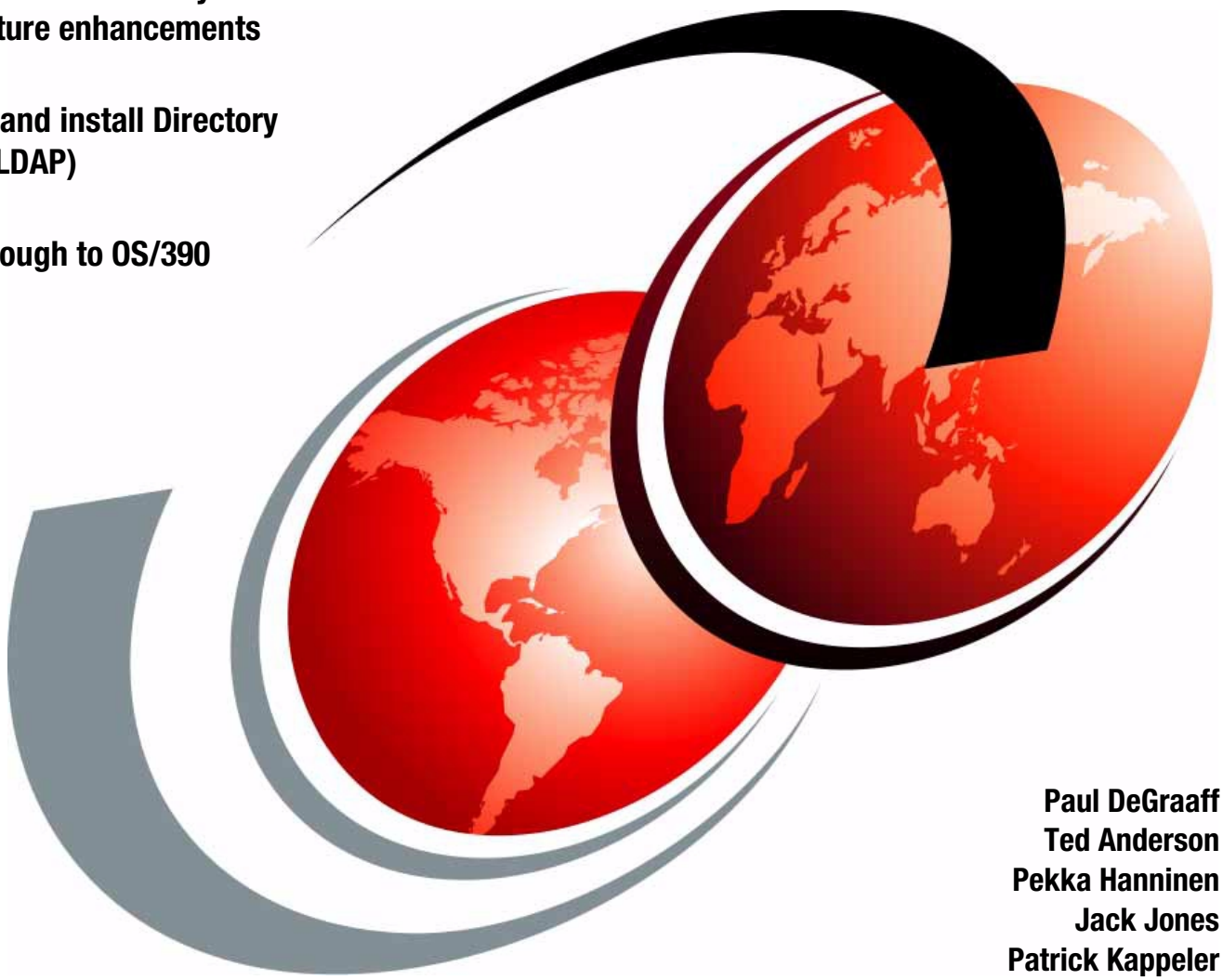
IBM

OS/390 Security Server 1999 Updates: Installation Guide

- Exploit RACF's Public Key Infrastructure enhancements
- Configure and install Directory Services (LDAP)
- Tunnel through to OS/390 using IKE



Paul DeGraaff
Ted Anderson
Pekka Hanninen
Jack Jones
Patrick Kappeler

Redbooks

ibm.com/redbooks

International Technical Support Organization

SG24-5629-00

OS/390 Security Server 1999 Updates: Installation Guide

August 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix F, “Special notices” on page 419.

First Edition (August 2000)

This edition applies to SecureWay Security Server for OS/390, Version 2 Release Number 8, Program Number 5647-A01 for use with the OS/390 Operating System.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook will help you install, tailor and configure the new functions provided in Version 2 Release 7 and 8 of the SecureWay Security Server for OS/390. It will be useful for system programmers, security administrators and webmasters enabling e-Business on the OS/390 platform.

This redbook discusses the following topics:

- OS/390 Cryptographic Services, which consists of:
  - Open Cryptographic Service Facility (OCSF)
  - Open Cryptographic Enhanced Plugin (OCEP)
  - System SSL
- RACF enhancements:
  - RACDCERT enhancements
  - DB2 Version 6 support
  - Generic Identity Mapping support
  - Certificate Name Filtering (CNF)
  - OS/390 Security Services in Java
- OS/390 UNIX System Services security enhancements:
  - Mapping of UIDs and GIDs
  - OS/390 UNIX System Services User Limits
  - Protected User IDs
  - OS/390 UNIX System Services Superuser Granularity
- The LDAP enhancements on OS/390, and their implementation.
- OS/390 Firewall Technologies has major new enhancements in the area of firewall management and support for “dynamic” Virtual Private Networks (VPNs).
- The Telnet Server on OS/390 (TN3270) has new security enhancements for the use of client certificates when using SSL communications, and protection of Telnet ports through new RACF profiles.
- As of Version 2 Release 8, RACFICE is now integrated into SecureWay Security Server for OS/390, enhancing RACF reporting.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Paul DeGraaff, the project leader, is a Certified IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of S/390 Security. Before joining the ITSO, Paul worked in IBM Global Services in the Netherlands as a Senior IT Specialist.

Ted Anderson is a Senior I/T Specialist with IBM’s Software Migration Project Office (SMPO) in North America. He has 18 years of experience in MVS and OS/390-related software. His areas of expertise include OS/390 systems programming, RACF and RACF migrations from competitive security software

and other OS/390 system software products. He holds a B.A. degree in Biology from Bethel College.

Pekka Hanninen is a Service Specialist in Finland. He has 25 years of experience in IBM Large Systems software. He has worked at IBM for three years. His areas of expertise include RACF, cryptography, and security administration. He has written extensively on OS/390 UNIX security enhancements.

Jack Jones is a Certified I/T Specialist working for the S/390 New Technology Center in Poughkeepsie, NY. He has 22 years of experience in the MVS and OS/390 data processing fields. His areas of expertise include system programming, data management, and security. Jack's current area of work is OS/390 security for e-Business: he works with customers to get their OS/390 systems ready for Internet access. Jack is a regular presenter at international conferences such as GUIDE, SHARE, the International Security Conference, and OS/390 Expo.

Patrick Kappeler, formerly a computer specialist in the French Air Force, joined IBM France in 1970, where he was first a S/370 diagnostic programs designer. He held several specialist and management positions, along with international assignments, all dealing with S/370 and S/390 technical support, including hardware and software support for S/390 Parallel Sysplex. In late 1996 he joined the EMEA S/390 New Technology Center in Montpellier, where he now provides consulting and pre-sale technical support in the area of e-Business security.

Thanks to the following people for their contributions to this project:

Terry Barthel
International Technical Support Organization, Poughkeepsie Center

Alison Chandler
International Technical Support Organization, Poughkeepsie Center

Rich Conway
International Technical Support Organization, Poughkeepsie Center

Alex Louwe Kooijmans
International Technical Support Organization, Poughkeepsie Center

Tatsuhiko Kakimoto
International Technical Support Organization, Raleigh Center

John Dayka
IBM Security Server Design

Rich Guski
IBM Security Server Design

Mark Nelson
IBM Security Server Design

Deborah Mapes
IBM Security Server Development

James Sweeny
IBM Security Server Development

Richard Planutis
IBM OS/390 Firewall Technologies Development

David Wierbowski
IBM OS/390 Firewall Technologies Development

Karen Gdaniec
IBM OS/390 LDAP Development

Timothy Hahn
IBM OS/390 LDAP Development

Frans Meij
IBM The Netherlands

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 433 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Part 1. Introduction

Chapter 1. SecureWay Security Server for OS/390

This chapter gives a high-level overview of the enhancements made in 1999 to the SecureWay Security Server for OS/390 and the security enhancements of the SecureWay Communication Server for OS/390.

1.1 SecureWay branding

IBM SecureWay software provides integrated directory, connectivity, and security between users and applications for e-business in a networked world. Every e-business application requires the ability to: locate resources, such as people, information and applications in the network; connect customers, partners, and employees to those resources across multiple systems; address the concern about how to secure communications, data, and transactions. SecureWay integrates these infrastructure requirements to provide the secure network platform needed for e-business. IBM SecureWay software is supported on multiple platforms, including OS/390.

With Release 8, the eNetwork Communications Server for OS/390 has been renamed IBM Communications Server for OS/390, and the OS/390 Security Server is renamed SecureWay Security Server for OS/390.

1.2 Introduction to the SecureWay Security Server for OS/390

Advances in the use of, and general familiarity with, small computers and data processing have increased the need for data security. OS/390 incorporates the SecureWay Security Server for OS/390, which provides a platform that gives you solid security for your entire enterprise, including support for the latest technologies. As a feature of OS/390, the SecureWay Security Server for OS/390 comes with these major components:

- **Resource Access Control Facility (RACF)**
  The primary component of the SecureWay Security Server for OS/390 is the *Resource Access Control Facility*, which works closely with OS/390 to protect its vital resources. Building from a strong security base provided by the RACF component, the Security Server is able to incorporate additional components that aid in securing your system as you make your business data and applications accessible by your intranet, extranets, or the Internet.
- **DCE Security Server**
  DCE Security Server provides user and server authentication for applications using the client-server communications technology contained in the Distributed Computing Environment for OS/390. Beginning with OS/390 Security Server Version 2 Release 5, the DCE Security Server can also interoperate with users and servers that make use of the Kerberos V5 technology developed at the Massachusetts Institute of Technology and can provide authentication based on Kerberos tickets.

  Through integration with RACF, OS/390 DCE support allows RACF-authenticated OS/390 users to access DCE-based resources and application servers without having to further authenticate themselves to DCE. In addition, DCE application servers can, if needed, convert a

DCE-authenticated user identity into a RACF identity and then access OS/390 resources on behalf of that user, with full RACF access control.

- **OS/390 firewall technologies**
  Implemented partly in the Security Server and partly in the SecureWay Communications Server for OS/390, *OS/390 firewall technologies* provide basic firewall capabilities on the OS/390 platform to reduce or eliminate the need for non-OS/390 platform firewalls in many customer installations.

  The Communications Server provides the firewall functions of IP packet filtering, IP security (VPN or tunnels), and Network Address Translation (NAT).

  The Security Server provides the firewall functions of FTP proxy support, SOCKS daemon support, logging, configuration, and administration.

- **LDAP**
  The *LDAP Server* provides secure access from applications and systems on the network to directory information held on OS/390 using the Lightweight Directory Access Protocol. A LDAP client is also provided.

- **OCEP**
  OCEP is the Open Cryptographic Enhanced Plugin. OCEP consists of two service provider modules (which are also called “plug-ins”) that are intended to be used with the Open Cryptographic Services Facility (OCSF) Framework:

  - Trust Policy
  - Data Storage Library

  These service provider modules enable applications to use OS/390 Security Server (RACF), or equivalent product, to provide security functions for digital certificates and key rings.

The SecureWay Security Server for OS/390 provides “one-stop shopping” for security on OS/390. With its integration of RACF and DCE security, its contribution to the OS/390 Firewall Technologies, the LDAP server, and RACF support for client authentication via digital certificates, the Security Server provides complete security both for traditional host-based data processing and for safely expanding your enterprise onto the Internet.

1.3 RACF enhancements

This section highlights the enhancements to the RACF element of the SecureWay Security Server for OS/390.

1.3.1 Improvements to server digital certificates

The RACF component of the OS/390 Release 4 Security Server provides the ability to store digital certificates in the RACF database, and to associate a digital certificate with a RACF user ID. Typically, this is used to map a browser user certificate to a RACF user ID for controlling access to S/390 resources.

A crucial part of implementing digital certificates is managing the certificates used by server applications, and ensuring there is an uncompromised chain of trust. These certificates also have associated encryption keys that are private and must not be revealed.

In OS/390 Release 8, the SecureWay Security Server for OS/390 provides functions to help manage server certificates and to help protect server private keys in a uniform and secure way. The primary application interface to these new functions is provided by Open Cryptographic Enhanced Plug-ins (OCEP), a new component of Security Server. The functions are incorporated into two plug-ins: one for data library services, and one for a trust policy manager. OCEP functions are to be used by applications complying with Common Data Security Architecture (CDSA) standard interfaces. This makes it easier for application developers and independent software vendors (ISVs) to develop and port applications to the S/390 platform. It also helps customers apply consistent security rules to e-business applications that use digital certificates.

1.3.2 Improvements to client digital certificates

The Security Server previously provided the ability to map digital certificates to a RACF user ID. Users could enter the system to log on using their certificate, and access server resources using their RACF user ID. This support has been enhanced such that each user certificate does not need to be installed individually in RACF. You now have the ability to map groups of certificates to a RACF user ID based on criteria such as information contained within the subject's or issuer's distinguished name (DN) or an application or system variable. This function is available on Release 8 via APAR OW40129 and APAR OW40130.

1.3.3 Improvement to user-identity mapping

In Release 8, the Security Server provides the ability to associate RACF user IDs to *Lotus Domino user IDs*. This feature is used in the OS/390 WebSphere HTTP Connector to Domino function of Lotus Domino for S/390; it allows a Web user who has been authenticated with a certificate or RACF user ID by IBM HTTP Server for OS/390 to access Lotus Domino for S/390 without the need for a separate logon to Lotus Domino.

1.4 LDAP enhancements

Lightweight Directory Access Protocol (LDAP) has multiple enhancements in Release 7, including: Java™ support, improved access to data, user ID and password authentication, and server enhancements.

Highlights of the Release 7 enhancements are as follows:

- **LDAP access to Security Server (RACF) data**
  RACF data presents a large set of user and group information that is useful to applications in other environments or on other systems. This item makes RACF user and group information that is accessible through the SAF `R_admin` callable service available via an OS/390 LDAP server to programs on and off the OS/390 platform.
- **LDAP authentication using Security Server RACF**
  User ID and password authentication of LDAP client access to OS/390 LDAP Directory Server can be optionally handled by Security Server RACF rather than by accessing user IDs and passwords stored within the LDAP Server Directory.
- **LDAP multi-server enhancement**
  The LDAP multi-server capability already available with OS/390 Version 2 is

enhanced. It supports multiple LDAP servers on multiple OS/390 systems in a Parallel Sysplex environment, and ensures that LDAP servers that access the same directory are managed according to the goals of the enterprise.

- **LDAP Java (JNDI) support**
  This support provides the industry-standard Java Naming and Directory Interface (JNDI) for the OS/390 LDAP client so that developers can build portable, Java directory-enabled applications that can run on OS/390. It provides access for Java applications on OS/390 to LDAP information stored on OS/390 as well as other LDAP-supported platforms.

In Release 8, the LDAP client and server will be distributed as fully enabled (no charge) portions of the Security Server. LDAP is a fast-growing technology for new network application development. It is a standards-based directory capability implemented with both LDAP Client and LDAP Server components on OS/390. In Release 8, the LDAP Server is being enhanced to support the LDAP Version 3 protocol.The OS/390 Release 8 LDAP Server is able to interoperate with LDAP clients on OS/390 and other platforms which support the V3 protocol.

Highlights of the Release 8 enhancements are as follows:

- Applications are able to interrogate a server to determine whether desired operations are supported by that server.
- Multiple servers are able to manage different parts of the LDAP namespace in a standardized and seamless way. This allows multiple LDAP servers, perhaps in multiple geographic locations, to appear as one server to an application. This can greatly simplify LDAP application design and configuration.
- Applications will be able to authenticate themselves to an LDAP server using an X.509 digital certificate via Secure Socket Layer (SSL). Optionally, certificates can be stored in RACF. Digital certificates can now be used to provide strong authentication of LDAP application users.
- The LDAP server can now handle UTF-8 data. This allows storage and retrieval of any international character data without data loss. This is essential to cross-platform and cross-geography interoperation of LDAP servers and clients.

1.5 OS/390 UNIX System Services (USS) security enhancements

In Release 8, enhancements to OS/390 UNIX System Services ensure that your applications are running in a secure environment. UNIX System Services SuperUser Controls allow for selective assignment of UNIX System Services security. RACF access control can be used to grant limited (or selected) subsets of superuser privileges to specific users, rather than having to grant complete superuser authority. Also, limits that were previously set at a system level (such as the maximum number of threads per process) can now be assigned at a user level. These new controls improve system security by limiting the number of users who require system-wide superuser authority.

Protected user IDs are introduced. Many internal OS/390 processes are assigned RACF user identities which allow auditing and authorization, but are not intended for users (or other systems). In today's e-business environment, protecting such system resources from unauthorized use is imperative. In Release 8, RACF user

IDs that are defined for OS/390 UNIX, UNIX daemons, and other important subsystems or started tasks can be protected from being used for other purposes. You can now define RACF user IDs that cannot be used for activities such as logging on to TSO, or signing on to CICS. This change provides you with further protection from hackers or malicious users.

1.6 OS/390 Cryptographic Services

A new base element is introduced in OS/390 Version 2 release 7, providing cryptographic services. It consists of three elements:

- Integrated Cryptographic Service Facility (ICSF)
- Open Cryptographic Service Facility (OCSF)
- System SSL

The two new elements are discussed next.

1.6.1 OS/390 Open Cryptographic Services Facility

Cryptography comprehensively helps meet multiple security needs, such as confidentiality, authentication and non-repudiation. Open Cryptographic Services Facility for OS/390 addresses these requirements in the emerging Internet, intranet, and extranet application domains. This set of layered cryptographic and security services, (intended to augment the Integrated Cryptographic Services Facility (ICSF)) is suitable for use in applications and application middleware on OS/390 which is targeted for use in the OS/390 UNIX System Services execution environment. It is designed to be compatible with industry standards, such as OpenGroup's Common Data Security Architecture (CDSA). The open architecture can help protect the investment made in implementation of applications by facilitating the reuse of core components of the architecture for different products.

Open Cryptographic Services Facility for OS/390 includes software from RSA Data Security Inc. You can use the software cryptographic services provided by Open Cryptographic Services Facility for OS/390 for application development or test purposes only, provided you first register with RSA Data Security Inc. Refer to the OS/390 Licensed Program Specification section for additional details.

1.6.2 OS/390 System Secure Sockets Layer (System SSL)

System SSL is part of the Cryptographic Services element of OS/390. It repackages SSL function as a set of DLLs and makes the SSL APIs available for use by OS/390 applications. CICS Transaction Server 1.3 will exploit System SSL.

1.7 IBM Communication Server for OS/390

IBM Communication Server provides the enterprise-class network computing infrastructure for OS/390 e-business environments. You can locate resources such as people, information and applications throughout the enterprise. It connects users and applications to these resources across multiple systems with secure communications of data and transactions.

1.7.1 Security improvements

The following security enhancements are available with release 7 of the Communication Server:

- Virtual Private Network (VPN) support meets the latest level of IPSec (RFC 2401-2406 and 2410) to increase its connectivity to servers and clients that may have also upgraded to the latest level.
- Virtual Private Network (VPN) authentication and encryption capability is more robust and Internet Protocol Security (IPSec) performance is improved. CS OS/390 includes recent updates to the evolving standards for IPSec, also known as VPN, that strengthen the ability to authenticate the origin of a message, that it has not been modified during transmission, and that it is "fresh", meaning that it is not a resend of a packet already received. This is achieved through the addition of stronger authentication algorithms, HMAC-SHA and HMAC-MD5, and "replay protection". Together these provide stronger security against brute force attack on the network. Triple DES support is added to provide dramatically improved encryption capability for network security. Also, CS OS/390 IPSec support has been revised to improve performance. S/390 encryption hardware is used, when present, for additional performance improvement.
- Enhancements are also provided to allow the configuration of Firewall Technologies on as many TCP/IP stacks on OS/390 as are connected to the Internet/intranet.
- Scalability of the Firewall FTP and SOCKS daemons is being enhanced to allow a greater volume of concurrent user connections with more efficient utilization of system resources.
- Configuration tasks are being simplified by the addition of a Graphical User Interface (GUI). This GUI is similar to the one provided by the AIX and NT versions of the IBM Firewall. This similarity should ease administration tasks for you, if you have these firewalls installed in your enterprise.

