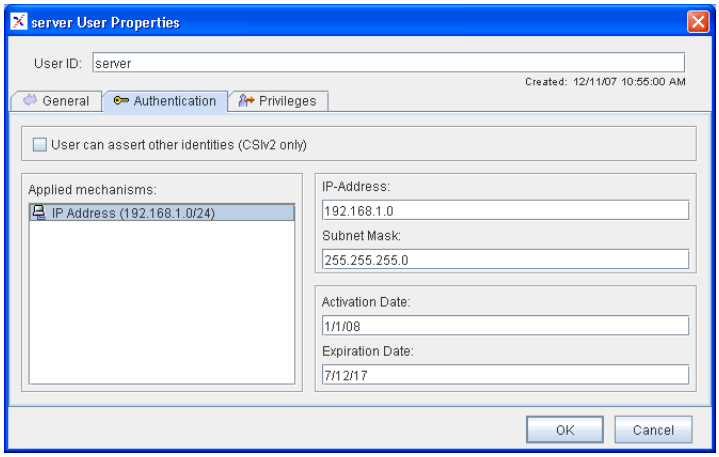


Figure 99 Authentication Methods: IP address

As a last step go to the "User Properties – Privileges" panel and add the resource "XDN:Callback:1.0" to the user's privileges.

Applications (Application Domains)

Application domains group roles and resources into one logical unit. This is useful when several different services are administered with the Administration 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 Administration Console but these administrators can be given the rights to configure only a certain application.

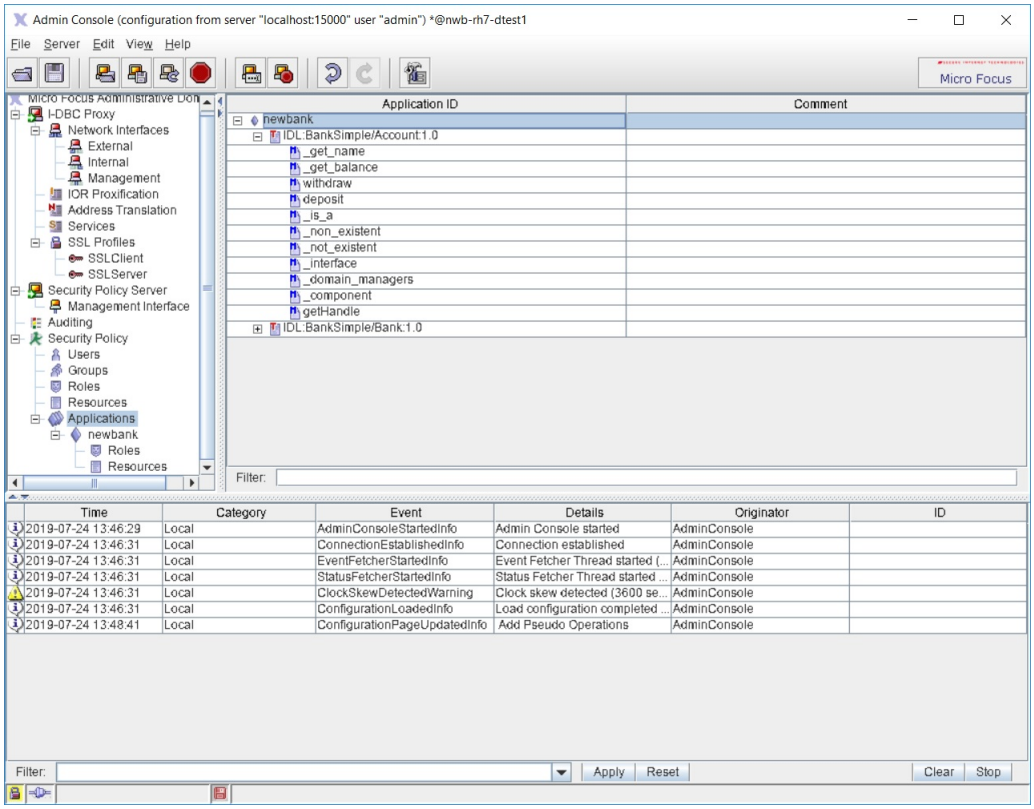


Figure 100 Application Domains

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 Administration 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" and section "Resources" 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.