The following security enhancements are available with release 8 of the SecureWay Communication Server:

- Internet Key Exchange (IKE) is an Internet Engineering Task Force-endorsed key and security association management protocol for IPSec. With the IKE support available in the Security Server, CS OS/390 can automatically create and securely distribute encryption keys for dynamic IP clients. This will help to reduce efforts to manage and distribute encryption keys. The availability of non-disruptive key refresh makes it much more practical to change keys frequently. This practice fortifies your protection against brute-force attacks on the network.
- Security against unauthorized access to SNA applications by TCP/IP clients is improved with the addition of Secure Sockets Layer (SSL) client authentication. This capability requires end-users to present a valid certificate before accessing the TN3270 server.
- TN3270 exploitation of RACF certificates provides an additional security checking mechanism to ensure an end-user is authorized to use the TN3270 server. This pre-login access control is exclusive to S/390. It is provided by mapping an SSL-authenticated client certificate to a RACF user ID.

- SNA users can now take advantage of Triple DES session-level encryption for improved encryption capability. These enhancements combine Internet-based technology and mainframe strengths to provide more secure access to mission-critical SNA applications.

  The Triple DES function requires at least a 9672 R5 Server. If you have a S/390 Parallel Enterprise Generation 4 Server and you plan to use the Triple DES support in ICSF, you will need the appropriate LIC code, the Triple DES feature, and a new enablement diskette that supports Triple DES. For the Generation 4 Server, you will need driver 98 and LIC codes EC F10640 or EC F10667. For Generation 5 and 6 Servers, the Triple DES support is included in the product.

  Note: Triple DES is export-controlled and restricted to certain industries abroad. It may not be exported except pursuant to an appropriate license.

1.8 Java for OS/390

Java for OS/390 at the JDK 1.1.8 level includes functional enhancements such as:

- Migration Aid for Java 2 Security Framework
- Additional security support

The Java 2 backport items (Migration Aid, RMI/IIOP, and SWING) will help OS/390 customers to begin using some Java 2 functions as they bring their Java for OS/390 at the JDK 1.1.8 level into production.

Security enhancements provide support for Java resource access authorization checking through the OS/390 System Authorization Facility (SAF) interfaces. They are designed to work with the Security Server (RACF) or equivalent.

These enhancements, described on the Java for OS/390 Web site, are the first installment of support that integrates Java security with our existing base system security.

1.9 IBM HTTP Server for OS/390

Digital Certificate authentication is supported for any X.509 format digital certificate issued by any Certificate Authority. In addition, the IBM HTTP Server for OS/390 supports strong authentication of digital certificates issued by IBM Vault Registry product, including checks for revoked certificates via Vault Registry's Certificate Revocation List (CRL).

HTTP Server Certificate Authority (CA) Servlet can be used to issue locally-produced digital certificates suitable for use in Netscape and Microsoft Internet Explorer browsers and in other SSL applications that support X.509 certificates.

Part 2. Enhancements to SecureWay Security Server for OS/390

Chapter 2. OS/390 cryptographic services

This chapter describes the installation and implementation steps necessary to install the new elements of the OS/390 Cryptographic services available with OS/390 Version 2 Release 7 and OS/390 Security Server Version 2 Release 8, specifically:

- Open Cryptographic Services Facility (OCSF)
- Open Cryptographic Enhanced Plugin (OCEP)
- System SSL

2.1 Open Cryptographic Services Facility

The Open Cryptographic Services Facility (OCSF) is a derivative of the IBM Keyworks technology, which is an implementation of the Common Data Security Architecture (CDSA) for applications running in the UNIX services environment.

Recently, cryptography has come into widespread use in meeting multiple security needs, such as confidentiality, integrity, authentication, and non-repudiation. In order to address these requirements in the emerging Internet, intranet, and extranet application domains, the Open Cryptographic Services Facility was developed. The OCSF is a comprehensive set of layered security services suitable for use in operating systems such as IBM AIX(R), OS/390, and OS/400(R). On Windows NT/95, Solaris, and HP-UX are middleware programs that are embedded in applications, or are provided as a component of cryptographic security toolkits. The OCSF focuses on security in peer-to-peer, store-and-forward, and archival applications. It is designed to be compliant with industry standards such as OpenGroup, and is applicable to a broad range of hardware and operating system platforms. OCSF is intended to include full life cycle key management and portable credentials. The definition of such a set of layered security services and an open architecture protects the investment made in implementation of security applications by facilitating the reuse of core components of the architecture for different products.

The security services available in the OCSF are defined by the categories of service provider modules that the architecture accommodates. These service providers are:

- Cryptographic Service Providers
- Trust Policy Libraries
- Certificate Libraries
- Data Storage Libraries

Figure 1 on page 14 gives an overview of the OCSF Framework.

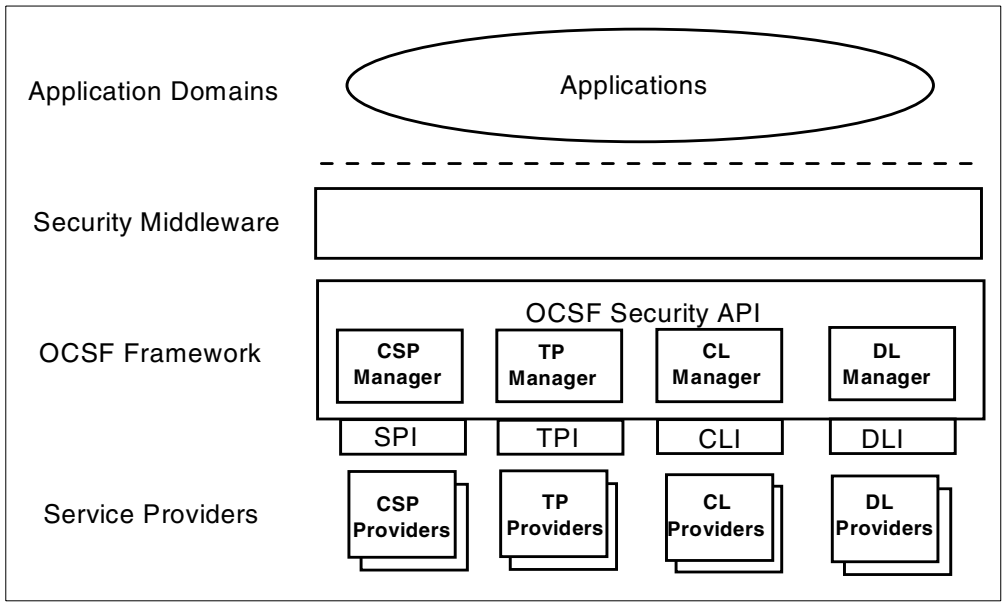


Figure 1. OCSF framework

For a technical presentation on the OCSF framework see the redbook *OS/390 Security Server 1999 Updates: Technical Presentation Guide*, SG24-5627.

The next section will focus on the installation and implementation steps to enable OCSF.

2.1.1 Installation of OCSF

The installation of the OCSF framework is achieved through running a set of installation scripts. Depending on the Security Level features available to you, you will run the following installation scripts:

- Security Level 2 or Level 3
  - ocsf_install_basic_crypto, see 2.1.3, “ocsf_install_basic_crypto” on page 15.
  - ocsf_install_strong_crypto, see 2.1.4, “ocsf_install_strong_crypto” on page 16.
- Security Level 1 or the French feature
  - ocsf_install_basic_crypto, see 2.1.3, “ocsf_install_basic_crypto” on page 15.

2.1.2 RACF setup for OCSF

The use of OCSF services is controlled by the following RACF facility class profiles:

CDS.CSSM Authorizes the daemon to call OCSF services

CDS.CSSM.CRYPTO Authorizes the daemon to call a Cryptographic Service Provider (CSP)

CDS.CSSM.DATALIB Authorizes the daemon to call a Data Library (DL) Service Provider

You need to define these profiles before running any OCSF application, or before running the OCSF installation scripts. If these facility class profiles are not defined, OCSF services are unavailable.

An OCSF application, and the OCSF installation scripts described, must execute under the security context of a user identity which has been granted *read* access to the OCSF facility class profiles.

To define those profiles issue the following RACF commands:

```
RDEFINE FACILITY CDS.CSSM UACC(NONE)
RDEFINE FACILITY CDS.CSSM.CRYPTO UACC(NONE)
RDEFINE FACILITY CDS.CSSM.DATALIB UACC(NONE)
```

Next, permit the user executing the installation scripts READ access to these facility class profiles, issuing the following RACF commands:

```
PERMIT CDS.CSSM CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT CDS.CSSM CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT CDS.CSSM CLASS(FACILITY) ID(userid) ACCESS(READ)
```

The user executing the installation scripts needs to have access to the BPX.SERVER and BPX.DAEMON facility class profile with read access. Issue the following RACF command to give the user access:

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT BPX.DAEMON CLASS(FACILITY) ID(userid) ACCESS(READ)
```

Note: The RACF FACILITY class might be raclisted, so you need to refresh the instorage profiles.

Note: We feel these authorities should only be granted at the time of installation. No one other then UNIX daemons require these authorizations.

2.1.3 ocsf_install_basic_crypto

To execute the installation script `ocsf_install_basic_crypto` take the following steps:

1. Make sure the /`usr/lpp/ocsf/bin` directory is added to your LIBPATH:

   ```
   export LIBPATH=$PATH:/usr/lpp/ocsf/bin
   ```

2. Go to the following directory:

   ```
   CD /usr/lpp/ocsf/bin
   ```

3. Run the required Installation script:

   ```
   ocsf_install_basic_crypto
   ```

```
GRAAFF @ SC57:/usr/lpp/ocsf/bin>ocsf_install_basic_crypto
Installing CSSM...
CSSM Framework successfully installed
Installing IBMTP...
Addin successfully installed.
Installing IBMTP2...
Addin successfully installed.
Installing IBMCL...
Addin successfully installed.
Installing IBMCL2...
Addin successfully installed.
Installing IBMDL2...
Addin successfully installed.
Installing IBMWKCSP...
Addin successfully installed.
Installing IBMCCA...
Addin successfully installed.
GRAAFF @ SC57:/usr/lpp/ocsf/bin>
```

2.1.4 ocsf_install_strong_crypto

To execute the installation script `ocsf_install_strong_crypto` take the following steps:

1. Make sure the `/usr/lpp/ocsf/bin` directory is added to your LIBPATH:

   ```
   export LIBPATH=$PATH:/usr/lpp/ocsf/bin
   ```

2. Go to the following directory:

   ```
   CD /usr/lpp/ocsf/bin
   ```

3. Run the required Installation script:

   ```
   ocsf_install_strong_crypto
   ```

```
GRAAFF @ SC57:/usr/lpp/ocsf/bin>ocsf_install_strong_crypto
Installing IBMSWCSP...
Addin successfully installed.
GRAAFF @ SC57:/usr/lpp/ocsf/bin>
```

2.1.5 Common errors

The error message `module cdsport.dll was not found` occurs when you do not have the `/usr/lpp/ocsf/bin` directory in your LIBPATH.

```
GRAAFF @ SC57:/usr/lpp/ocsf/bin>ocsf_install_basic_crypto
Installing CSSM...
CEE3501S The module cdsport.dll was not found.
         From entry point ProcessArgs at compile unit offset +00000550 at addres
s 0F609EA8.
Error: could not install CSSM
```

The error `bad rc = 65` occurs when the user executing the installation script is not authorized to the facility class profiles mentioned in 2.1.2, “RACF setup for OCSF” on page 14.

```
GRAAFF @ SC57:/usr/lpp/ocsf/bin>ocsf_install_basic_crypto
Installing CSSM...
CSSM Framework successfully installed
Installing IBMTP...
CSSM_Init failed
Exiting with bad rc = 65
Error: could not install IBMTP.
GRAAFF @ SC57:/usr/lpp/ocsf/bin>
```

2.1.6 Installation verification procedures for OCSF

Depending again on the Security Level installed, you need to run one or two *installation verification procedures* (IVP):

- Security Level 2 or 3
  - ocsf_baseivp, see 2.1.6.1, “ocsf_baseivp” on page 17.
  - ocsf_scivp, see 2.1.6.2, “ocsf_scivp” on page 18.
- Security Level 1 or the French Feature
  - ocsf_baseivp, see 2.1.6.1, “ocsf_baseivp” on page 17.

2.1.6.1 ocsf_baseivp

To run ocsf_baseivp take the following steps:

1. Go to the following directory:

   ```
   CD /usr/lpp/ocsf/ivp
   ```

2. Run the IVP:

   ```
   ocsf_baseivp
   ```

```
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>ocsf_baseivp
Starting OCSF base addins ivp

Initializing CSSM
CSSM Initialized

Attaching ibmwkcsp

 ************************************************************************
 * Portions of the IBM Software Cryptographic Service Provider or IBM    *
 * Weak Software Cryptographic Service Provider contain software         *
 * provided by RSA Data Security, Inc. Before use, see                   *
 * /usr/lpp/ocsf/README.FIRST for required terms.                        *
 ************************************************************************
Attach successful, Detaching ibmwkcsp
Detach of ibmwkcsp successful
Attaching ibmcca
Attach successful, Detaching ibmcca
Detach of ibmcca successful
Attaching ibmcl
Attach successful, Detaching ibmcl
Detach of ibmcl successful
Attaching ibmcl2
Attach successful, Detaching ibmcl2
Detach of ibmcl2 successful
Attaching ibmdl2
Attach successful, Detaching ibmdl2
Detach of ibmdl2 successful
Attaching ibmtp
Attach successful, Detaching ibmtp
Detach of ibmtp successful
Attaching ibmtp2
Attach successful, Detaching ibmtp2
Detach of ibmtp2 successful
Completed OCSF base addins ivp
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>
```

2.1.6.2 ocsf_scivp

To run ocsf_scivp take the following steps:

1. Go to the following directory:

   ```
   CD /usr/lpp/ocsf/ivp
   ```

2. Run the IVP:

   ```
   ocsf_scivp
   ```

```
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>ocsf_scivp
Starting OCSF strong crypto ivp

Initializing CSSM
CSSM initialized

Attaching ibmswcsp

 ********************************************************************************
 * Portions of the IBM Software Cryptographic Service Provider or IBM    *
 * Weak Software Cryptographic Service Provider contain software         *
 * provided by RSA Data Security, Inc. Before use, see                   *
 * /usr/lpp/ocsf/README.FIRST for required terms.                        *
 ********************************************************************************
Attach successful, Detaching ibmswcsp
Detach of ibmswcsp successful

Completed OCSF strong crypto ivp
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>
```

When you have run the installation verification procedures and they run correctly, your installation is complete.

For an example of the usage of OCSF, see 5.4, “Encryption support for password values stored in LDAP” on page 188 on how LDAP uses OCSF to encrypt password values stored in a LDAP server on OS/390.

2.2 Open Cryptographic Enhanced Plugin (OCEP)

This section details the installation of the services that are provided by Open Cryptographic Enhanced Plug-ins (OCEP).

OCEP consists of two service provider modules (which are also called “plug-ins”) that are intended to be used with the OCSF Framework:

- Trust Policy
- Data Storage Library

These service provider modules enable applications to use OS/390 Security Server (RACF), or equivalent products, to provide security functions for digital certificates and key rings.

The OCEP service provider modules implement a subset of the application programming interfaces (APIs) that are defined by OCSF. Applications can use these OCEP service provider modules, and their supported APIs, to retrieve and use digital certificates and private keys that are stored in the RACF database on an OS/390 system.

In addition to the OCSF Framework, the OCEP service provider modules are intended to work with the OCSF Certificate Library and Cryptographic Service Provider modules. As Figure 2 shows, the OCSF Framework itself manages the interactions between the service provider modules and the applications that use them.

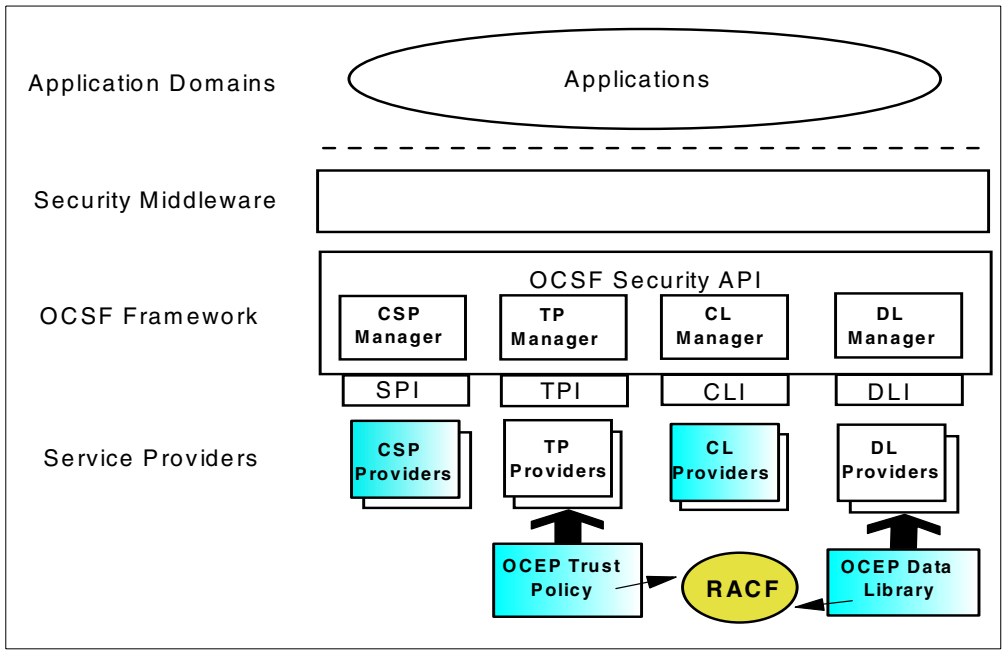


Figure 2. OCSF/OCEP infrastructure

For a technical presentation on OCEP, see the redbook: *OS/390 Security Server 1999 Updates: Technical Presentation Guide,* SG24-5627.

2.2.1 Installation of OCEP

Before you can run any applications that use the OCEP service provider modules, you must first ensure that several tasks have been completed for Open Cryptographic Services Facility (OCSF). The following items must be reviewed and completed:

- The OCSF code must be properly installed and configured on your system, as documented in 2.1.1, “Installation of OCSF” on page 14.
- Any necessary security authorizations must be granted and program controls must be established, as documented 2.1.2, “RACF setup for OCSF” on page 14.

OCEP provides an installation script, called `ocep_install`, that installs the OCEP code and registers the service provider modules with the OCSF Framework.

We recommend that the script be run from a superuser, which is a user ID that has been defined with a UID of 0.

To install the OCEP service provider modules, perform the following steps:

1. Go to the directory where OCEP is installed:

   ```
   cd /usr/lpp/ocsf/bin
   ```

2. Run the OCEP installation script:

   ```
   ocep_install
   ```

```
GRAAFF @ SC57:/usr/lpp/ocsf/bin>ocep_install
Installing IBMOCEPTP...
Addin successfully installed.
Installing IBMOCEPDL...
Addin successfully installed.
GRAAFF @ SC57:/usr/lpp/ocsf/bin>
```

2.2.2 Installation verification procedure for OCEP

After you have completed the steps described in 2.2.1, “Installation of OCEP” on page 20, you must run `ocep_ivp`, the OCEP installation verification program, to ensure that the OCEP code is installed and configured correctly. As with the installation script, we recommend that this IVP be run from a superuser.

Perform the following steps:

1. Go to the directory that contains the IVP:

   ```
   cd /usr/lpp/ocsf/ivp
   ```

2. Run the OCEP IVP program:

   ocep_ivp

```
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>ocep_ivp
Starting OCEP IVP

Initializing CSSM
CSSM Initialized

Attaching ibmocepdl
Attach successful, Detaching ibmocepdl
Detach of ibmocepdl successful

Attaching ibmoceptp
Attach successful, Detaching ibmoceptp
Detach of ibmoceptp successful

Completed OCEP IVP
GRAAFF @ SC57:/usr/lpp/ocsf/ivp>
```

When you have run the installation verification procedures and they run correctly, your installation is complete.

2.3 System SSL

Secure Sockets Layer (SSL) is a communications protocol that provides secure communications over an open communications network (for example, the Internet). The SSL protocol is a layered protocol that is intended to be used on top of a reliable transport, such as Transmission Control Protocol (TCP/IP). SSL provides data privacy and integrity, as well as server and client authentication based on public key certificates. Once an SSL connection is established between a client and server, data communications between client and server is transparent to the encryption and integrity added by the SSL protocol.

OS/390 provides a set of SSL C/C++ callable application programming interfaces that, when used with the OS/390 Sockets APIs, provide the functions required for applications to establish this secure sockets communications.

2.3.1 Dependencies

OS/390 System SSL has the following software dependencies:

- Cryptographic Services System SSL (FMID HCPT270)

  System SSL is part of Cryptographic Services Base element of OS/390. (The Cryptographic Services Base members are installed in the *pdsname*.SGSKLOAD PDS.)

- Cryptographic Services Security Level 2 (FMID JCPT283)

  When you order the Cryptographic Services Security Level 2 support, `GSKCMSEX` is installed as a member of the *pdsname*.SGSKLOAD PDS.

  Note: *pdsname*.SGSKLOAD is the PDS in which the Cryptographic Services Base members are installed.

- Cryptographic Services Security Level 3 (FMID JCPT271)

  When you order the Cryptographic Services Security Level 3 support, `GSKCMSUS` is installed as a member of the *pdsname*.SGSKLOAD PDS.

  Note: *pdsname*.SGSKLOAD is the PDS in which the Cryptographic Services Base members are installed.

- Japanese (FMID JCPT272)

  Contains Japanese message text files for the GSKKYMAN utility. The gskkmcmd.cat and gskkmerr.cat files are installed in the /usr/lpp/gskssl/lib/nls/msg/Ja_JP directory.

2.3.2 Encryption capabilities by FMIDs

Table 1 lists the encryption capabilities for each FMID:

Table 1. Encryption capabilities per FMID

Encryption Types/ Key Sizes	Base Security Level HCPT270	Security Level 2 JCPT283	Security Level 3 JCPT271
512 bit keys	X	X	X
1024 bit keys		X	X
1- SSL V2 RC4 US			X
2- SSL V2 RC4 EXPORT	X	X	X
3- SSL V2 RC2 US			X
4- SSL V2 RC2 EXPORT	X	X	X
6- SSL V2 DES US		X	X
7- SSL V2 Triple DES US			X
01 - SSL V3 NULL MD5	X	X	X
02 - SSL V3 NULL SHA	X	X	X
03 - SSL V3 RC4 MD5 EXPORT	X	X	X

Encryption Types/ Key Sizes	Base Security Level HCPT270	Security Level 2 JCPT283	Security Level 3 JCPT271
04 - SSL V3 RC4 MD5 US			X
05 - SSL V3 RC4 SHA US			X
06 - SSL V3 RC2 MD5 US	X	X	X
09 - SSL V3 DES SHA		X	X
0A - SSL V3 Triple DES SHA			X

2.3.3 Installation

System SSL is part of the Cryptographic Services Base element of OS/390. If you choose to install the OS/390 Release 8 Server Pack, you will not need to install the Cryptographic Services Base element separately. If you choose the OS/390 PDO, you can install the Cryptographic Services Base element using SMP/E. The OS/390 Program Directory Version 2 Release 8 contains the directions for installing the Cryptographic Services Base element using SMP/E.

2.3.4 Certificate/key management

SSL connections make use of public/private key mechanisms for authenticating each side of the SSL session and agreeing on bulk encryption keys to be used for the SSL session. To use public/private key mechanisms (termed PKI), public/private key pairs must be generated. In addition, X.509 certificates (which contain public keys) must be created or certificates must be requested, received, and managed.

System SSL supports the following two methods for managing PKI private keys and certificates:

- An OS/390 shell-based program called GSKKYMAN, which creates, fills in, and manages an OS/390 HFS file that contains PKI private keys, certificate requests, and certificates. This OS/390 HFS file is called a *key database* and, by convention, has a file extension of *.kdb*.
- The OS/390 Security Server (RACF) RACDCERT command. RACDCERT installs and maintains PKI private keys and certificates in RACF and optionally in ICSF. Refer to Chapter 3, “Digital certificate support enhancements” on page 25 for details on the RACDCERT command enhancements. RACF supports multiple PKI private keys and certificates to be managed as a group. These groups are called key rings.

  Note: RACF should be the preferred method when the application supports RACF keyrings, because it provides better security.

The System SSL application uses the keyring parameter of the gsk_init_data structure during the `gsk_initialize()` call to specify the locations of the PKI private keys and certificates to System SSL. If you are using an OS/390 HFS key database, the key database file name is passed in this parameter. If you are using a RACF key ring, the name of the key ring is passed in this parameter.

See 5.2.2.2, “SSL setup using a RACF key ring” on page 160 to learn how the LDAP server utilizes the RACF key ring support when using System SSL.

2.3.5 GSKKYMAN

The System SSL utility gskkyman is a replacement for the MKKF andIKEYMAN for OS/390 applications that make use of System SSL, like the OS/390 LDAP Server.

You can use GSKKYMAN to perform a number of required and optional tasks on the key database files. Whether a task is required or optional depends on what types of certificates the installation is using. For example, if only self-signed server certificates are used, you do not have to formulate a certificate request to be sent to an external certificate authority (CA) for approval. However, in this case SSL clients do have to import the server's self-signed certificate so that it can be verified during SSL handshake processing.

The tasks that GSKKYMAN can perform include the following:

- Creating a key database
- Creating a certificate request (and public/private key pair)
- Creating a self-signed certificate
- Exporting a certificate request to a file
- Receiving a certificate after a certificate request has been fulfilled
- Exporting a certificate to a file
- Importing a certificate from a file
- Marking a certificate as a trusted CA certificate
- Marking a certificate (and private key) as the default certificate for the key database
- Listing certificate information
- Removing a certificate (and private key) from a key database
- Removing a key database.

For an example of GSKKYMAN usage, see 9.1.1.5, “SSL setup for the firewall configuration client” on page 260.

Chapter 3. Digital certificate support enhancements

This chapter details the enhancements made to RACF elements of the OS/390 Security Server in support of digital certificates.

The RACDCERT command has been enhanced to support:

- Generation of digital certificates
  - Definition of site certificates (like `www.ibm.com`)
  - Definition of Certificate Authority (CA) certificates
- Generation of digital certificate requests
- Exporting of digital certificates
- Aggregation of digital certificates into keyrings
- Importing of digital certificates in PKCS#12 format
- Renaming a label associated with a digital certificate stored in the RACF database

A new function, called Certificate Name Filtering (CNF), is introduced in OS/390 Security Server Version 2 Release 9 and made available to Version 2 Release 8 through APAR OW40129.

We first explain the RACDCERT enhancements, which then leads into the new CNF function.

3.1 Digital certificate enhancements

Prior to OS/390 Security Server Version 2 Release 8, the RACDCERT command is only used to associate a client or personal digital certificate with a RACF user ID.

The benefit of having a certificate associated with a RACF user ID is shown in Figure 3.

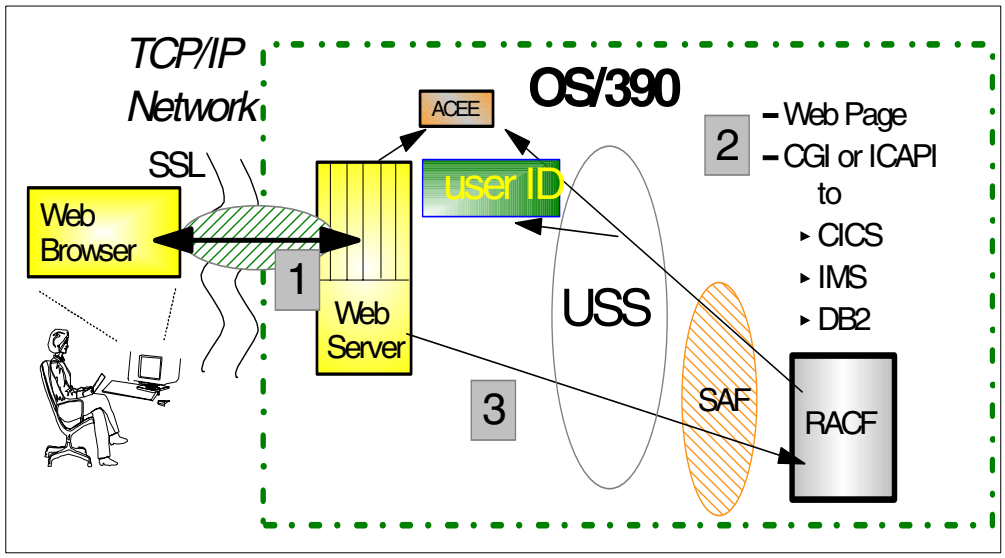


Figure 3. Websphere usage of mapping a certificate to a RACF user ID

The IBM HTTP Server for OS/390 (formerly Domino GO Webserver) is one of the first applications to use this “mapping support”. The flow shown on the figure is as follows:

1. The user authenticates himself to the HTTP server with a client certificate over a Secured Sockets Layer (SSL) secured link.
2. The user requests access to OS/390 secured resources.
3. The HTTP Server invokes RACF via UNIX System Services (USS) to build local security context (ACEE), passing the client SSL validated certificate instead of prompting for user ID and password.

Other applications on OS/390 are beginning to use this mapping support as well, like the CICS Transaction Server Version 1 Release 3 and TN3270 Server, as discussed in Chapter 10, “Enabling SSL on Telnet” on page 327.

3.1.1 RACDCERT ADD command syntax

Figure 4 shows the command syntax for the RACDCERT ADD command in OS/390 version 2 Release 8.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]
ADD(data-set-name)
         TRUST|NOTRUST
         WITHLABEL('label-name')
         PASSWORD('pkcs12-password')
         ICSF
```

Figure 4. RACDCERT command syntax

The data set name variable of the `ADD` keyword specifies that the data set with the specified name must contain the digital certificate. Each user ID can be associated with more than one digital certificate, but they must be added individually. The specified data set should contain only one digital certificate. The digital certificate must be in one of the following formats:

1. A single DER-encoded X.509 certificate.
2. A Privacy Enhanced Mail (PEM) encoded X.509 certificate. If the input is in this format, only the Originator Certificate are used.
3. One or more X.509 certificates contained within a PKCS#7 DER encoding. If the input is in this format, only the first certificate in the PKCS#7 encoding is used.
4. A Base64 encoded certificate.
5. One or more X.509 certificates and private keys contained within a PKCS#12 DER encoding. If the input is in this format, only the first user certificate and private key is used.

Note: PKCS#12 is also known as Private Information Exchange (PFX). The PFX V0.02 standard is no longer supported.

Note: A package is one or more X.509 certificates encoded under the new PKCS standard.

Care must be taken when transporting the different certificate encodings to and from an OS/390 system. The BER encoded X.509, PKCS#7, and PKCS#12 formats are binary and must be transported in their exact binary format. Do not perform any ASCII to EBCDIC translations on these formats. PEM and Base64, however, are text-based protocols and should be transported as such, meaning as text. If transporting from an ASCII system, the ASCII to EBCDIC translation must be performed for the PEM format and Base64 format certificate.

The formats, other then PKCS#12, are not always easy to come by. Usually the client will have to retrieve this format from the directory service the certificate authority provides. The certificate authority will store the digital certificate of the client in some format in a directory (LDAP). As an example we use the certificate authority Verisign Inc., which offers directory services to retrieve your digital certificates in various formats.

3.1.1.1 Retrieving your digital certificate from VeriSign Inc.

At VeriSign's home page, click **Individual Certificates**, as shown in Figure 5 on page 28.

Figure 5. VeriSign's home page

The next window presents VeriSign's so-called "Digital ID Center", as shown in Figure 6.

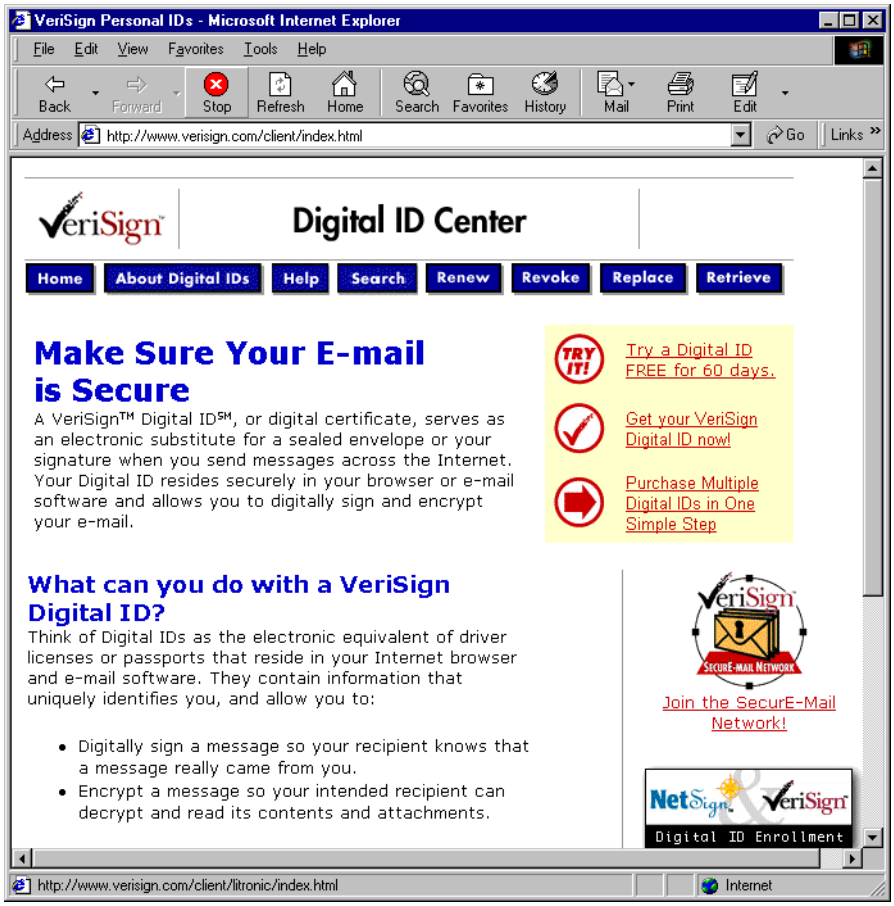


Figure 6. VeriSign's Digital ID Center window

This window gives you the option to search for your digital certificates. Click **Search** to perform a search for your certificates.

The next window, shown in Figure 7 on page 30, gives you the ability to search for your digital certificates based on a search criteria. We used the e-mail address we entered when we requested our digital certificates.

Figure 7. VeriSign's Digital ID Center search window

We entered our e-mail address (`graaff@us.ibm.com`) in the search criteria field, as shown in Figure 7. You can restrict your search if you want to, for instance by retrieving only valid digital certificates or expired digital certificates. Next, click **Search** to perform a directory search of your digital certificates.

Figure 8 on page 31 shows the search results of our query. As you can see, all our retrieved digital certificates are expired. This is not important for our example, because we only want to show you how to retrieve a digital certificate in a format that is supported by the RACDCERT command.

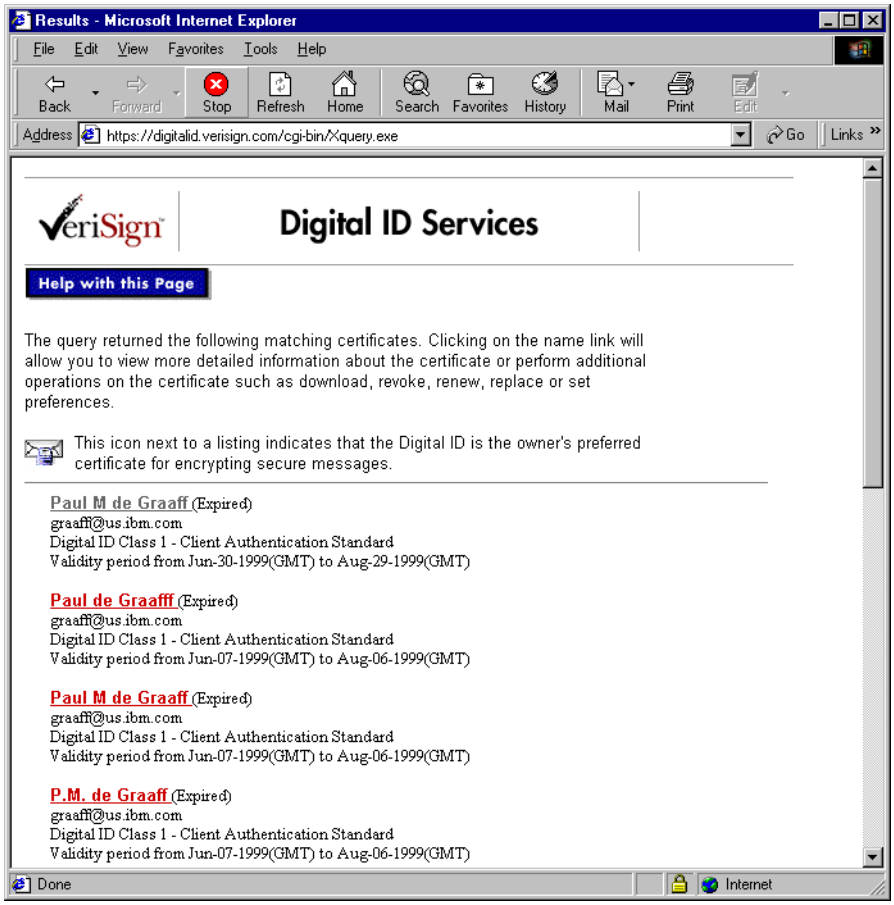


Figure 8. Digital ID Center search results window

Next, we select the certificate we want to retrieve by clicking on the name field (Paul M de Graaff in our example). The window shown in Figure 9 on page 32 displays the selected digital certificates in more detail and allows the digital certificate to be downloaded (one of the functions allowed).

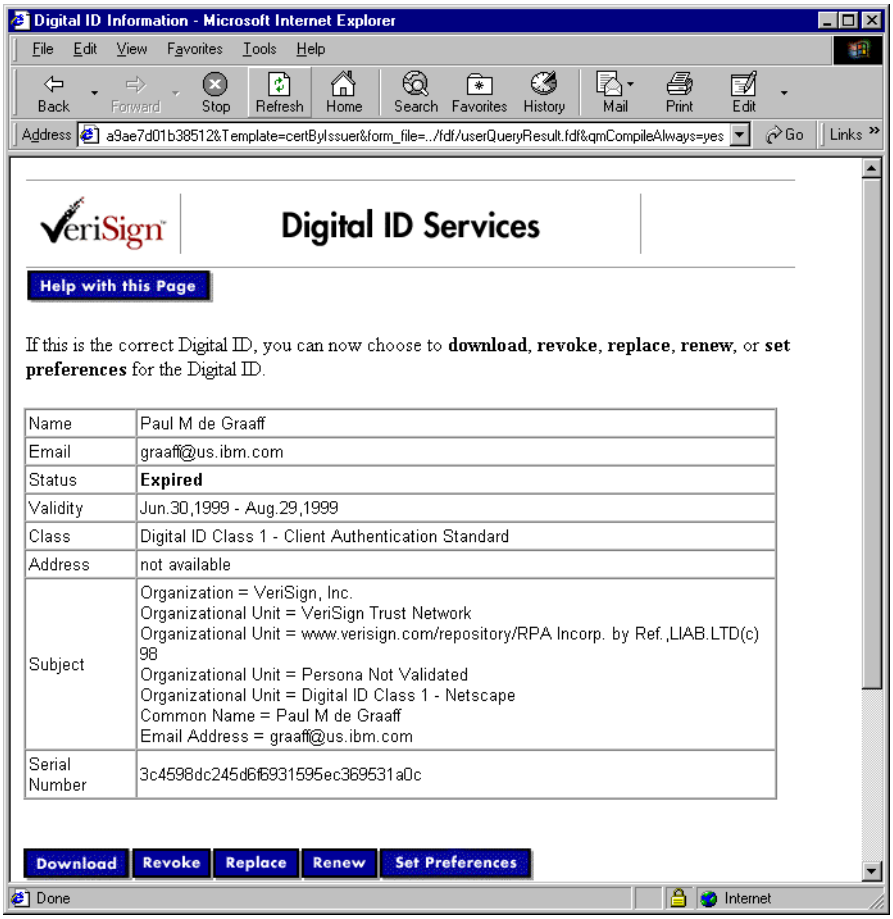


Figure 9. Digital ID Services window

Click **Download** and you are able to download the digital certificates in a variety of formats, as shown in Figure 10 on page 33.