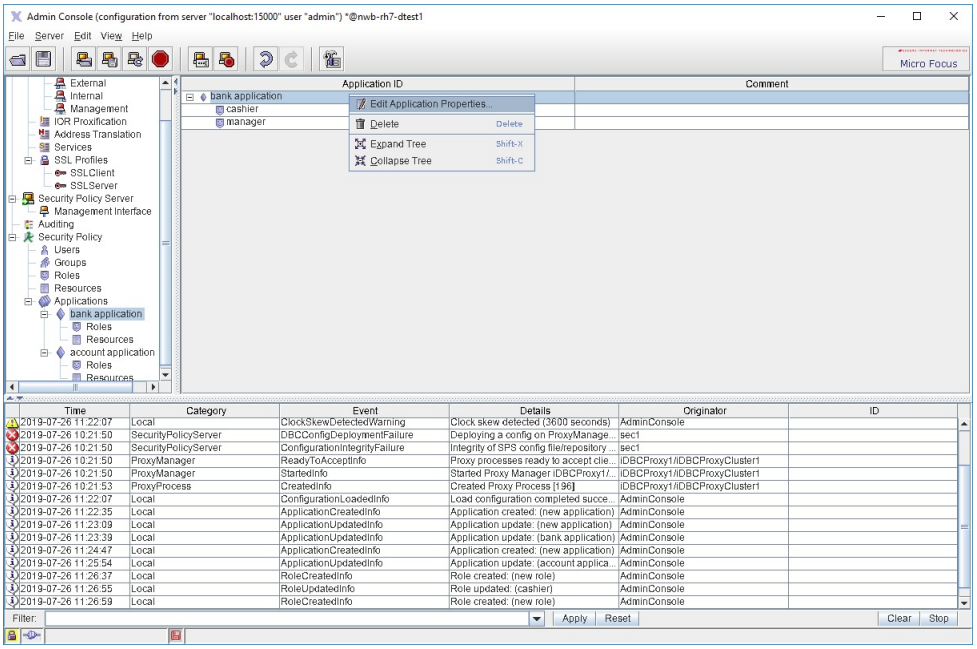


Figure 101 Application Domains - Edit Properties

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).

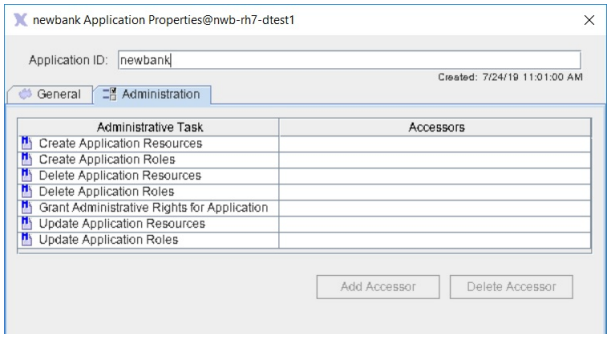


Figure 102 Application Domains - Administration

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 Administration 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").

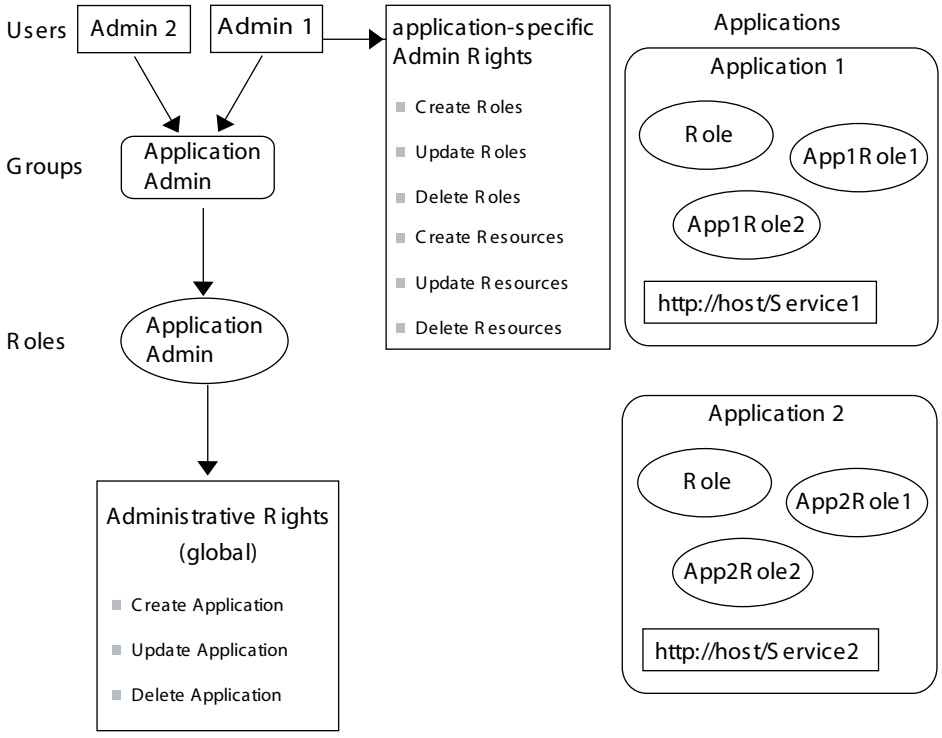


Figure 103 Example Application Domains and administrative rights

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 viceversa.

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 Administration 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.

Learning Mode

The Administration Console offers a learning mode to comfortably compose a security policy. The idea is to run the application across the DBC and to extract recognized users and resources from the DBC's event trace and to built up a security policy based on the extracted information. Before using the Learning Mode you have to carry out the following steps:

- Import/Export IOR: Import the original IOR of your service on the "Initial IORs" panel, and export the proxified IOR.
- Activate audit events: Activate the required audit events on the "Audit Policy" panel (alternatively, you may click the button "Enable All"):
  - `ProxyEngineProxyObjectAccessInfo`,
  - `AuthenticationIPbasedAuthentication` Info, Success and Failure,
  - `SSLAuthenticationCertificate` Success and Failure, and
  - `SSLTransportPeerRecognizedInfo`.
- Run your application through the DBC: Depending on the source of events chosen in the Learning Mode Wizard this step can be done before invoking the wizard or while in learning mode (by selecting **File > Import > Policy From Event Trace...**). Start your client with the proxified IOR. The DBC is configured with an empty security policy. Invoke different methods the service offers, the methods will be recognized by the learning mode.

Follow the steps of the wizard (explained in detail in the following section).

Learning Mode Wizard

Select Trace Source

You can select one of the following event trace sources:

- local event log file (choose the file with the file chooser)
- the event browser of the Administration Console
- events from live stream (directly recorded from the SPS and the Proxy)

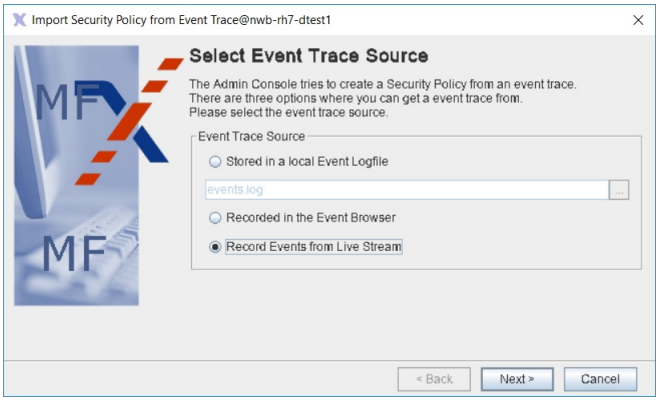


Figure 104 Select Event Trace Source

When selecting "Record events from live stream", you have to run your application through the DBC now. Otherwise this should have been done before.

Note that when choosing to read the event trace from the Administration Console's event browser you should set the history buffer size to a reasonable number, e.g. 100 (the default is 10). The history buffer size can be configured on the **Edit > Preferences > Event Browser** panel.

Select Resources to Import

The recognized resources and methods are displayed. The methods of a service can only be recognized by the learning mode if you invoke the method with the CORBA client. Chose the resources that you want to import. Note that a callback resource is automatically generated for each recognized service even if the service does not support callbacks (for more details on callback resources, please see "Callback Support").

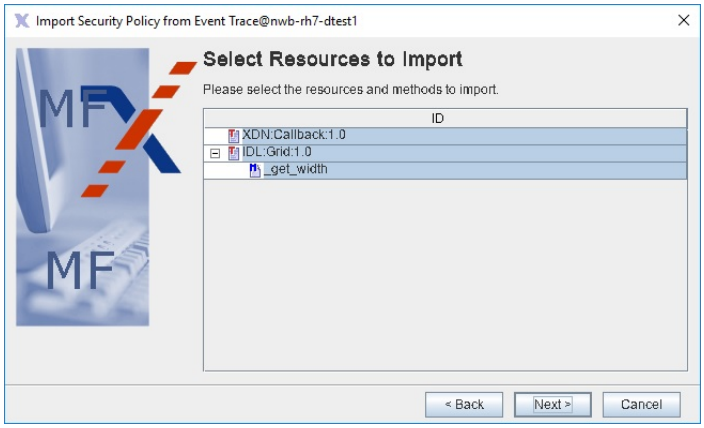


Figure 105 Select Resources to Import

Define Role Memberships

You can define roles here directly. The selected roles will be added to the resource.