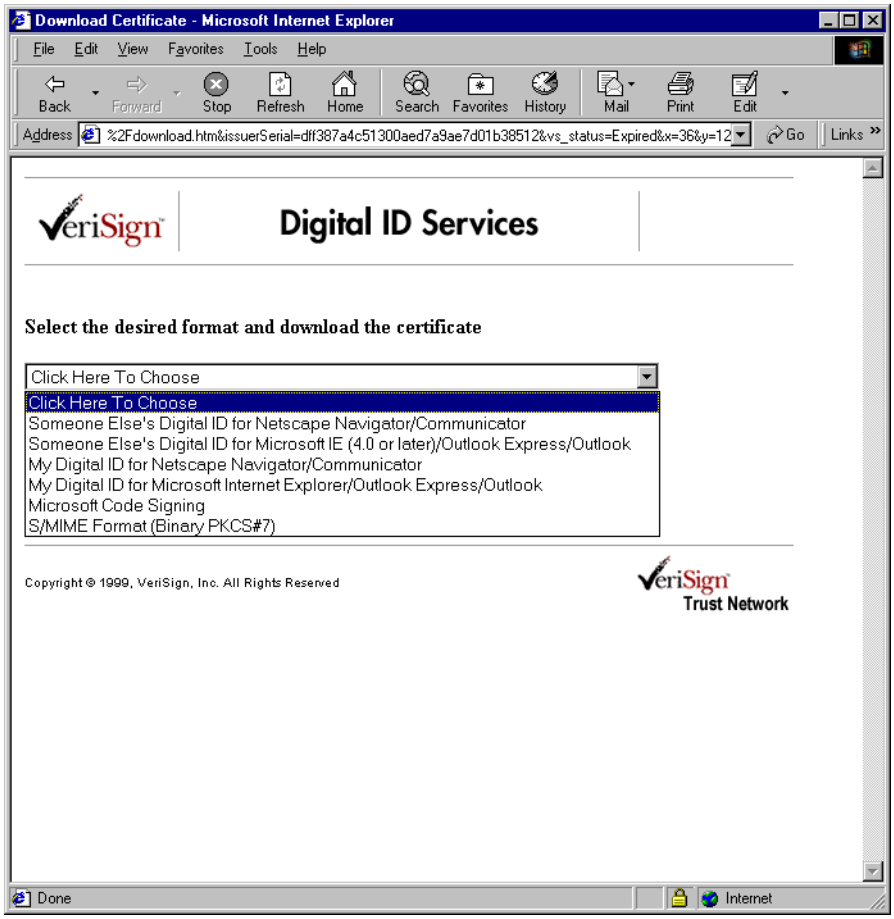


Figure 10. Digital ID Services download window

Select the **S/MIME Format (Binary PKCS#7)** option and click **Download this certificate** to save the certificate to your workstation, as shown in Figure 11.

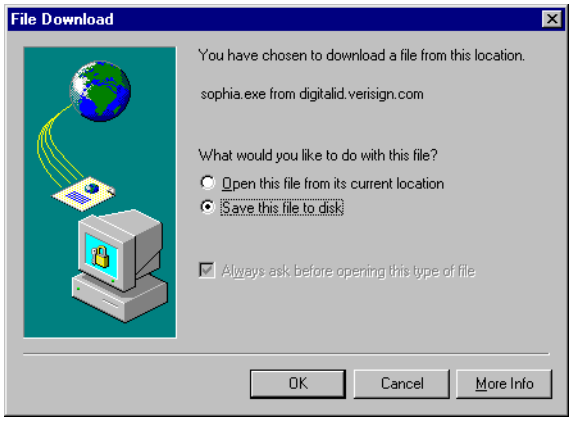


Figure 11. Download file window

Select the **Save this file to disk** option and click **OK**. You can then select the directory where you want to store the file. The next step is to upload the file in binary format to an OS/390 data set for RACDCERT to process.

We used the IND$FILE transfer process to transfer the file to an OS/390 file called graaff.paul.p7cbin. Next we issued the RACDCERT command to associate user ID graaff with the digital certificate stored in the file.

```
racdcert add(paul.p7cbin)
```

Note: We did not specify id(*userid*) on the RACDCERT command, which means the certificate is associated with the user ID executing the RACDCERT command.

We received the following error message:

```
 IRRD113I The certificate that you are adding is expired. The certificate is
added with NOTRUST status.
```

This error message is correct because the digital certificate was indeed expired.

We can now list the association between the RACF user ID graaff and the digital certificate we just installed. The RACDCERT command creates a profile in the DIGTCERT class. If you do not specify the LABEL parameter of the RACDCERT command, a label will be generated by RACF. To list the digital certificate(s) associated with user ID graaff, we issue the following command:

```
RACDCERT LIST ID(GRAAFF)

Label: LABEL00000004
 Status: NOTRUST
 Start Date: 1999/06/29 19:00:00
 End Date:   1999/08/29 18:59:59
 Serial Number:
      >3C4598DC245D6F6931595EC369531A0C<
 Issuer's Name:
      >CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU<
      >=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=Ver<
      >iSign Trust Network.O=VeriSign, Inc.<
 Subject's Name:
      >graaff@us.ibm.com.CN=Paul M de Graaff.OU=Digital ID Class 1 - Netscap<
      >e.OU=Persona Not Validated.OU=www.verisign.com/repository/RPA Incorp.<
      > by Ref.,LIAB.LTD(c)98.OU=VeriSign Trust Network.O=VeriSign, Inc.<
 Private Key Type: None
 Ring Associations:
 *** No rings associated ***
```

Figure 12. RACDCERT LIST output listing

Note: Notice there is no private key information, because PKCS#7 format does *not* contain the private key.

3.1.2 RACDCERT ADD enhancement to support PKCS#12

The RACDCERT ADD command is enhanced in OS/390 Security Server Version 2 Release 8 to support the import of a digital certificate stored in a PKCS#12 format. The PKCS#12 format is supported by the most popular browsers (Netscape Navigator and Micrsoft's Internet Explorer).

3.1.2.1 Internet Explorer

You can export digital certificates by using the Content tab from the Internet Options window, as shown in Figure 13.

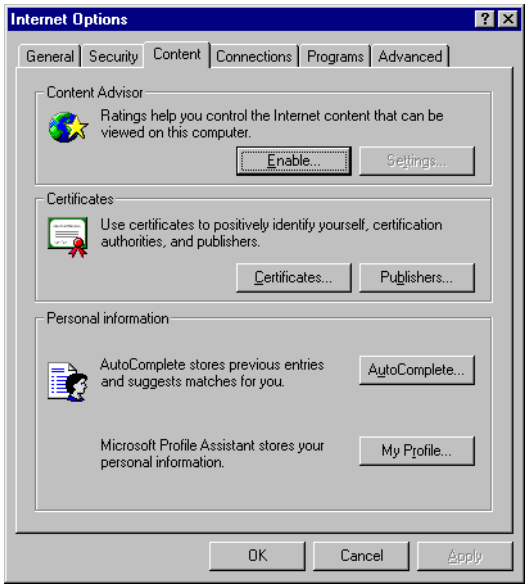


Figure 13. Internet options window

In this example we are using Internet Explorer 5. When we press **Certificates...** , the Certificate Manager window is displayed, as shown in Figure 14.

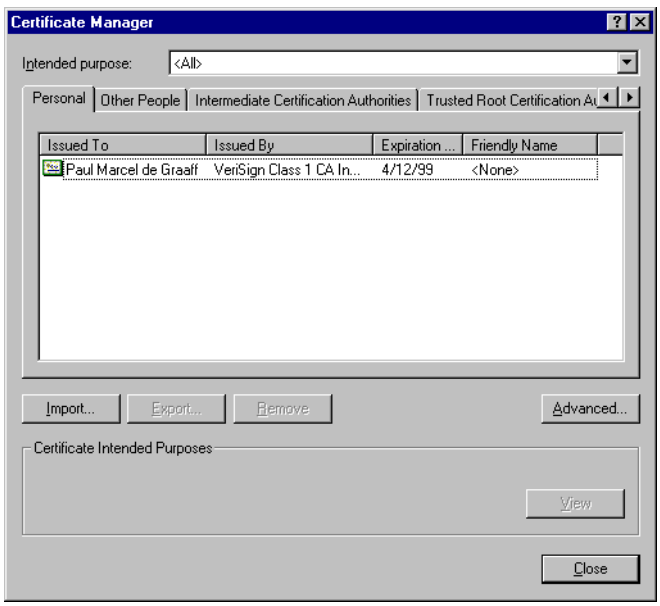


Figure 14. Certificate Manager window

Select the digital certificate you want to export by clicking it once, then click **Export**. The Certificate Manager Export Wizard window is displayed, as shown in Figure 15, and takes you through the export process.

Figure 15. Certificate Manager Export Wizard window

The next step is to tell the wizard whether you want to export the digital certificate with the private key or without. In our example we export the digital certificate with the private key, as shown in Figure 16.

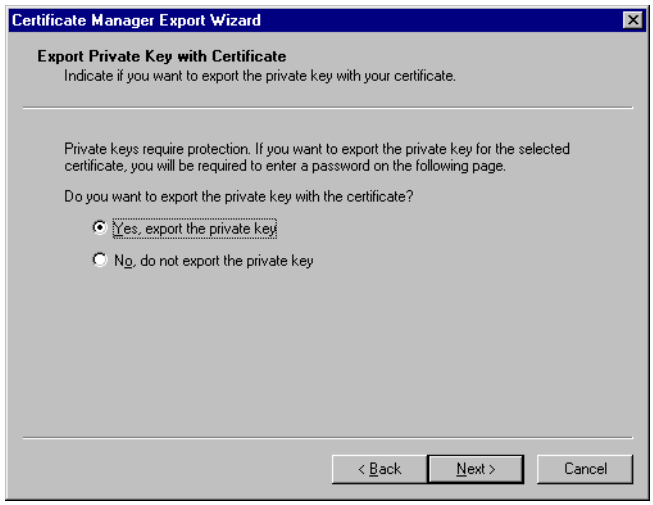


Figure 16. Certificate Manager Export Wizard export window

By selecting the **Yes, export the private key** option and clicking **Next**, the Certificate Export File window will be presented, as show in Figure 17 on page 37.

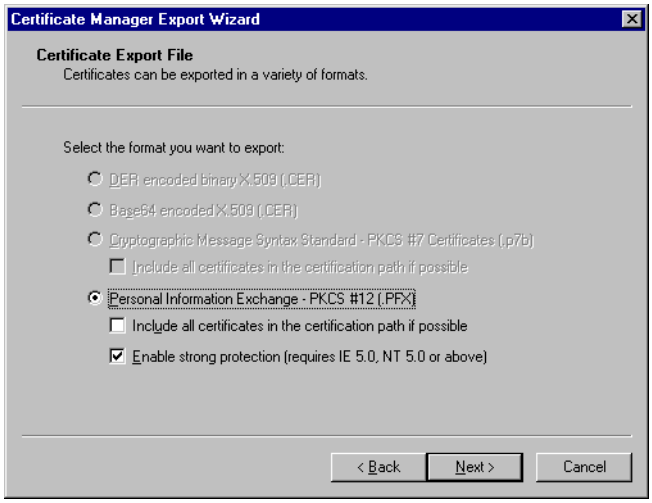


Figure 17. Certificate Export File window

The **Enable strong protection** option is selected; this will secure the exported digital certificate file with a password to prevent disclosure of the private key. To continue, we click **Next** and the window, as shown in Figure 18, will allow us to enter a password to secure the file that contains the exported digital certificate.

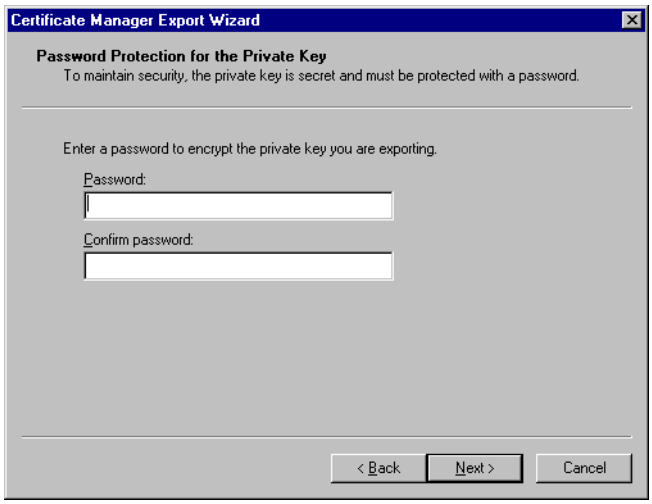


Figure 18. Password Protection window

After entering the password twice, click **Next** to go to the Export file name window, shown in Figure 19 on page 38, to select where you want to store the file containing your digital certificate.

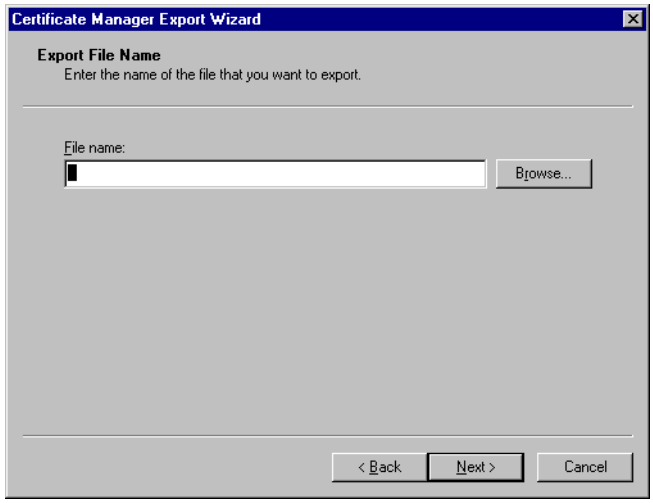


Figure 19. Export File Name windows

To select the location for the file, click **Browse** to locate the directory where you want the file stored.

In our example, we used `D:\Download\certificates` as shown in Figure 20.

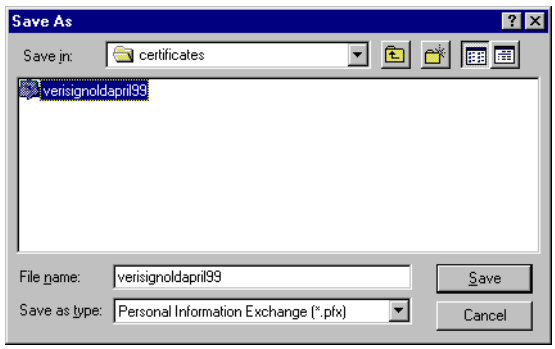


Figure 20. Save as window

Note: Notice the file format of the exported digital certificate, Personal Information Exchange (PFX). This is through a PKCS#12 file format.

When you select the directory and file name, click **Next** to complete the export function, as shown in Figure 21 on page 39.

Figure 21. Certificate Management Export Wizard complete window

Click **Finish** to complete the export of your digital certificate.

The next step is to upload the file in binary format to an OS/390 data set for RACDCERT to process. We used the IND$FILE transfer process to transfer the file to an OS/390 file called graaff.paul.pfxbin. Next we issued the RACDCERT command to associate user ID GRAAFF with the digital certificate stored in the file.

```
racdcert add(paul.pfxbin) id(graaff)
```

Figure 22. RACDCERT ADD example

Note: If we had not specified id(graaff) on the RACDCERT command, the certificate would have been associated with the user ID executing the RACDCERT command.

We received the following error message:

```
IRRD127I The data set contains a PKCS12 encrypted certificate. The PASSWORD
keyword must be specified to process the certificate. The certificate is not
processed.
```

We forgot to enter the password, that we entered earlier when we exported the file. We reentered the RACDCERT with the PASSWORD parameter as shown next.

```
racdcert add(paul.pfxbin) password('paul') id(graaff)
```

We still receive an error message, as shown next, but that was expected because the digital certificate we added is expired.

```
 IRRD113I The certificate that you are adding is expired. The certificate is
added with NOTRUST status.
```

We can now list the association between the RACF user ID GRAAFF and the digital certificate we just installed. The RACDCERT command creates a profile in the DIGTCERT class. If you do not specify the LABEL parameter of the RACDCERT command, a label will be generated by RACF. To list the digital certificates associated with user ID GRAAFF, we issue the following command:

```
RACDCERT LIST ID(GRAAFF)

Label: a523f796-c1f1-11d2-b8d3-64a83100
Status: NOTRUST
Start Date: 1999/02/10 19:00:00
End Date:   1999/04/12 18:59:59
Serial Number:
     >5FC2860A2C1629FB724E249285F206A7<
Issuer's Name:
     >CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU<
     >=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=Ver<
     >iSign Trust Network.O=VeriSign, Inc.<
Subject's Name:
     >graaff@us.ibm.com.CN=Paul Marcel de Graaff.OU=Digital ID Class 1 - Mi<
     >crosoft.OU=Persona Not Validated.OU=www.verisign.com/repository/RPA I<
     >ncorp. by Ref.,LIAB.LTD(c)98.OU=VeriSign Trust Network.O=VeriSign, In<
      >c.<
 Private Key Type: Non-ICSF
 Private Key Size: 512
 Ring Associations:
 *** No rings associated ***
```

Note: Notice the private key information is shown because PKCS#12 format *does* contain the private key.

Internet Explorer Version 5 does not *only* allow for export of digital certificates with a private key (e.g. PKCS#12), but also allows export without a private key. These formats are usually used by the directory service of a certificate authority as discussed in 3.1.1.1, “Retrieving your digital certificate from VeriSign Inc.” on page 27. The export formats supported by the Certificate Manager Export Wizard are:

- DER encoded binary X.509 (filename.CER)
- Base64 encoded X.509 (filename.CER)
- PKCS#7 - Cryptographic Message Syntax Standard (filename.P7B)

These formats are all supported by the RACDCERT command as well. Just remember that files in PKCS#7 and DER-encoded format need to be uploaded in binary format to the host (OS/390) and the Base64-encoded format, which is a text-based protocol, in text format (ASCII).

3.1.2.2 Netscape Navigator

Netscape Navigator or Communicator supports export of digital certificates in a PKCS#12 format as well. From the main window, select the **Security** option. This presents the security window, as shown in Figure 23 on page 41.

Figure 23. Netscape Navigator security window

The Security window shows a Certificates option, where you can select Yours, Web Sites, or Signers digital certificates. To perform operations (like export) on your digital certificates, select **Yours**. The Security window changes, and shows your digital certificates stored in the certificate database, as shown in Figure 24 on page 42.