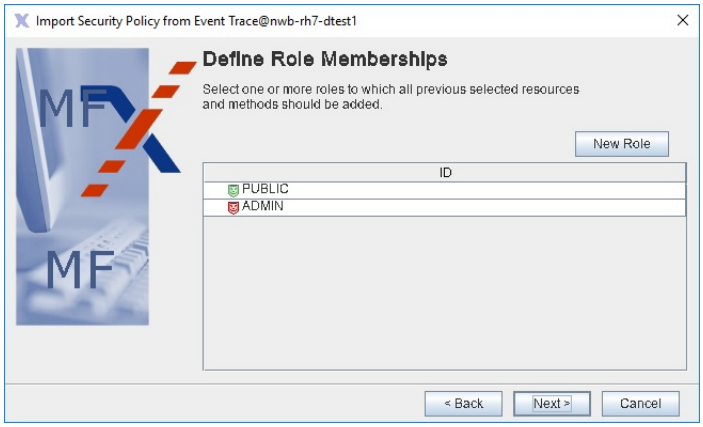


Figure 106 Define Role Memberships

Recognizing Users

The learning mode will always propose to define an "IP-based" user. The userID of this user is the IP-address of the host from which the client connects. Authentication by IP-address will be configured as authentication method for this user.

If you start the CORBA client with SSL the learning mode will process the user's certificate and propose the definition of a corresponding user. The

value of the distinguished name field CN in the client certificate will be used as userID. SSL will be configured as the authentication mechanism for this user.

Users as Actors

If role memberships were defined on the previous panel all recognized users will be displayed in the list. Choose those users from the list that shall be actors in the previously selected roles.

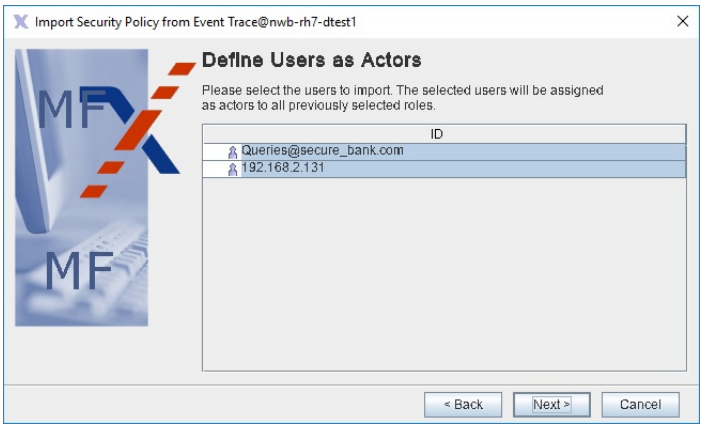


Figure 107 Define Users as Actors

Users as Accessors

If users are recognized they are displayed here. Select those users that shall have access to the previously selected resources.

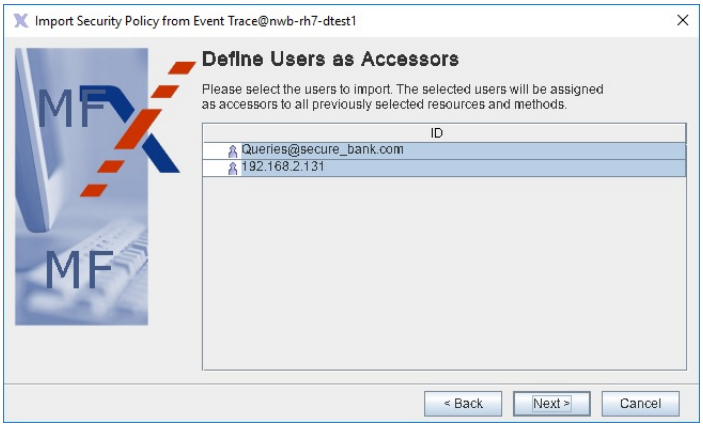


Figure 108 Define Users as Accessors

Pressing the "Next" button will finish the learning mode wizard. The imported security policy should be revised.

Note that the authentication and protection policy of imported resources is "no authentication" and "no protection". Before actually deploying the imported policy you should, for example, configure SSL as required authentication and protection (on the "Resources Properties" panel, see "Resource Properties – Transport Layer Protection" and "Resource Properties – Authentication").

Test the Assembled Policy

Activate access control (on the DBC Proxy panel), save the policy to the SPS, and restart your client. The client will now be allowed to access only those resources that you previously defined in the policy.

Regular Expressions

Regular expressions can be used to define filter rules for parameter checking. The following section describes regular expressions understood by the I-DBC.

I-DBC – Regular Expression Syntax

A regular expression describes strings of characters. It is a pattern that matches certain strings and does not match others. The regular expressions used in the I-DBC for parameter checking follows the POSIX standard for regular expressions. Some features, however, extend the POSIX standard. This will be indicated.

A regular expression consists of one or more branches, separated by '|', matching anything that matches any of the branches. A branch is zero or more constraints or 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.

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 to n (inclusive) matches of the atom (m < n).

The forms using '{' and '}' are known as bounds. The numbers m and n are unsigned decimal integers with permissible values from 0 to 255 inclusive.

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 'admi*' will match for the string "adm", "admi", and "admiiii". Evaluating the string "administration" against the regular expression 'admi*' will also yield 'true', because "admi" is a substring of the "administration" and the regular expression is unterminated.

Atoms

An atom is one of:

- (re) (where `re` is any regular expression) matches for `re`, with the match noted for possible reporting,
- [chars] a bracket expression, matching any one of the chars (see "Bracket Expressions" for more detail),
- . matches any single character,

- \k (where k is a non-alphanumeric character) matches that character taken as an ordinary character, e.g., '\\' matches a backslash character,
- \c where c is alphanumeric (possibly followed by other characters), an escape (extends POSIX), see "Escapes",
- { when followed by a character other than a digit, matches the left-brace character,
- x where x is a single character with no other significance, matches that character.

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".

Constraints

A constraint matches an empty string when specific conditions are met. A constraint may not be followed by a quantifier. The simple constraints are as follows; some more constraints are described later, under "Escapes":

- ^ matches at the beginning of a line,
- $ matches at the end of a line.

Example

The regular expression '^admi.*' matches "adm", "admi", and "admiii" only if they are at the beginning of a line. The regular expression '^administration$' only matches the string "administration" if that is the complete string. Thus, enclosing the regular expression in '^' and '$' is used for a complete match of the whole input.

Note that '^' and '$' are often used to match the exact string, e.g., the expression '^Administration$' matches only the string "Administration" and not the string "Administration Console".

Bracket Expressions

A bracket expression 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 (inclusive) in the collating sequence, e.g., '[0-9]' in ASCII matches any decimal digit. Two ranges may not share an endpoint, e.g., 'a-c-e' is illegal.

To include a literal] or '-' in the list, the simplest method is to enclose it in '[. and .]' to make it a collating element (see below). Alternatively, make it the first character (following a possible '^'), or (extends POSIX) precede it with '\'. Alternatively, for '-', make it the last character, or the second endpoint of a range. To use a literal '-' as the first endpoint of a range, make it a collating element or (extends POSIX) precede it with '\'. With the exception of these, some combinations using '[', and escapes, all other special characters lose their special significance within a bracket expression.

Within a bracket expression, a collating element (a character, a multi-character sequence that collates as if it were a single character, or a collating-sequence name for either) enclosed in '[.' and '.]' stands for the

sequence of characters of that collating element. The sequence is a single element of the bracket expression's list.

Example

The expression '^[1-9][0-9]{3}$' defines a number between 1000 and 9999. Note that if you leave out the '^' and the '$' numbers greater than 9999 will also match, because parts of greater numbers, e.g., "4567" in "1234567" match that regular expression.

Character Classes

Within a bracket expression, the name of a character class enclosed in '[: and :]' stands for the list of all characters (not all collating elements!) belonging to that class. Standard character classes are:

- alpha a letter,
- upper an upper-case letter,
- lower a lower-case letter,
- digit a decimal digit,
- xdigit a hexadecimal digit,
- alnum an alphanumeric (letter or digit),
- blank a space or tab character,
- space a character producing white space in displayed text,
- punct a punctuation character,
- graph a character with a visible representation,
- cntrl a control character.

A character class may not be used as an endpoint of a range.