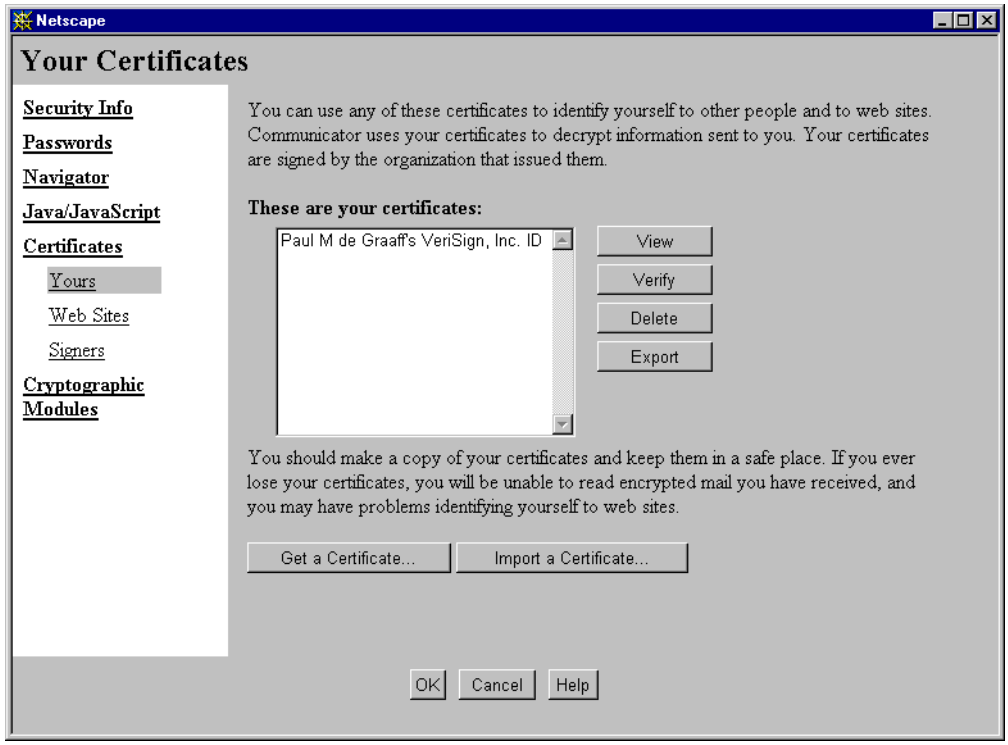


Figure 24. Netscape Navigator Certificates - Yours

To export a certificate, select it by clicking once on it and then clicking **Export**.

Note: In our example, only one digital certificate is stored in the certificate database.

After clicking **Export**, you are presented with a password entry dialog, as shown in Figure 25.

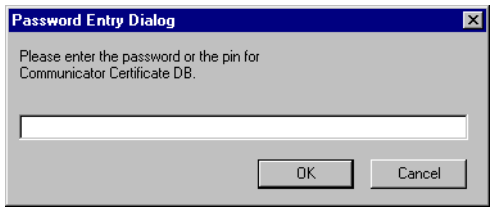


Figure 25. Password entry dialog

Your digital certificates are secured in the certificate database by means of a password. The first time you installed a digital certificate into Netscape Navigator, you were asked to provide a password to secure the certificate database.

After entering the correct password and clicking **OK**, you are presented with another password entry dialog window, as shown in Figure 26 on page 43.

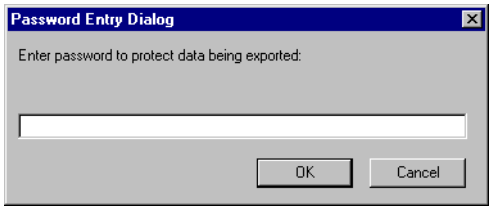


Figure 26. Password entry dialog

This password secures the actual output file, so your private key can not be compromised. Next enter a password to secure the file. After you click **OK**, you have to reconfirm the password on the next dialog window presented, as shown in Figure 27.

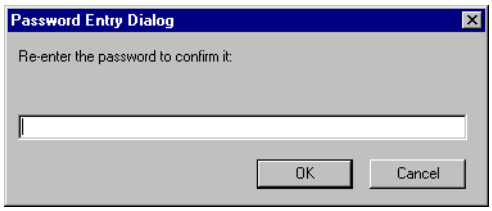


Figure 27. Password entry dialog

After you confirm the password, you are able to specify the output location of the file and the file name, as shown in Figure 28.

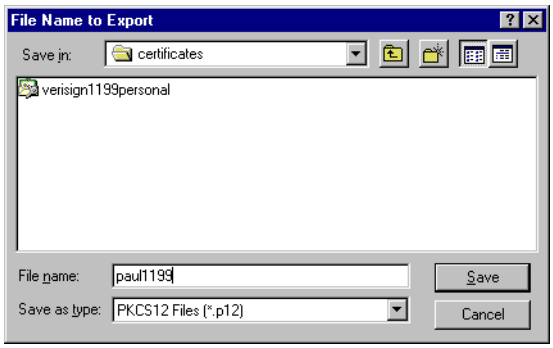


Figure 28. File Name in export

Note: Notice the file format of the exported digital certificate is PKCS#12 (filename.P12). This differs from Internet Explorer, which used PFX as the file format.

The export function ends with a successful termination message, as shown in Figure 29.

Figure 29. Netscape Navigator successful export message

3.1.3 RACDCERT ADD enhancement to support site certificates

The ADD keyword of the RACDCERT command supports *site* digital certificates in its processing. Site digital certificates are used to identify a trusted site (server), for example a Web site (Web server). Digital certificates are exchanged between the server and the client (optionally between the client and the server) when using the Secured Socket Layer (SSL) protocol for secure communications between a web browser and a web server. The site's digital certificate identifies the web server (like www.ibm.com).

The RACDCERT ADD command has been enhanced to support site certificates, as the command syntax shows in Figure 30.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]
ADD(data-set-name)
         TRUST|NOTRUST
         WITHLABEL('label-name')
         PASSWORD('pkcs12-password')
         ICSF
```

Figure 30. RACDCERT ADD command syntax for SITE Certificates

For example, we are going to add a site certificate that we used for our Telnet server (see 10.1.3, "Implementation scenario" on page 332). This site certificate was generated with the GSKKYMAN utility, supplied with System SSL. The steps necessary to transfer the site certificate from our Telnet keyring to RACF are:

1. Export the site certificate out of the keyring to a file.
2. Copy the file from USS to an OS/390 data set.
3. Execute the RACDCERT ADD for the site certificate stored in the OS/390 data set.

We use the GSKKYMAN utility to export the site certificate, as shown in Figure 31.

```
GRAAFF @ SC57:/u/graaff>gskkyman


            IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 2
Enter key database name or press ENTER for "key.kdb": telnet.kdb
Enter password for the key database.......>


 ===>
```

Figure 31. GSKKYMAN startup screen

Next we open the Telnet keyring (in our example, telnet.kdb) and enter the password to open the keyring. The screen, as shown in Figure 32, is displayed.

```
           Key database menu

Current key database is /u/graaff/telnet.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu):
 ===> 9
```

Figure 32. GSKKYMAN option menu

When we select option **9**, **Export keys**, we are able to export the site certificate to a PKCS#12 file, as shown in Figure 33.

```
 Export Key Menu

Current key database is /u/graaff/telnet.kdb

    1  - Export keys to another key database
    2  - Export keys to a PKCS12 file

    0  - Exit program

Enter option number (or press ENTER to return to the parent menu):
 ===> 2
```

Figure 33. GSKKYMAN: Export key menu

When we select option **2**, **Export keys to a PKCS12 file**, we are able to select the site certificate we want to export, as shown in Figure 34 on page 46.

```
           Key and certificate lists of the export key database

Key database name is /u/graaff/telnet.kdb

Please choose one of the following keys to work with.

    1  - verisign telnet server wtsc57
    2  - Verisign Trust CA
    3  - WTSC57 SELF SIGNED TELNET SERVER
    4  - Integrion Certification Authority Root
    5  - IBM World Registry Certification Authority
    6  - Thawte Personal Premium CA
    7  - Thawte Personal Freemail CA
    8  - Thawte Personal Basic CA
    9  - Thawte Premium Server CA

Enter the numbers of the keys that you want to export (for example, 2,5)
 or press ENTER for more labels:
 ===> 3
Enter output PKCS12 file name or press ENTER for "key.p12": telnetself.p12
Enter a password to protect the output PKCS12 file:
Enter password again for verification.....>

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
 ===> 1
```

Figure 34. GSKKYMAN: Export key selection menu

The site certificate we want to export is key number 3 (`WTSC57 SELF SIGNED TELNET SERVER`). To export this key, we entered **3**. Next we are prompted for the file name that will contain our site certificate. In our example (see Figure 34), we used `telnetself.p12` as our file name. Next the PKCS#12 file is secured with a password, to prevent potential disclosure of the private key.

The next step is to copy the file from UNIX System Services (USS) to an OS/390 data set. We can use the TSO `OGET` command for that, as shown Figure 35.

```
OGET '/u/graaff/telnetself.p12' 'graaff.telnets.p12' BINARY
```

Figure 35. OGET command example

Note: Remember to specify the binary option because PKCS#12 is a binary format.

The last step is to actually install the site certificate into the RACF database with the RACDCERT ADD command, as shown in Figure 36.

```
racdcert site add(telnets.p12) password('telnet') withlabel('WTSC57 TELNET SERVER'

IRRD119I Certificate Authority not defined to RACF.  Certificate added with
TRUST status.
```

Figure 36. RACDCERT ADD command example to add a site certificate

Note: The informational message appears because this site certificate is a self-signed certificate.

We can list the site certificate, using the RACDCERT LIST SITE command as shown in Figure 37.

```
RACDCERT LIST SITE

Digital certificate information for user irrsitec:

Label: WTSC57 TELNET SERVER
 Status: TRUST
 Start Date: 1999/06/24 15:07:40
 End Date:   2000/06/24 15:07:40
 Serial Number:
      >EE81DC7D516F1B1F<
 Issuer's Name:
      >CN=wtsc57.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
 Subject's Name:
      >CN=wtsc57.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
 Private Key Type: Non-ICSF
 Private Key Size: 1024
 Ring Associations:
 *** No rings associated ***
```

Figure 37. RACDCERT LIST SITE command output

Note: Notice the private key information is shown because PKCS#12 format *does* contain the private key.

3.1.4 RACDCERT ADD enhancement to support CA certificates

The ADD keyword of the RACDCERT command supports *Certificate Authority* (CA) digital certificates in its processing. CA certificates are used to sign other certificates, like personal, site or other CA certificates.

The command syntax in Figure 38 shows the enhancement made to the RACDCERT ADD command to support CA certificates.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]
ADD(data-set-name)
         TRUST|NOTRUST
         WITHLABEL('label-name')
         PASSWORD('pkcs12-password')
         ICSF
```

Figure 38. RACDCERT ADD command syntax for CA certificates

A list of well-known CA certificates (like VeriSign Inc.) is stored in a USS file called a *keyring*. RACF support for keyrings is discussed in 3.2.1, “RACDCERT ADDRING: Creating a key ring” on page 78, but we need to introduce the term keyring here.

Keyrings are created with USS utilities, like MKKF, IKEYMAN, or GSKKYMAN. When these utilities are used to create a keyring, a list of well-known CA certificates is added to the keyring. These utilities also allow you to list all the trusted CAs. Figure 39 on page 48 shows a list of trusted CAs from the keyring (telnet.kdb) we used earlier in our examples.

```
           Trust CA certificate list

Key database name is /u/graaff/telnet.kdb

Please choose one of the following keys to work with.

    1  - verisign telnet server wtsc57
    2  - Verisign Trust CA
    3  - WTSC57 SELF SIGNED TELNET SERVER
    4  - Integrion Certification Authority Root
    5  - IBM World Registry Certification Authority
    6  - Thawte Personal Premium CA
    7  - Thawte Personal Freemail CA
    8  - Thawte Personal Basic CA
    9  - Thawte Premium Server CA

Enter a key number or press ENTER for more labels:
 ===>
```

Figure 39. Trusted CA certificate list

The support in RACF for key rings and CA certificates is different in that a CA certificate is *associated* with a key ring. A CA certificate that needs to be installed in a RACF keyring is first installed into the RACF database using the RACDCERT ADD command, and then associated with a RACF key ring using the RACDCERT CONNECT command (see 3.2.2, “RACDCERT CONNECT: Install a certificate in a key ring” on page 79).

As we discuss in 3.2.1, “RACDCERT ADDRING: Creating a key ring” on page 78, when a RACF key ring is created, no list of well-known CA certificates is added to the key ring, unlike with the MKKF, IKEYMAN or GSKKYMAN utilities. These well-known certificates are in the RACF database, but still require an association with a RACF keyring to be performed.

To see a list of the well-known CA certificates that are in the RACF database, issue the RACF command, as shown in Figure 40 on page 49.

```
racdcert certauth list

Digital certificate information for user irrcerta:

Label: Verisign Class 3 Primary CA
 Status: NOTRUST
 Start Date: 1996/01/28 19:00:00
 End Date:   2004/01/07 18:59:59
 Serial Number:
      >00E49EFDF33AE80ECFA5113E19A4240232<
 Issuer's Name:
      >OU=Class 3 Public Primary Certification Authority.O=VeriSign, Inc..C=<
      >US<
 Subject's Name:
      >OU=Class 3 Public Primary Certification Authority.O=VeriSign, Inc..C=<
      >US<
 Private Key Type: None
 Ring Associations:
 *** No rings associated ***

 Label: Verisign Class 2 Primary CA
 Status: NOTRUST
 Start Date: 1996/01/28 19:00:00
 End Date:   2004/01/07 18:59:59
 Serial Number:
      >00BA5AC94C053B92D6A7B6DF4ED053920D<
 Issuer's Name:
>OU=Class 2 Public Primary Certification Authority.O=VeriSign, Inc..C=<
      >US<
 Subject's Name:
      >OU=Class 2 Public Primary Certification Authority.O=VeriSign, Inc..C=<
      >US<
 Private Key Type: None
 Ring Associations:
 *** No rings associated ***
```

Figure 40. RACDCERT CERTAUTH LIST command output

Note: The list in Figure 40 is not a complete list of all well-known CA certificates stored in the RACF database, merely a snapshot. For a full list of the CA certificates available, see Appendix D, “RACF list of certificate authority certificates” on page 413.

To add a CA certificate to RACF, we first need to get a CA certificate. We can use our web browser for that purpose. Both popular browsers have a list of well-known CA certificates, as shown in Figure 41 and Figure 42 on page 50.

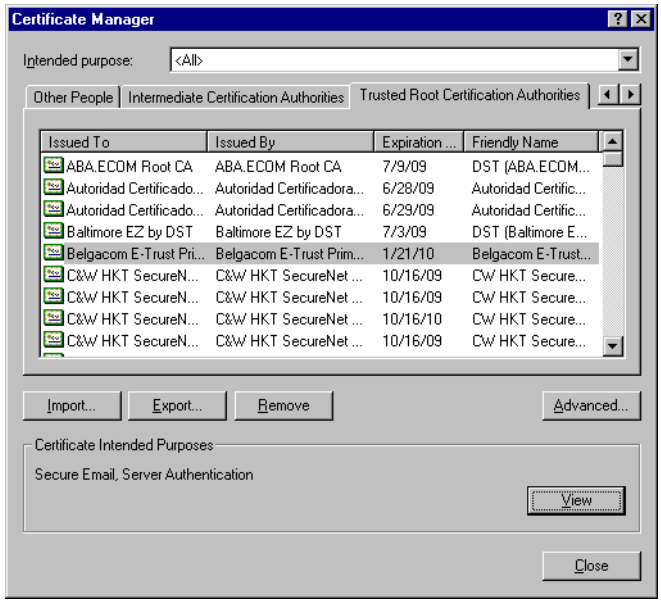


Figure 41. Internet Explorer: Trusted Root Certification Authorities

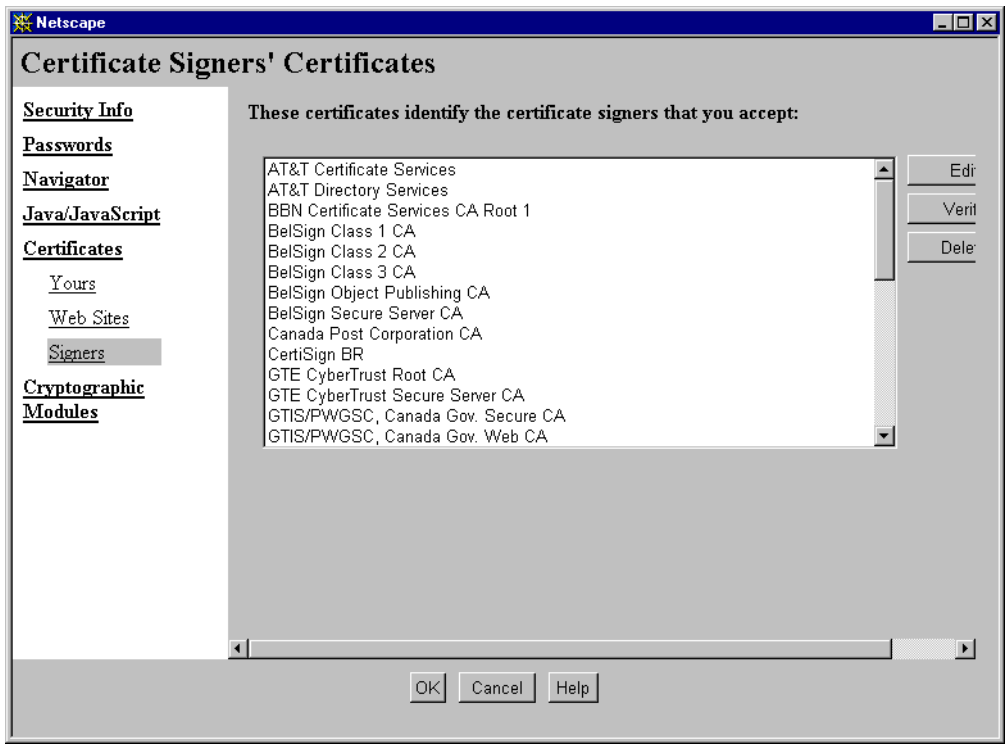


Figure 42. Netscape Navigator: Certificate Signers' Certificates

Internet Explorer offers the capability to export a CA certificate. For our example, we export the CA certificate of Belgacom. Click **Export** on the Certificate Manager window, as shown in Figure 41 on page 50.

Figure 43. Certificate Manager Export Wizard window

On the Certificate Manager Export wizard window, click **Next** as shown in Figure 43.

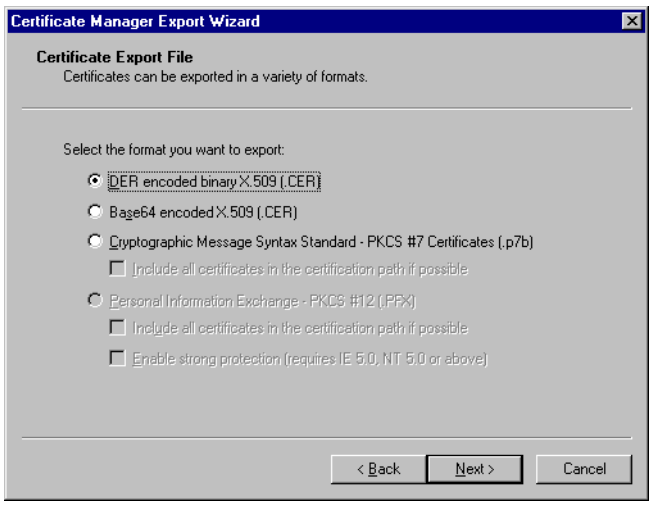


Figure 44. Certificate Export File window

Figure 44 shows the Certificate Export File window, where we can specify what type of export format (DER, Base64 or PKCS#7) we want. In our example, we used the DER format. Click **Next** to continue the export process.

Figure 45 on page 52 shows the output file location window for the exported certificate.

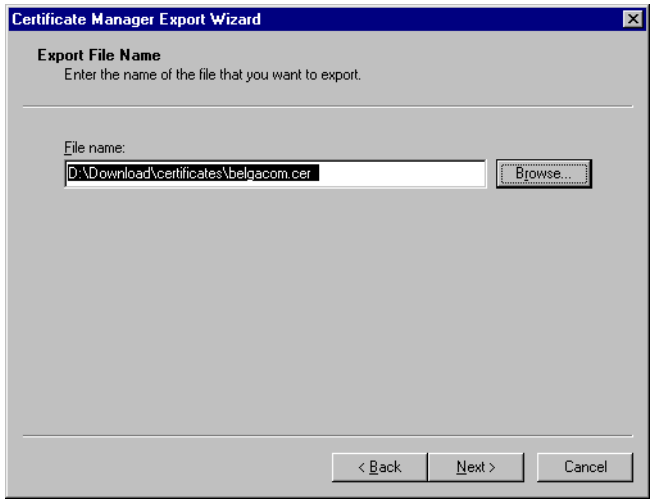


Figure 45. Export File Name window

We chose `d:\download\certificates\belgacom.cer` as our file name. Click **Next** to continue the export process. As shown in Figure 46, the export process is now completed.

Figure 46. Certificate Manager Export Wizard completion window

Click **Finish**; a confirmation message is presented indicating the successful export, as shown in Figure 47.