Example

The regular expression '^[[:digit:]]{1,3}$' matches 1-3 digits, e.g., 0, 22, 999.

Escapes

Escapes (extends POSIX), which begin with a '\' followed by an alphanumeric character, come in several varieties: character entry, class shorthands, constraint escapes, and back references. A '\' followed by an alphanumeric character but not constituting a valid escape is illegal.

Character-entry escapes (extends POSIX) exist to make it easier to specify non-printing and otherwise inconvenient characters in regular expressions:

- \a alert (bell) character, as in C,
- \b backspace, as in C,
- \B synonym for '\' to help reduce backslash doubling in some applications where there are multiple levels of backslash processing,
- \cX (where X is any character) the character whose low-order 5 bits are the same as those of X, and whose other bits are all zero. For example, '\cG' matches Ctrl-G,
- \e the character whose collating-sequence name is 'ESC', or failing that, the character with octal value 033,

- \f formfeed, as in C,
- \n newline, as in C,
- \r carriage return, as in C,
- \t horizontal tab, as in C,
- \v vertical tab, as in C,
- \xhhh (where `hhh` is any sequence of hexadecimal digits) the character whose hexadecimal value is 0xhhh (a single character no matter how many hexadecimal digits are used). For example, '\x41' matches "A",
- \0 the character whose value is 0,
- \xy (where `xy` is exactly two octal digits, and is not a back reference (see below)) the character whose octal value is 0xy. For example, '\46' matches "&",
- \xyz (where `xyz` is exactly three octal digits, and is not a back reference (see below)) the character whose octal value is 0xyz. For example, '\101' matches "A".

Hexadecimal digits are '0'-'9', 'a'-'f', and 'A'-'F'. Octal digits are '0'-'7'.

The character-entry escapes are always taken as ordinary characters. For example, '\135' is] in ASCII, but '\135' does not terminate a bracket expression.

Example

The regular expression '^[^\n].' expresses that the beginning of a line is not a new line. Remember that in a bracket expression the '^' at the beginning of the list matches any single character not included in the rest of the list.

Shorthand Escapes for Character Classes

Class-shorthand escapes (extends POSIX) provide shorthands for certain commonly-used character classes:

- \d [[:digit:]],
- \s [[:space:]],
- \w [[:alnum:]_] (note underscore),
- \D [^[:digit:]] (any character but a digit matches),
- \S [^[:space:]] (any character but a white space matches),
- \W [^[:alnum:]_] (note underscore).

Within bracket expressions, '\d', '\s', and '\w' lose their outer brackets, and '\D', '\S', and '\W' are illegal.

Example

The regular expression '[[:digit:]]{1,3}' is equivalent to '\d{1,3}'. The regular expression '[a-c[:digit:]]' is equivalent to '[a-c\d]'.

Note that '[a-c\D]', (which is equivalent to '[a-c^[:digit:]]') is illegal.

Constraint Escapes

A constraint escape (extends POSIX) is a constraint, matching the empty string if specific conditions are met, written as an escape (note that constraint escapes are illegal within bracket expressions):

- \A matches only at the beginning of the string,
- \m matches only at the beginning of a word,
- \M matches only at the end of a word,
- \y matches only at the beginning or end of a word,
- \Y matches only at a point that is not the beginning or end of a word,
- \Z matches only at the end of the string,
- \m (where `m` is a nonzero digit) a back reference, see below,
- \mnn (where `m` is a nonzero digit, and nn is some more digits, and the decimal value mnn is not greater than the number of closing capturing parentheses seen so far) a back reference, see below.

A word is defined as a sequence of word characters that is neither preceded nor followed by word characters. A word character is an alnum character or an underscore (_) (in shorthand '\w').

A back reference (extends POSIX) matches the same string matched by the parenthesized subexpression specified by the number, so that, e.g., '([bc])\ 1' matches 'bb' or 'cc' but not 'bc'. The subexpression must entirely precede the back reference in the regular expression. Subexpressions are numbered in the order of their leading parentheses. Non-capturing parentheses do not define subexpressions.

In the context of regular expressions, there is an inherent historical ambiguity between octal character-entry escapes and back references, which is resolved by heuristics, as hinted at above. A leading zero always indicates an octal escape. A single non-zero digit, not followed by another digit, is always taken as a back reference. A multi-digit sequence not starting with a zero is taken as a back reference if it comes after a suitable subexpression (i.e., the number is in the legal range for a back reference), and otherwise is taken as octal.

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:

```
^([[:digit:]]{1,3}\.){3}([[:digit:]]{1,3})$
```

The first part of this notation (`([[:digit:]]{1,3}\.)`) states that we expect 1 to 3 digits followed by a period. `{3}`means that the previous expression occurs exactly three times. The last part of the expression states that we expect once more 1 to 3 digits.

Instead of `[[:digit:]]` you can also use the shorthand `\d` or `[0-9]`.

Note that if you want to match an exact value always put '^' in the beginning and '$' at the end of the regular expression. Otherwise subsets

also match. For our example without the '^' and the '$' a value as "1234.22.3.3234" would match and that's maybe not what you want.

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:

```
^[[:digit:]]{4}-[[:digit:]]{2}-[[:digit:]]{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]+)
```

The regular expression can be optimized to: `[-+]?([0-9]*\.)?[0-9]+`

We can modify the regular expression for floating point numbers to also matches floats 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").

Part IV

Appendices

In this part

This part includes appendices that might be helpful to explain certain issues in more detail::

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.

Events in alphabetical order

The following table lists all audit events for the I-DBC.

Note that the *Category* is part of the event name, although it is displayed in a separate column in the event browser. For example, in the event log file a *LoginInfo* event with the category *SecurityPolicyServer* will be displayed as *SecurityPolicyServerLoginInfo* event.

hidden events

Some audit events are not listed in the Administration 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.

Events	Description	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.	**Y**
RequestAcceptedSuccess	A request has been accepted.	
RequestDeniedFailure	A request has been rejected by the ADF. See "ADFRequestDeniedFailure" for details.	
TimerFailure	An ADF request timed out, indicates overload situation.	**Y**
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.	
BasicAuthenticationSuccess	The Proxy successfully authenticated a user with Basic Authentication.	

Events	Description	Hidden
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.	
HTTPBasicAuthenticationFailure	Unused, kept for downward compatibility.	**Y**
HTTPBasicAuthenticationSuccess	Unused, kept for downward compatibility.	**Y**
IPbasedAuthenticationFailure	Failed to authenticate user by ip address.	
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.	
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.	**Y**
MessagingThreadFailure	Internal error in messaging mechanism. Previous message has no events.	**Y**
ReadingFailure	Failure while trying to read input from child.	**Y**
TerminationSuccess	Child has terminated without error.	**Y**
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.	**Y**
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)		
ActivationInfo	Persistence activation state changed.	
FileCreateFailure	Persistence file creation failed.	

Events	Description	Hidden
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 Administration Console.	
IntegrityFailure	Integrity of policy repository has been compromised.	
MiscFailure	LDAP exception, file open exception or entry expired.	**Y**
RetrieveFailure	The policy information defining access rights for the requested target was not found in the repository (see also "PolicyRepositoryRetrieveFailure").	
SyntaxFailure	Wrong format of entity identifier.	**Y**
ValidityFailure	The repository entry is not valid, i.e., current date is not between the entry's activation date and expiration date.	
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.	
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.	

Events	Description	Hidden
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").	
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).	
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.	**Y**
ChildExitedInfo	The proxy (child) has exited normally when shutting down.	**Y**
ChildSignalledFailure	Child exited due to signal.	**Y**
FailedOverInfo	Proxy manager failed over to a different SPS.	
FileError	The ProxyManager failed to open a file.	
InternalFailure	Exception during thread run.	**Y**
ReadyToAcceptInfo	The proxy manager has started and configured all engines, which are now ready to process client connections.	

Events	Description	Hidden
RequestLimitFailure	Number of requests in queue limit reached, request dropped.	**Y**
StartedInfo	The proxy manager has started.	
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.	**Y**
ThreadThresholdInfo	The number of threads has reached the warning level. This warning level is not yet critical.	**Y**
UnknownChildFailure	Child process could not be found.	**Y**
Category: ProxyProcess		
CreatedInfo	The proxy process has started.	**Y**
ProxyManagerCommunicationFailure	Decoding a message from the proxy manager has failed.	**Y**
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 "SSLAuthenticationCertificateFailure".	
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 "SSLTransportHandshakeFailure".	
InternalFailure	An SSL internal error occurred.	**Y**
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).	