Figure 47. Certificate Export confirmation message

We now have to upload the file `belgacom.cer` to the host to process it with the RACDCERT command. We issued the following RACF command to add the CA certificate to the RACF database:

```
racdcert certauth add(belgacom.cerbin) withlabel('BELGACOM CA')

IRRD119I Certificate Authority not defined to RACF.  Certificate added with TRU
ST status.
```

Figure 48. RACDCERT CERTAUTH ADD command example

We are now able to list the Belgacom certificate using the following RACF command:

```
racdcert certauth list(label('BELGACOM CA'))

Digital certificate information for user irrcerta:

Label: BELGACOM CA
 Status: TRUST
 Start Date: 1998/11/04 08:04:39
 End Date:   2010/01/21 08:04:39
 Serial Number:
      >378767AC<
 Issuer's Name:
      >info@e-trust.be.CN=Belgacom E-Trust Primary CA.OU=MTM.O=Belgacom.C=be<
 Subject's Name:
      >info@e-trust.be.CN=Belgacom E-Trust Primary CA.OU=MTM.O=Belgacom.C=be<
 Private Key Type: None
 Ring Associations:
 *** No rings associated ***
```

Figure 49. RACDCERT CERTAUTH LIST command output listing

3.1.5 RACDCERT CHECKCERT enhancement

The `CHECKCERT` keyword of the RACDCERT command has been enhanced to support digital certificates in a PKCS#12 format.

The `CHECKCERT` keyword also supports the evaluation of site certificates and certificate authority certificates. It indicates if the certificate is defined and to whom it is defined after checking the resource `IRR.DIGTCERT.LIST` in the FACILITY class. READ authority is required if the certificate is associated with the user issuing the command. UPDATE authority is required if the certificate is associated with a user other than the issuer of the command. CONTROL authority is required if the certificate is a certificate authority or a site certificate.

The `CHECKCERT` function can be used on the same set of certificate packages that is allowed by RACDCERT ADD.

First we show an example of the CHECKCERT function, to CHECKCERT a certificate authority certificate. We use the same data set that we used earlier in our RACDCERT ADD example of a certificate authority, as shown Figure 48.

```
racdcert checkcert(belgacom.cerbin)

Digital certificate information for user irrcerta:

  Label: BELGACOM CA
  Status: TRUST
  Start Date: 1998/11/04 08:04:39
  End Date:   2010/01/21 08:04:39
  Serial Number:
       >378767AC<
  Issuer's Name:
        >info@e-trust.be.CN=Belgacom E-Trust Primary CA.OU=MIM.O=Belgacom.C=be<
  Subject's Name:
        >info@e-trust.be.CN=Belgacom E-Trust Primary CA.OU=MIM.O=Belgacom.C=be<
  Private Key Type: None
```

Figure 50. RACDCERT CHECKCERT: Example 1

Note: In this example we see that the certificate authority certificate is already installed.

Next we show an example of using the CHECKCERT function against a PKCS#12 formatted file, contain a user's digital certificate. Again we use the same data set we used in our RACDCERT ADD example of a user's digital certificate, as shown in Figure 22 on page 39.

```
racdcert checkcert(paul.pfxbin)

IRRD127I The data set contains a PKCS12 encrypted certificate.  The PASSWORD
keyword must be specified to process the certificate.  The certificate is not
processed.
```

Figure 51. RACDCERT CHECKCERT: Example 2

Figure 51 shows that the CHECKCERT function realizes that the data set contains a digital certificate in a PKCS#12 format and a password is required. So we issue the RACDCERT CHECKCERT command again with the PASSWORD keyword, as shown in Figure 52 on page 55.

```
racdcert checkcert(paul.pfxbin) password('paul')

Digital certificate information for user GRAAFF:

  Label: a523f796-c1f1-11d2-b8d3-64a83100
  Status: NOTRUST
  Start Date: 1999/02/10 19:00:00
  End Date:   1999/04/12 18:59:59
  Serial Number:
       >5FC2860A2C1629FB724E249285F206A7<
Issuer's Name:
     >CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU<
     >=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=Ver<
     >iSign Trust Network.O=VeriSign, Inc.<
Subject's Name:
     >graaff@us.ibm.com.CN=Paul Marcel de Graaff.OU=Digital ID Class 1 - Mi<
     >crosoft.OU=Persona Not Validated.OU=www.verisign.com/repository/RPA I<
     >ncorp. by Ref.,LIAB.LTD(c)98.OU=VeriSign Trust Network.O=VeriSign, In<
     >c.<
Private Key Type: Non-ICSF
Private Key Size: 512
```

Figure 52. RACDCERT CHECKCERT: Example 2

3.1.6 RACDCERT GENCERT: Generating a digital certificate

You can now use RACF to create, register, store, and administer digital certificates and their associated private keys, and to build certificate requests that can be sent to a certificate authority for signing. Digital certificates are managed in RACF using the RACDCERT command or by using an application that invokes the R_datalib callable service (IRRSDL00) or the initACEE callable service (IRRSIA00). The R_datalib callable service provides an application programming interface to the Common Data Security Architecture (CDSA) data library functions, and is used by SSL and System SSL to establish secure sessions between servers. The initACEE callable service can be used to manage digital certificates for RACF-authenticated users.

The GENCERT keyword of the RACDCERT command is used to create digital certificates and digital certificate requests. The command syntax for the GENCERT keyword is shown in Figure 53 on page 56.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]

  GENCERT[(request-data-set-name)]
      SUBJECTSDN([CN('common-name')]
                 [T('title')]
                 [OU('organizational-unit-name1'
                         [,'organizational-unit-name2',...]
                          )]
                 [O('organization-name')]
                 [L('locality')]
                 [SP('state-or-province')]
                 [C('country')] )]
      [SIZE(key-size)]
      [[NOTBEFORE([DATE(yyyy-mm-dd)]       [TIME(hh:mm:ss)])]
      [NOTAFTER([DATE(yyyy-mm-dd)]         [TIME(hh:mm:ss)])]
      [WITHLABEL('label-name')]
      [SIGNWITH([CERTAUTH|SITE] LABEL('label-name'))]
      [ICSF]
```

Figure 53. RACDCERT GENCERT syntax

The command syntax shows you can generate digital certificates for an individual user (ID), a site certificate (SITE), or a certificate authority (CERTAUTH).

The next examples show the usage of the GENCERT keyword of the RACDCERT command to generate certificates.

3.1.6.1 RACDCERT GENCERT example: Generate a certificate

We first show an example of how to generate a certificate that we may use for a server on OS/390 that can utilize certificates stored in the RACF database.

```
racdcert gencert subjectsdn(cn('wtsc57.itso.ibm.com') OU('ITSO') O('IBM')
L('Poughkeepsie') SP('New York') C('US'))
```

Figure 54. RACDCERT GENCERT: Example 1

Figure 54 shows an example of how to generate a certificate that is related to a user ID, in this case the user ID executing the RACDCERT command. Basically, most of the keywords of the RACDCERT GENCERT command example are left to default. When we perform a list of the certificate we created in our example, the defaults show, as shown in the RACDCERT LIST command output in Figure 55 on page 57.

```
racdcert list

Label: LABEL00000005
Status: TRUST
Start Date: 2000/02/01 00:00:00
End Date:   2001/02/01 23:59:59
Serial Number:
     >00<
Issuer's Name:
     >CN=wtscicf.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Subject's Name:
     >CN=wtscicf.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***
```

Figure 55. RACDCERT LIST of example certificate

The following keywords of the RACDCERT GENCERT command were not used; the values shown in bold in Figure 55, are the default values:

SIZE This specifies the key size of the private key. The default key size depends on the U.S. export regulations for your country. In the U.S. it is 1024.

WITHLABEL This allows a label to be specified for the digital certificate. The default value is *label* and a sequence number. LABEL00000005 was the default value in our example.

NOTBEFORE This indicates the start date and time when the digital certificate is first valid. The default value is the current date. As shown in Figure 55, the start date indicates the date the command was issued.

NOTAFTER This indicates the end date and time when the digital certificate expires. The default value is one year from the current date. As shown in Figure 55, the end date indicates a date that is a year from the date the command was issued.

There are two other keywords that we need to mention:

ICSF This keyword indicates that the private key associated with this certificate is stored in ICSF. If the `ICSF` keyword is not specified, the private key is stored in the RACF database. We discuss the ICSF keyword in more detail in 3.1.10, “RACDCERT and ICSF” on page 74.

SIGNWITH This keyword specifies the certificate with a private key that is signing the certificate. If not specified, the default is to sign the certificate with the private key of the certificate that is being generated. This creates a self-signed certificate. In our example we created a self-signed certificate because we did not specify the SIGNWITH parameter.

3.1.6.2 RACDCERT GENCERT example: Generate a SITE certificate

The next example shows how to generate a SITE certificate.

```
racdcert gencert site subjectsdn(cn('wtscicf.itso.ibm.com') OU('ITSO') O('IBM')
L('Poughkeepsie') SP('New York') C('US')) withlabel('Pauls trusted server')
```

Figure 56. RACDCERT GENCERT SITE example

The example shown in Figure 56 creates a SITE certificate for the site `wtscicf.itso.ibm.com` with the label "Pauls trusted server". The site certificate basically indicates that we *trust* the site `wtscicf.itso.ibm.com`. The trust part of a site certificate becomes more obvious when we discuss the key ring support in RACF in 3.2.1, "RACDCERT ADDRING: Creating a key ring" on page 78. The characteristics of the certificate are shown in Figure 57.

```
racdcert site list(label('Pauls trusted server')

Digital certificate information for SITE:

  Label: Pauls trusted server
  Status: TRUST
  Start Date: 2000/02/02 00:00:00
  End Date:   2001/02/02 23:59:59
  Serial Number:
       >00<
Issuer's Name:
     >CN=wtscicf.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Subject's Name:
     >CN=wtscicf.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***
```

Figure 57. RACDCERT LIST of example SITE certificate

The SITE certificate is no different than a certificate that was generated for a specific user ID, without the SIGNWITH keyword. The difference is in the ownership of the certificate. SITE certificates are not owned by a standard user ID. A new user ID is automatically created in RACF when you first IPL a release 8 called *irrsitec*. The user ID *irrsitec* is the owner of all SITE certificates in your RACF database. Therefore, when you create certificates for an OS/390 server you will likely create them with a normal user ID as owner because the server will require that the owner of the certificate is the user ID of the process running (started task).

It is more likely for you to add a SITE certificate that may be a peer in, for example, a VPN connection, rather than actually creating a SITE certificate.

3.1.6.3 RACDCERT GENCERT example: Generate a CA certificate

The next example shows how to generate a certificate authority (CA) certificate that we will use to sign other certificates with.

```
racdcert gencert certauth subjectsdn(cn('IBM-ITSO MAIN CA') OU('ITSO') O('IBM')
L('Poughkeepsie') SP('New York') C('US')) withlabel('ITSO CA')
```

Figure 58. RACDCERT GENCERT CERTAUTH example

The example shown in Figure 58 creates a CA certificate for the certificate authority “IBM-ITSO CA” with the label “ITSO CA”. This CA certificate can now be used for signing other certificates. The CA certificate that was created has the characteristics shown in Figure 59.

```
racdcert certauth list(label('ITSO CA')

Digital certificate information for CERTAUTH:

  Label: ITSO CA
  Status: TRUST
  Start Date: 2000/02/02 00:00:00
  End Date:   2001/02/02 23:59:59
  Serial Number:
       >00<
Issuer's Name:
     >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Subject's Name:
     >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***
```

Figure 59. RACDCERT LIST of example CA certificate

Note: When you become your own certificate authority there are legal implications to consider, especially when you start issuing certificates to external parties (for example, to customers and others).

To accept certificates signed by a certificate authority, the certificate authority certificate has to be in the key ring (key database) of the application that accepts certificates for authentication, like Websphere or the TN3270 server on OS/390. We discuss this further when we describe the RACDCERT EXPORT command in 3.1.7, “RACDCERT EXPORT - exporting a certificate” on page 63.

The ownership of a CA certificate, like SITE certificates, is a special user ID called irrcerta. This user ID is automatically created as well in the RACF database when you first ipl a release 8. For further information on these user IDs, see 3.1.9, “A word about irrcerta and irrsitec” on page 73.

3.1.6.4 RACDCERT GENCERT example: Signing with a CA certificate

The next example shows how to generate a certificate and sign it with the CA certificate we created previously.

```
racdcert gencert subjectsdn(cn('wtscicf.itso.ibm.com') OU('ITSO') O('IBM')
L('Poughkeepsie') SP('New York') C('US')) signwith(certauth label('ITSO CA'))
```

Figure 60. RACDCERT GENCERT: Example 3

The SIGNWITH keyword indicates the label of the CA certificate we used to sign the certificate.

```
racdcert list(label('LABEL00000007'))

Digital certificate information for user GRAAFF:

  Label: LABEL00000007
  Status: TRUST
  Start Date: 2000/02/02 00:00:00
  End Date:   2001/02/02 23:59:59
  Serial Number:
       >01<
Issuer's Name:
     >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Subject's Name:
     >CN=wtscicf.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***
```

Figure 61. RACDCERT LIST of signed example certificate

Figure 61 shows the certificate that is created; the Issuer's Name indicates the certificate authority that signed the certificate.

Note: The label LABEL00000007 used in this example is a label generated by RACF. User ID GRAAFF owned more certificates, but we only listed our example certificate.

3.1.6.5 RACDCERT GENCERT example: Usage of a request data set

The RACDCERT GENCERT command allows you to specify a so called *request data set*. The request data set contains the user's generated public and X.509 distinguished name. The request data must be signed, DER-encoded, and then Base64 encoded according to the PKCS#10 standard.

In our example, we use a request data set that is created with the GSKKYMAN utility supplied with System SSL. The next screen shows the creation of the certificate request being made, using GSKKYMAN:

```
GRAAFF @ SC57:/u/graaff>gskkyman

            IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number:
 ===>  2
Enter key database name or press ENTER for "key.kdb": cakey.kdb
Enter password for the key database.......>
```

Figure 62. GSKKYMAN example - generate certificate request (1)

After opening the key database cakey.kdb, we can generate a certificate request using option 3 as shown in Figure 63:

```
Key database menu

Current key database is /u/graaff/cakey.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 3
Enter certificate request file name or press ENTER for "certreq.arm": racftest.a
rm
Enter a label for this key.............> racftest2
Select desired key size from the following options (512):
    1:    512
    2:    1024
Enter the number corresponding to the key size you want: 2
Enter certificate subject name fields in the following.
    Common Name (required).............> wtsc57.itso.ibm.com
    Organization (required)............> ibm
    Organization Unit (optional).......> itso
    City/Locality (optional)...........> Poughkeepsie
    State/Province (optional)..........> new york
    Country Name (required 2 characters)..> us

Please wait while key pair is created...

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
```

Figure 63. GSKKYMAN example - generate certificate request (2)

We now have the certificate request in the file racftest.arm. The file contents are shown in Figure 64.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrDCCARUCAQAwbDELMAkGA1UEBhMCdXMxCzAJBgNVBAgTAm55MRUwEwYDVQQH
EwxQb3VnaGtlZXBzaWUxDDAKBgNVBAoTA2libTENMAsGA1UECxMEaXRzbzEcMBoG
A1UEAxMTd3RzYzU3Lml0c28uaWJtLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEA8oz7LxHAWYU40//dkauAVQiVungpxACqXLlJFdnpBTe3/rvZbDM5DL8n
HrIUoOZDyzhTVPCy3Ee5N9RPpPV4lmdk/SbLEpPbFo4HaAKbtmAs8tUx9Ka74DSH
6NaHuG+JWLz2XM/mPtLmgAMQdENUXOL9JzJlwIGl1sm3meNS/YMCAwEAAaAAMA0G
CSqGSIb3DQEBBAUAA4GBACdm0gJBmbzfvLQ1jUhnk4mKgIPm0tGh91DpQZtTY/vS
bFtSdSEwyuHoWI2JTCUdfOgK7ZD31CG0E3EaJLZsKPXwBQi4pE/117lBXvUlUlK6
QxaRpm6ELhrPLLbTvYrOscwHcZvnU0oBx8AIhKdRLxqbDS4lD3JPiyecVi5vEEuT
-----END NEW CERTIFICATE REQUEST-----
```

Figure 64. Certificate request content example

We now have to copy the file racftest.arm from the HFS directory to an OS/390 data set so we can use it for input to the RACDCERT GENCERT command.

Note: Remember that the file is in base64-encoded format, so do *not* copy it in binary format.

We can now issue RACDCERT GENCERT to sign the certificate request and add the signed certificate to the RACF database. The command issued is shown in Figure 65.

```
racdcert gencert(racftest.arm) signwith(certauth label('ITSO CA'))
IRRD113I The certificate that you are creating has an incorrect date range.
The certificate is added with NOTRUST status.
```

Figure 65. RACDCERT GENCERT command example with request data set

Note: The informational message indicates a date problem. Basically, we tried to sign a certificate with a CA certificate that expired before the certificate expired. That is why it is added with NOTRUST. The certificate we generated using the request data set defaulted to today date, as shown in Figure 66.

```
racdcert list(label('LABEL00000009')

Digital certificate information for user GRAAFF:

  Label: LABEL00000009
  Status: NOTRUST
  Start Date: 2000/02/07 00:00:00
  End Date:   2001/02/07 23:59:59
  Serial Number:
   >04<
  Issuer's Name:
   >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
  Subject's Name:
   >CN=wtsc57.itso.ibm.com.OU=itso.O=ibm.L=Poughkeepsie.SP= new york.C=us<
  Private Key Type: None
  Ring Associations:
  *** No rings associated ***
```

Figure 66. RACDCERT LIST of signed certificate from request data set

The culprit, as stated before, is the end date of the CA certificate, shown in Figure 59 on page 59. The end date is `2001/02/02` and, as you can see in Figure 66, the end date of the certificate is `2001/02/07`. To avoid the informational message you can issue the RACDCERT GENCERT command with an end date before `2001/02/07` as shown in Figure 67.

```
racdcert gencert(racftest.arm) signwith(certauth label('ITSO CA'))
notafter(date(2001-01-21))
```

Figure 67. RACDCERT GENCERT example with notafter date

The certificate is now added without any warning message and added with TRUST status, as shown in Figure 68 on page 63.

```
racdcert list(label('LABEL00000010')

Digital certificate information for user GRAAFF:

  Label: LABEL00000010
  Status: TRUST
  Start Date: 2000/02/07 00:00:00
  End Date:   2001/01/21 23:59:59
  Serial Number:
     >05<
  Issuer's Name:
        >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
  Subject's Name:
        >CN=wtsc57.itso.ibm.com.OU=itso.O=ibm.L=Poughkeepsie.SP= new york.C=us<
  Private Key Type: None
  Ring Associations:
   *** No rings associated ***
```

Figure 68. RACDCERT LIST of example signed certificate with other date

3.1.7 RACDCERT EXPORT - exporting a certificate

The RACDCERT EXPORT command allows for the export of the certificates to a file in a specific format. The command syntax is shown in Figure 69.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]
         EXPORT(LABEL('label-name'))
         DSN(output-data-set-name)
         [FORMAT(CERTDER|CERTB64)]
```

Figure 69. RACDCERT EXPORT command syntax

The export can be done in either:

CERTDER A DER-encoded X.509 certificate format

CERTB64 A DER-encoded X.509 certificate which has been encoded using base64. This format is the default value.

Note: Both of these formats do *not* contain the private key. The export function does not support the export of the private key.

As an example, we use the CA certificate we created earlier. The command issued is shown in Figure 70.

```
racdcert certauth export(label('ITSO CA')) dsn(itsoca.der) format(certder)
```

Figure 70. RACDCERT EXPORT example

The output data set contains the CA (certauth) certificate in a DER-encoded X.509 certificate format. The DER encoded X.509 certificate format is a *binary* format.

Depending on the usage of the CA certificate, the browsers out there do not know about the certificate authority ITSO CA. Having a client (browser) connect to a secure server on OS/390 using a server certificate signed by this certificate authority will fail if the CA certificate was not added to the browser. The next

steps are to get the CA certificate installed into the browser. This can achieved in either of two ways:

1. Serve the file that contains the CA certificate in DER format through a Web server and when the client (browser) selects the file, the CA certificate will be installed as a “trusted root” certificate.

   It requires a mime type of `.der` to be defined in the HTTPD.CONF file of the OS/390 Web server as shown:

```
AddType .der application/x-x509-ca-cert binary 1.0 #Browser CA certificate
```

2. Download the file to the workstation and point the browser to the file. This will install the certificate as a “trusted root” certificate. The browser already has a mime type defined for files with an extension of .der.

3.1.7.1 Install CA certificate using an OS/390 Web server

To install the CA certificate through a Web server we have to copy the exported certificate to an HFS directory that is being served up by our Web server. Make sure you do this in binary mode. Next we can point our browser to the URL containing our CA certificate, as shown in Figure 71. We used Internet Explorer in our examples: Netscape presents similar dialogs to install a new CA certificate.

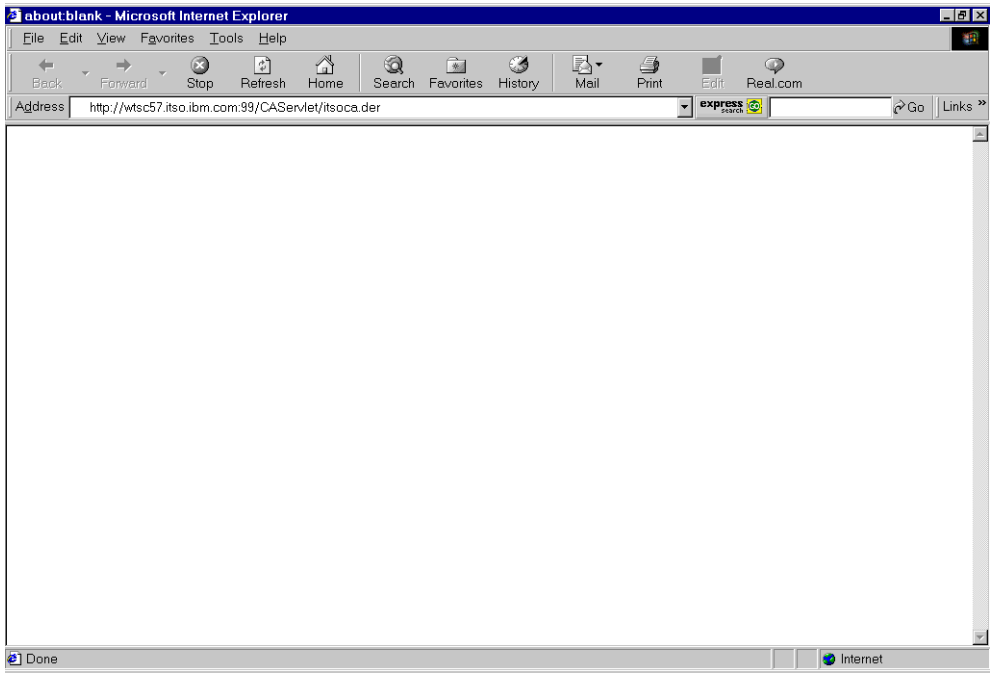


Figure 71. CA certificate served through Web server

Next we are prompted to save or open the file, as shown in Figure 72 on page 65.

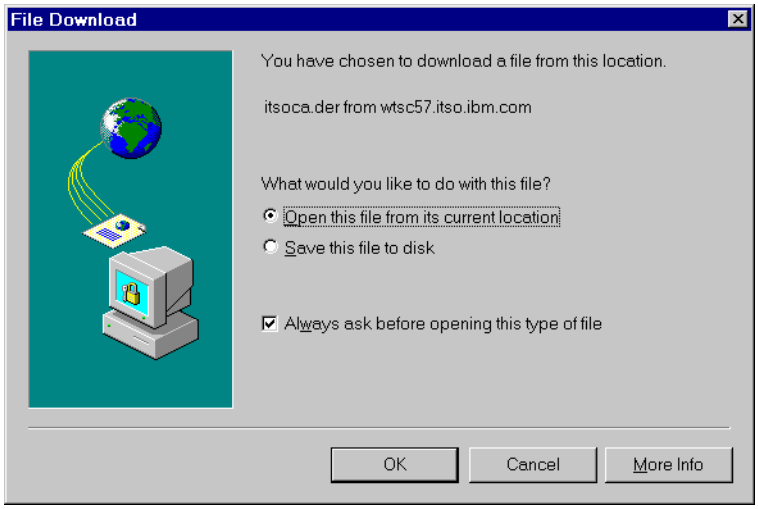


Figure 72. CA Certificate opened from location

Choose **Open this file from its current location** to start the installation dialog as shown in Figure 73.

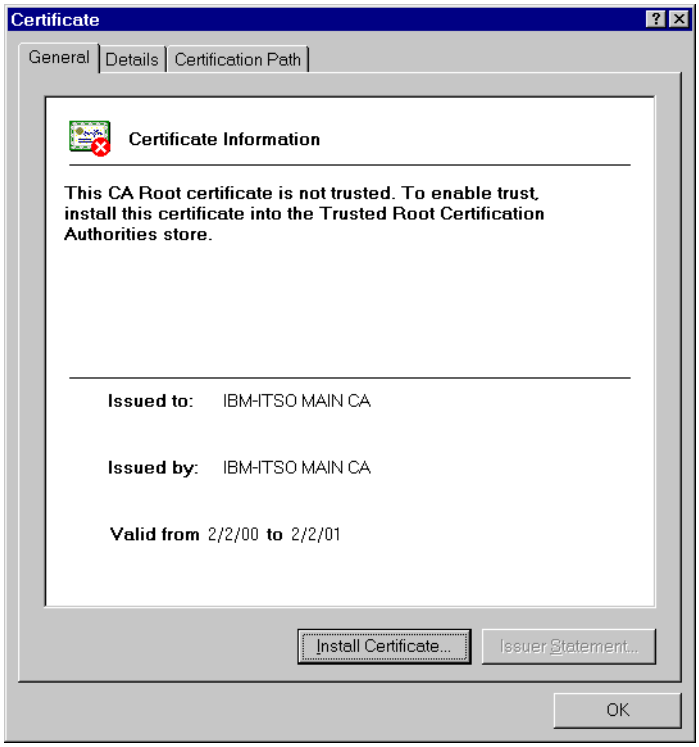


Figure 73. Installation dialog of CA certificate

Next click **Install certificate**. This starts the Certificate Manager Import Wizard, as shown in Figure 74 on page 66.

Figure 74. Certificate Manager Import Wizard: Screen 1

The wizard guides you to properly install the CA certificate into the right certificate store of your browser. Click **Next** to continue the installation of the CA certificate.

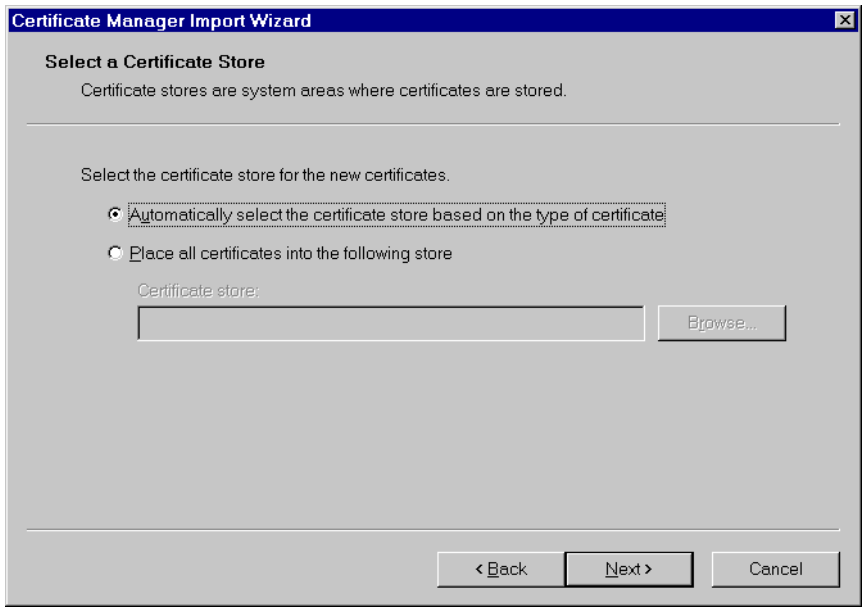


Figure 75. Certificate Manager Import Wizard: Screen 2

Figure 75 shows the next window, where you select the certificate store where you want to store the CA certificate. You can either let Internet Explorer (IE) do it itself or you can guide IE to where you want to store. In our example, we let IE choose the right certificate store for us. Click **Next** to continue the installation of the CA certificate.

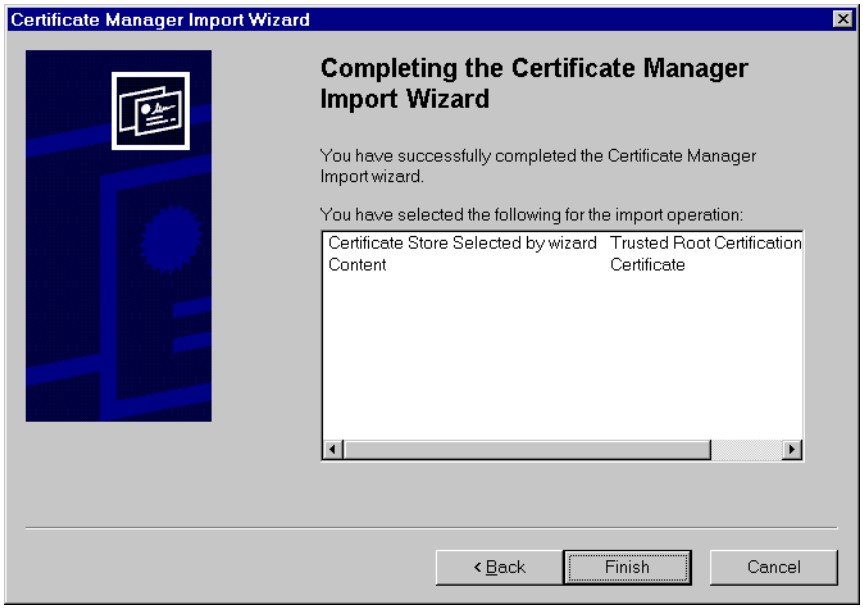


Figure 76. Certificate Manager Import Wizard: Screen 3

Figure 76 shows the window that tells you the certificate store selected for the installation. IE selected the Trusted Root Certification Authorities as the store for the CA certificate. Click **Finish** to continue with the installation of the CA certificate.

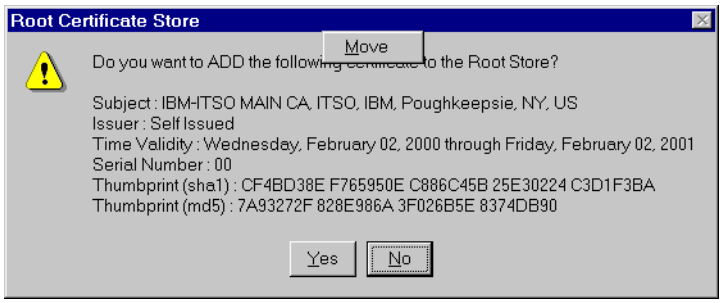


Figure 77. Certificate Manager Import Wizard: Screen 4

Figure 77 shows the conformation window where you specify whether you really want to add the new CA certificate to the “root store”. Click **Yes** to complete the installation of the CA certificate. A successful import message is displayed, as shown Figure 78, when the import successfully completes.

Figure 78. Certificate Manager Import Wizard: Screen 5

3.1.7.2 Install CA certificate using a web browser

To install the CA certificate through a web browser you have to have the file that contains the CA certificate on your local filesystem or available through a network drive. Make sure that when you download the file you do this in *binary* mode, because the DER-encoded format is in binary form.

In this example we use Netscape Navigator. From the menu bar select the **File** option and then select **Open File**. A window appears as shown in Figure 79, where you make the file selection that contains the CA certificate.

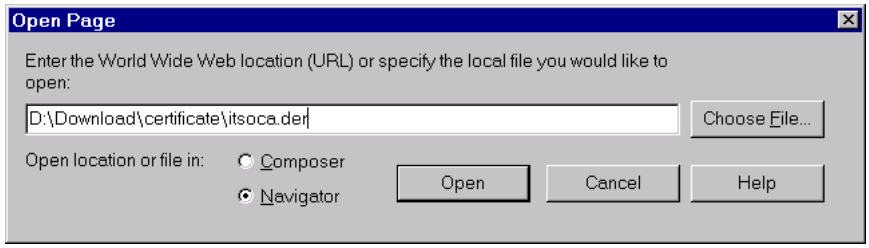


Figure 79. Netscape Navigator Choose File selection window

We downloaded our CA certificate to a directory called d:\download\certificate and called our file itsoca.der. Click **Choose File** to select your file. Next click **Open** to start the installation process of the CA certificate.

Figure 80. Netscape Navigator New Certificate Authority: Window 1

Figure 80 shows the window that appears next, to inform you that you are about to install a new CA certificate. Click **Next** to continue the installation of the CA certificate.

Netscape Navigator presents another warning window, requiring you to confirm the installation of the new CA certificate, as shown in Figure 81 on page 69.

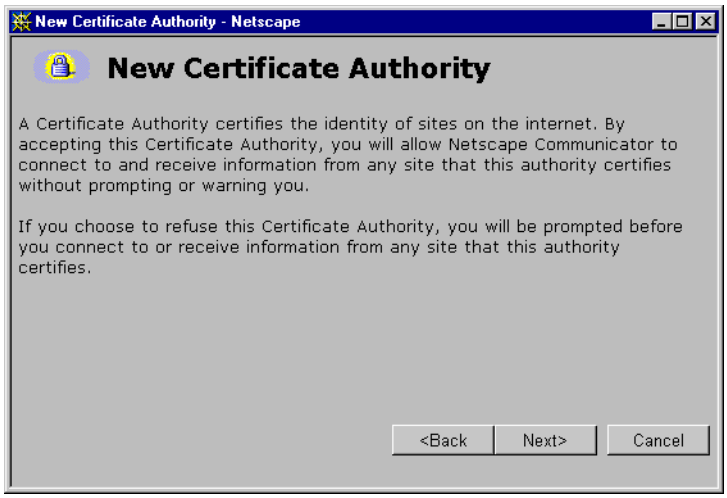


Figure 81. Netscape Navigator New Certificate Authority: Window 2

Click **Next** to continue the installation of the CA certificate.

Figure 82. Netscape Navigator New Certificate Authority: Window 3

Figure 82 shows the next window of the installation process of the CA certificate. It gives you the opportunity to examine the CA certificate by clicking **More Info...** This displays the content of the CA certificate, as shown in Figure 83 on page 70.

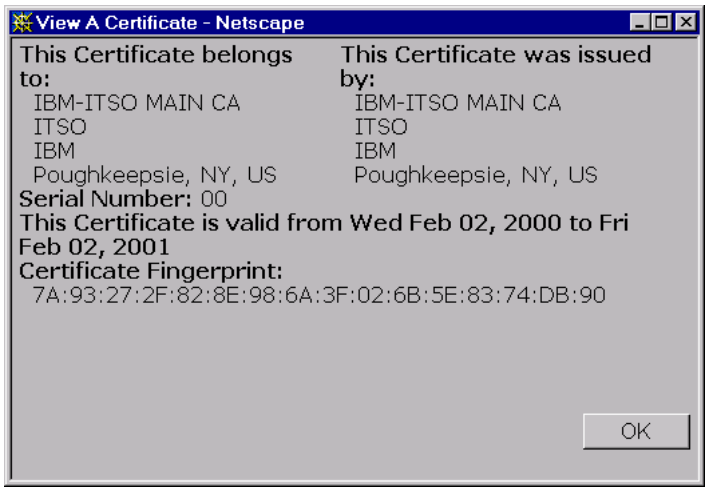


Figure 83. Netscape Navigator: View a certificate window

After verifying the information presented to you, click **OK** to return to the window shown in Figure 82 on page 69. Click **Next** to continue the installation of the CA certificate.

Figure 84. Netscape Navigator New Certificate Authority: Window 4

Figure 84 shows the next window to confirm the installation of the CA certificate. Here you choose what you want the CA certificate to do, as stated on the window presented to you. In our example we just want to certify network sites (our OS/390 server that uses a certificate signed by this CA). Click **Next** to continue the installation.

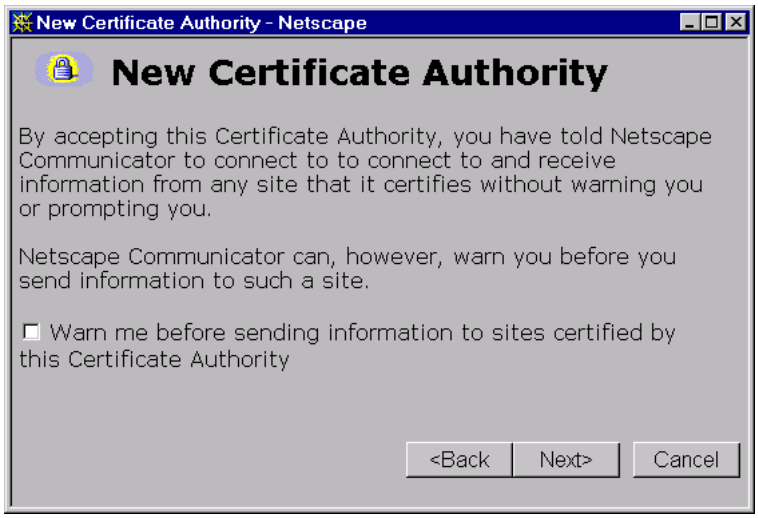


Figure 85. Netscape Navigator New Certificate Authority: Window 5

Netscape Navigator shows another informational window, as shown in Figure 85, where you can indicate whether you want to be prompted with an informational message when you connect to a site that has a certificate signed by this CA. In our example we did not want this to occur. Click **Next** to continue the installation. The next window, as shown in Figure 86, allows us to enter a name that identifies the CA in our certificate database. In this example we used ITSO CA.

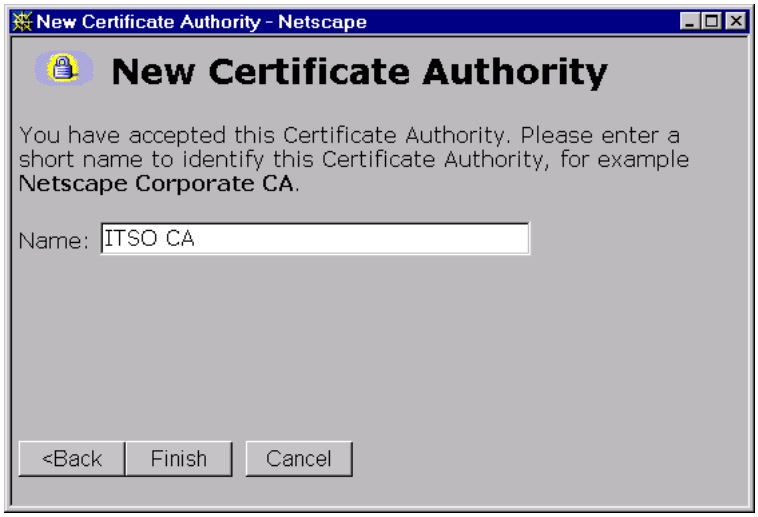


Figure 86. Netscape Navigator New Certificate Authority: Window 6

Click **Finish** to complete the installation process of the CA certificate. To verify the successful installation you can select **Security** on the main Navigator window. This presents the security window, as shown in Figure 87 on page 72, where you can select certificates of signers (Netscape uses the word *signers* for Certificate Authorities). See if the name you entered previously for your new CA certificate is there.

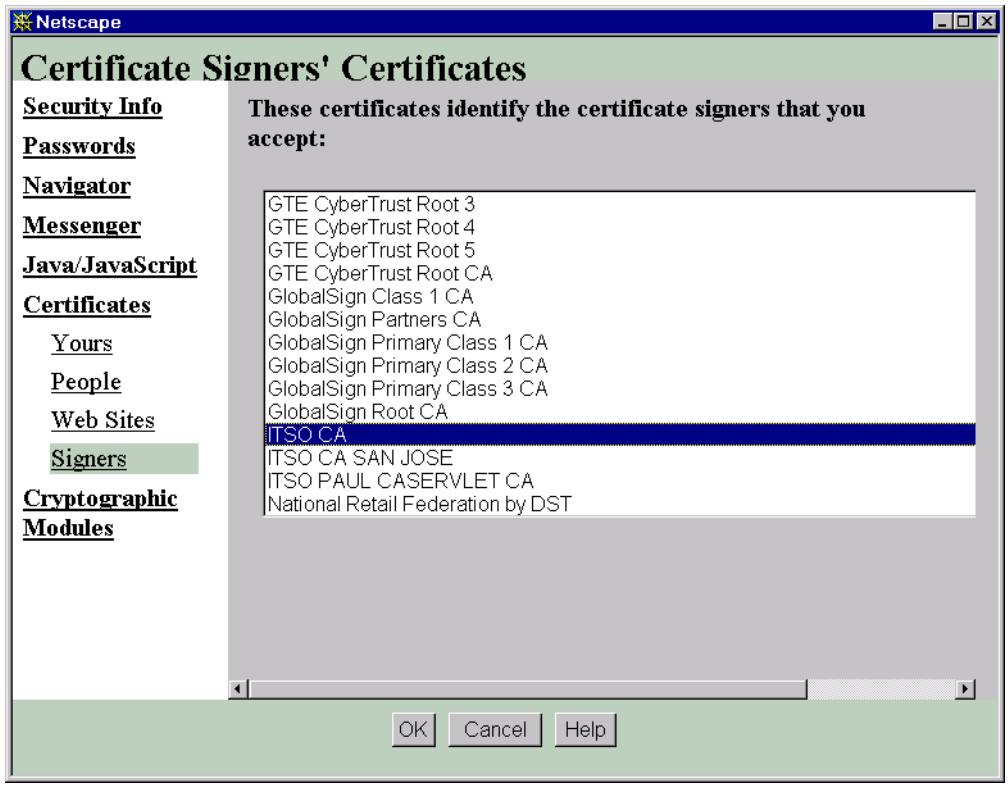


Figure 87. Netscape Navigator Certificate Signers' Window

3.1.8 RACDCERT GENREQ: Create a certificate request

The RACDCERT GENREQ command creates a PKCS#10 base64-encoded certificate request and writes it to a data set. This request contains the subject's distinguished name and public key, and is signed with the private key associated with the specified certificate. Typically, these requests are sent to a certificate authority; however, they can also be imported into (and signed by) RACF using the GENCERT function with a *request-data-set-name*.

The command syntax is shown in Figure 88.