Events	Description	Hidden
Category: SecurityPolicyServer		
AccessFailure	A client failed to access the SPSSecurity Policy Server.	
ClientDisconnectedInfo	Client disconnected from the SPSSecurity Policy Server.	**Y**
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 SPSSecurity Policy Server has been changed.	
ConfigReadInfo	A new configuration was written to the SPSSecurity Policy Server.	
ConfigWrittenFailure	The writing of a new configuration to the Security Policy ServerSecurity Policy Server failed.	
ConfigWrittenSuccess	The writing of a new configuration to the Security Policy ServerI-DBC ProxySecurity Policy Server was successful.	
ConfigurationFailure	There was an error in the configuration.	**Y**
ConfigurationIntegrityFailure	Integrity of the configuration file (dbc.config) was compromised (e.g., when editing the configuration outside the Administration 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 Administration Console, for example, to mark the start of a test run.	
PeerConnectionEstablishedFailure	Failed to establish a connection to the peer SPS.	
PeerConnectionEstablishedSuccess	Successfully established a connection to the peer SPS.	
PolicyChangeDetailInfo	Detailed information about a policy change.	
PolicyChangeInfo	The security policy changed.	
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.	

Important Audit Events

This section lists the most important events, explains the information contained in them, and gives possible reasons why it was sent.

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 has been configured to require SSL authentication. Adapt the client to use the required ALA or change the resource's authentication policy with the Administration 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 messages. Adapt the client to use the required protection mechanism or change the resource's protection policy with the Administration 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 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 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.

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.
- The resource mapping is incorrect. Start the Administration 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.

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 Administration Console, go to the "Security Policy – Users" panel, and check that the user is included in the list (section "User Properties – General" 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 Administration 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 Administration 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" 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" 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.

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. I-DBC clients may receive CORBA system exceptions which may be raised or passed on by the Proxy. These are explained in the second part ("CORBA System Exceptions and Minor Codes"). In most cases the DBC Proxy will also generate an audit event. Audit Events are explained in the appendix "Audit Events". Please refer to that appendix for a more detailed description of possible causes.

DBC and SPS Error Messages

The following sections explain the format of 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.

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.

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	182
Address lookup failed	182
Error while reading key file	182
Address already in use	182
Server Socket bind failed	183
System Exception "No Permission"	183
Cannot set SSL private key	183
Invalid key file format	184
Cannot open SSL certificate file	184
Unable to get local issuer certificate	184
Client does not accept Server's certificate	184

License not found

```
Error loading license <filename>
```

Copy the license file (`license.txt`) you received with your software package or via email into the `adm` directory of the SPS and the Proxy. Contact Micro Focus SupportLine (http://supportline.microfocus.com/) for further information.

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 Administration 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. You must then restart the DBC Proxy.

Error while reading key file

```
error while reading key file 'SPSKey.der': No such file or
directory
```

The commonest cause for this is a failure to generate 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 1500015000
```

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 `sps` (as shown above) all should be well, i.e., a Security Policy 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 192.168.1.5:8885: 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 Administration 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 Policy 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 `corba`.

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").

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 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".

CORBA System Exceptions and Minor Codes

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".

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 Naming Service.

Minor code (hexadecimal/decimal value)	**Description**
NoSuchProxyOperation 0x58540001 1481900033	Requested operation is not known by the DBC service object.

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 Naming 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.

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.
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.

INITIALIZE

An 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.

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 Naming Service).

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.

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.

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".
- 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".

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.

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.

Minor code (hexadecimal/decimal value)	Description
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.

NO_RESOURCES

A NO_RESOURCE system exception is raised by the I-DBC if a 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.

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 Naming Service).

Minor code (hexadecimal/decimal value)	Description
PseudoOpNotImplemented 0x58540002 1481900034	The requested pseudo operation is not implemented by the targeted DBC service object (DBC Authenticator, DBC Naming Service).

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.

This exception is commonly created in the I-DBC Proxy: the requested IOR is not available in the I-DBC Proxy. This may be caused by the initial IOR not being configured, or by a new initial IOR table not being saved to the Security Policy Server.

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 Administration Console on the "Initial IOR Table" panel). Another reason might be that the IOR is a transient one and the server was restarted.

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).

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.

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 4 Permitted cipher strings

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.

Table 4 Permitted cipher strings

Cipher String	Meaning
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.
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.
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.
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.
TLSv1, SSLV3, SSLV2	Protocol versions TLS v1.1 and v1.2, 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.
RC4	Cipher suites using RC4.
MD5	Cipher suites using MD5.
SHA1, SHA	Cipher suites using SHA1.

Cipher Suites Offered by the DBC

As part of the SSL configuration in the Administration 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

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W