```
RACDCERT [ID(userid) | SITE | CERTAUTH]

  GENREQ(LABEL('label-name')) DSN(output-data-set-name)
```

Figure 88. RACDCERT GENREQ command syntax

The GENREQ keyword requires that the certificate has a private key associated with it. If no private key is associated with the certificate, an informational message is issued and processing stops. *Label-name* identifies the certificate.

Therefore, we first have to have a "self-signed" certificate installed in our RACF database to be able to create a certificate request. This can be achieved using the RACDCERT GENCERT or ADD command.

Note: This function might be confusing, but the GENREQ keyword does not generate private/public key pairs. So this has to be done through a RACDCERT ADD or GENCERT.

In our example, we use the self-signed certificate we created earlier in 3.1.6.1, “RACDCERT GENCERT example: Generate a certificate” on page 56. We issue the following RACDCERT GENREQ command to create the certificate request data set, as shown in Figure 89.

```
racdcert genreq(label('LABEL00000005')) dsn('graaff.racfreq.pkcs10')
```

Figure 89. RACDCERT GENREQ command example

The data set graaff.racfreq.pkcs10 contains the certificate request, which can now be shipped (cut/paste), for example, to the external Certificate Authority that signs the certificate request. Another example is to sign the certificate with a RACF GENCERT(request-data-set-name).

3.1.9 A word about irrcerta and irrsitec

The anchor points for CA and site certificates, irrcerta and irrcitec respectively, require special attention.

These user IDs show up as revoked in the RACF database, and have no default group, so logons will fail. Care should be taken when you specify these user IDs in a RACF ADDUSER, LISTUSER and DELUSER command. A list of all user IDs (LISTUSER *) will return these user IDs, and so will a RACF SEARCH when the search criteria match with these user IDs. The same goes for the RACF macros that will perform operations on entries in the RACF database.

```
SR NOMASK CLASS(USER)

irrcerta
irrsitec
.....
```

Figure 90. RACF SR NOMASK CLASS(USER) output

Figure 91 on page 74 shows an LU * output, to show the RACF database entries for irrcerta and irrsitec.

```
LU *

USER=irrcerta  NAME=CERTAUTH Anchor       OWNER=irrcerta  CREATED=99.173
 DEFAULT-GROUP=          PASSDATE=00.000  PASS-INTERVAL=N/A
 ATTRIBUTES=REVOKED
REVOKE DATE=NONE   RESUME DATE=NONE
  LAST-ACCESS=UNKNOWN
  CLASS AUTHORIZATIONS=NONE
  NO-INSTALLATION-DATA
  NO-MODEL-NAME
  LOGON ALLOWED   (DAYS)          (TIME)
  ------------------------------------------
  ANYDAY                          ANYTIME
 SECURITY-LEVEL=NONE SPECIFIED
 CATEGORY-AUTHORIZATION
  NONE SPECIFIED
 SECURITY-LABEL=NONE SPECIFIED
USER=irrsitec  NAME=SITE Anchor           OWNER=irrsitec  CREATED=99.173
 DEFAULT-GROUP=          PASSDATE=00.000  PASS-INTERVAL=N/A
 ATTRIBUTES=REVOKED
 REVOKE DATE=NONE   RESUME DATE=NONE
 LAST-ACCESS=UNKNOWN
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
 ------------------------------------------
 ANYDAY                          ANYTIME
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
```

*Figure 91. LU * output to show irrcerta and irrsitec entries*

Note: When you perform an LU irrcerta directly, the LU command fails because the argument irrcerta is translated to uppercase automatically.

```
LU IRRCERTA

ICH30001I UNABLE TO LOCATE USER    ENTRY IRRCERTA
```

Figure 92. LU irrcerta example

3.1.10 RACDCERT and ICSF

When generating or adding a certificate, you have the option to utilize ICSF for storage of the private key in the ICSF PKDS.

This ICSF keyword is ignored if no private key is involved. If the ICSF keyword is not specified, or is specified but ICSF is not configured for PKA operations, the key is stored in the RACF database as a non-ICSF key and *no* error message is displayed. If the key is stored in ICSF, RACF stores a label (which refers to the key) in the RACF database.

The following ICSF Callable services are invoked by the RACDCERT command:

CSFPKI PKA key import callable service

CSFPKRC PKDS Record Create

The ICSF Started Task needs access to the following ICSF Callable Service:

CSFOWH One-way hash generate callable service

In our example we created a new certificate and added the ICSF keyword on our RACDCERT GENCERT command, as shown in Figure 93.

```
racdcert gencert subjectsdn(cn('wtsc59.itso.ibm.com') OU('ITSO') O('IBM')
l('Poughkeepsie') SP('New York') C('US')) icsf
```

Figure 93. RACDCERT GENCERT example with the ICSF keyword

The certificate we created is shown in Figure 94.

```
racdcert list(label('LABEL00000012')

Digital certificate information for user GRAAFF:

  Label: LABEL00000012
  Status: TRUST
  Start Date: 2000/02/07 00:00:00
  End Date:   2001/02/07 23:59:59
  Serial Number:
        >00<
  Issuer's Name:
        >CN=wtsc59.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
  Subject's Name:
        >CN=wtsc59.itso.ibm.com.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
  Private Key Type: ICSF
  Private Key Size: 1024
  Ring Associations:
   *** No rings associated ***
```

Figure 94. RACDCERT LIST output of a certificate with private key in ICSF

Note: Notice the private key type of ICSF.

To verify whether PKA operations are allowed, use the ICSF panels, as shown in Figure 95 on page 76.

```
------------------- Integrated Cryptographic Service Facility------------------

Enter the number of the desired option.

  1  MASTER KEY -  Set or change the system master key
  2  KGUP       -  Key Generator Utility processes
  3  OPSTAT     -  Installation options and Hardware status
  4  OPKEY      -  Operational key direct input
  5  UTILITY    -  OS/390 ICSF Utilities
  6  CKDS       -  CKDS Refresh and Initialization
  7  USERCNTL   -  User Control Functions
  8  PPINIT     -  Pass Phrase Master Key/CKDS Initialization

     Licensed Materials - Property of IBM

     This product contains "Restricted Materials of IBM"
     5647-A01 (C) Copyright IBM Corp. 1998.  All rights reserved.
     US Government Users Restricted Rights - Use, duplication or
     disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Press ENTER to go to the selected option.
Press END   to exit to the previous menu.
OPTION ===> 3
```

Figure 95. ICSF primary option panel

From the primary option menu, select option **3**. The panel shown in Figure 96 is displayed.

```
----------------- OS/390 ICSF - Installation Option and Status ----------------

Enter the number of the desired option above.

  1  OPTIONS  - Display Installation Options
  2  STATUS   - Display Hardware Status
  3  EXITS    - Display Installation exits and exit options
  4  SERVICES - Display Installation Defined Services









Press ENTER to go to the selected option.
Press END   to exit to the previous menu.

OPTION ===> 1
```

Figure 96. ICSF Installation Option and Status panel

Next select option **1**, to display installation options and status in effect for your environment.

A panel as shown in Figure 97 is displayed, indicating the parameters in effect and the active PKDS data set.

```
------------------ OS/390 ICSF - Installation Option Display Row 1 to 15 of 15

        Active CKDS: ICSF.V2R1.SCSFCKDS
        Active PKDS: ICSF.V2R1.SCSFPKDS
  OPTION                                                  CURRENT VALUE
  ------                                                  -------------
  CHECKAUTH     RACF check authorized callers             YES
  COMPAT        Allow CUSP/PCF compatibility              NO
  COMPENC       Compatibility services encryption algorithm DES
  DOMAIN        Current domain index or usage domain index 0
  KEYAUTH       Key Authentication in effect              YES
  MAXLEN        Maximum data length                       66000
  SSM           Allow Special Secure Mode                 NO
  TRACEENTRY    Number of trace entries active            599
  USERPARM      User specified parameter data

  (Dynamic)     Dynamic CKDS update services allowed      YES
  (Pkacall)     PKA callable services available           YES
  (PKDSRead)    Allow PKDS Read                           YES
  (PKDSWrite)   Allow PKDS Create, Delete, and Write      YES
                Encryption algorithm available            DES,CDMF
```

Figure 97. ICSF installation option display panel

3.2 Digital certificates and key ring support

Key rings are an essential component of RACF's Release 8 support. Key rings contain all of the certificates, site certificates, and certificate authority certificates that can be used by a user ID.

In a traditional server environment, each server has its own key ring. The contents of the server's key ring are usually left up to the server's administrators. This means that each server administrator sets up their own security or trust policy by virtue of the CA certificates that they choose to place in the server's key ring.

With OS/390, key ring content is determined by the RACF security administrator. Server administrators can only define a trust policy that is a subset of the installation security policy. That is, they may place into the server's key ring only those certificate authority certificates which have been approved by the RACF security administrator.

Key ring information is stored in the new DIGTRING class.

The RACDCERT command has been enhanced to support key rings. New keywords are:

ADDRING Adds a keyring to the RACF database.

DELRING Deletes a keyring from the RACF database, but not the certificates it contains.

LISTRING Lists the contents of a specified key ring.

CONNECT Establishes the relationship between a certificate and the key ring.

REMOVE Disassociates the certificate with the specified keyring.

The next sections discuss in more detail the key ring support and show examples of the new keywords of the RACDCERT command.

3.2.1 RACDCERT ADDRING: Creating a key ring

The RACDCERT ADDRING command creates a key ring. This key ring must not already exist for this user. Key ring names become names of RACF profiles in the DIGTRING class, and can contain only characters that are allowed in RACF profile names. Although asterisks are allowed in ring-names, a single asterisk is not allowed.

The command syntax is show in Figure 98.

```
RACDCERT [ID(userid)]

    ADDRING(ring-name)
```

Figure 98. RACDCERT ADDRING command syntax

Note: *Ring-name* is the name of the key ring being created.

Note: You can only specify ID(*userid*) with the RACDCERT ADDRING command. Users are the only owners of key ring. You can *not* specify SITE or CERTAUTH for this command.

Unlike the utilities MKKF, IKEYMAN, and GSKKYMAN, that ship with Websphere, Telnet and others on OS/390, the RACDCERT ADDRING command creates an empty keyring.

The RACF administrator has to make a conscious decision about setting up a security policy in the key ring created. What we mean by *security policy* here is what CA certificates are going to be defined (connected) to this key ring. To allow a client (personal) certificate to be used on, for example, an SSL connection, the CA that signed that client certificate has to be defined in the key ring of the server the client is connecting to.

In our example we create a keyring called “pauls-keyring”, as shown in Figure 99.

```
RACDCERT ID(GRAAFF) ADDRING(pauls-keyring)
```

Figure 99. RACDCERT ADDRING command example

We now show that the keyring created is indeed empty, by using the RACDCERT LISTRING command, as show in Figure 100 on page 79. The RACDCERT LISTRING command is discussed in more detail in 3.2.4, “RACDCERT LISTRING: Listing the content of a key ring” on page 83.

```
RACDCERT ID(GRAAFF) LISTRING(pauls-keyring)

Digital ring information for user GRAAFF:

   Ring:
        >pauls-keyring<
   *** No certificates connected ***
```

Figure 100. RACDCERT LISTRING command output from example key ring

To install a certificate, whether personal, SITE or CA, we use the RACDCERT CONNECT command, which is discussed in the next section.

3.2.2 RACDCERT CONNECT: Install a certificate in a key ring

The RACDCERT CONNECT command is used to add a certificate to a key ring. This certificate must be added to the RACF database by a RACDCERT ADD or RACDCERT GENCERT command prior to issuing the CONNECT command. The command syntax for RACDCERT CONNECT is shown in Figure 101.

```
RACDCERT [ID( userid)]
  CONNECT([ID( userid) | SITE | CERTAUTH]
    LABEL( 'label-name')
    RING( ring-name)
   [DEFAULT]
   [USAGE(PERSONAL | SITE | CERTAUTH)])
```

Figure 101. RACDCERT CONNECT command syntax

The `ID( userid)` keyword indicates that the certificate being added to the key ring is a user certificate, and `userid` is the user ID that is associated with this certificate. If the `ID` keyword is not specified, it defaults to the value specified or the default value on the RACDCERT command. The SITE keyword indicates that the certificate being added to the key ring is a site certificate. The CERTAUTH keyword indicates that the certificate being added to the key ring is a certificate authority certificate.

The `LABEL( 'label-name')` keyword specifies the certificate that is being added to the key ring. When specifying the CONNECT keyword, LABEL must also be specified.

The `RING( ring-name)` keyword specifies the ring to which this certificate is being added. When specifying the CONNECT keyword, RING must also be specified.

The `DEFAULT` specifies that the certificate is the default certificate for the ring. Only one certificate within the key ring can be the default certificate. If a default certificate already exists, its *default* status is removed, and the specified certificate becomes the default certificate. If you want the specified certificate to be the default, `DEFAULT` must be explicitly specified. If you have a key ring with a default certificate and you want to remove the default status of the certificate without defining another certificate as the default certificate, `CONNECT` the certificate again without specifying the `DEFAULT` keyword.

The USAGE(PERSONAL), USAGE(SITE)or USAGE(CERTAUTH) keywords specify how this certificate is used within the specified ring. If no usage is specified, the usage is the same as the certificate that is being connected.

The USAGE keyword allows the altering of the trust policy within the confines of a specific key ring. For example, a CERTAUTH certificate connected with USAGE(PERSONAL) can be used to demote a certificate authority certificate in order to insure that it is not used as a certificate authority in this ring. It can be used as a personal certificate if a private key is present. However, typically one would not be present. Consequently, connecting a CERTAUTH certificate as USAGE(PERSONAL) is a way of marking it NOTRUST for this key ring only. Also, a personal certificate connected with USAGE(CERTAUTH) can be used to promote an ordinary user certificate to a certificate authority certificate. It can then be used to authenticate user certificates for this key ring only.

For the sake of consistency, other certificate and USAGE variations are supported. However, there is currently no practical application for them. When using the USAGE keyword to change the usage of a certificate, such as is done when a PERSONAL certificate is being used as a SITE or CERTAUTH certificate, RACDCERT must ensure that you have the ability to define a SITE or CERTAUTH certificate by authenticating that the command issuer has CONTROL authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. This ensures that a user cannot bypass the installation security policy through the use of USAGE.

In our examples we use the certificates we created earlier in the RACDCERT GENCERT examples.

3.2.2.1 RACDCERT CONNECT personal certificate example

In this example we use a previously installed personal certificate (not generated by RACF) and connect it in to the keyring we created previously called “pauls-keyring” owned by user ID GRAAFF. The command we issued is shown in Figure 102.

```
racdcert id(graaff) connect(id(graaff) label('LABEL00000002') ring(pauls-keyring))
```

Figure 102. RACDCERT CONNECT: Example 1

The ID keyword points to the owner of the keyring; the connect ID keyword is the owner of the certificate to be connected. In our example they are the same, but they do not have to be the same.

Another point to note is we did not specify the USAGE parameter, so this defaulted to the type of certificate it is, personal, SITE or CERTAUTH. When we now list the content of the key ring, we see the certificate has been connected, as shown in Figure 103 on page 81.

```
racdcert id(graaff) listring(pauls-keyring)

Digital ring information for user GRAAFF:

  Ring:
       >pauls-keyring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  LABEL00000002                       ID(GRAAFF)     PERSONAL     NO
```

Figure 103. RACDCERT LISTRING command output of our key ring: Example 1

Next we add a SITE and a certificate authority (CERTAUTH) certificate to this keyring, using the ones we created in our previous examples.

3.2.2.2 RACDCERT CONNECT SITE certificate example

In this example we use the SITE certificate created in 3.1.6.2, “RACDCERT GENCERT example: Generate a SITE certificate” on page 58 and connect it to the keyring we created previously called “pauls-keyring”, owned by user ID GRAAFF. The command we issued is shown in Figure 104.

```
racdcert id(graaff) connect(site label('Pauls trusted server') ring(pauls-keyring))
```

Figure 104. RACDCERT CONNECT SITE certificate example

Note: Again the USAGE keyword defaulted to what the certificate indicated.

When we now list the key ring, we can see both the personal certificate and the SITE certificate and their usage, as shown in Figure 105.

```
racdcert id(graaff) listring(pauls-keyring)

Digital ring information for user GRAAFF:

  Ring:
       >pauls-keyring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  LABEL00000002                       ID(GRAAFF)     PERSONAL     NO
  Pauls trusted server                SITE           SITE         NO
```

Figure 105. RACDCERT LISTRING command output for our key ring: Example 2

We now assume that this SITE certificate is the certificate that identifies our server. We now have to change the *default* status of this certificate from *no* to *yes*. We basically have to re-issue the RACDCERT CONNECT command as before, but now specify the DEFAULT keyword, as shown in Figure 106.

```
racdcert id(graaff) connect(site label('Pauls trusted server') ring(pauls-keyring)
default)
```

Figure 106. RACDCERT CONNECT SITE example with DEFAULT keyword

When we list the key ring, we see the change of the default status, as shown in Figure 107.

```
racdcert id(graaff) listring(pauls-keyring)

Digital ring information for user GRAAFF:

  Ring:
       >pauls-keyring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  LABEL00000002                       ID(GRAAFF)     PERSONAL     NO
  Pauls trusted server                SITE           SITE         YES
  ITSO CA                             CERTAUTH       CERTAUTH     NO
```

Figure 107. RACDCERT LISTRING command output for our key ring: Example 3

3.2.2.3 RACDCERT CONNECT CERTAUTH certificate example

In this example we use the CERTAUTH certificate we created in 3.1.6.3, “RACDCERT GENCERT example: Generate a CA certificate” on page 58 and connect it in to the keyring we created previously called “pauls-keyring”, owned by user ID GRAAFF. The command we issued is shown in Figure 108 on page 82.

```
racdcert id(graaff) connect(certauth label('ITSO CA') ring(pauls-keyring))
```

Figure 108. RACDCERT CONNECT CERTAUTH certificate example

Note: Again, the USAGE keyword defaulted to what the certificate indicated.

When we now list the key ring, we can see both the personal certificate, SITE and CERTAUTH certificate and their usage, as shown in Figure 109.

```
racdcert id(graaff) listring(pauls-keyring)

Digital ring information for user GRAAFF:

  Ring:
       >pauls-keyring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  LABEL00000002                       ID(GRAAFF)     PERSONAL     NO
  Pauls trusted server                SITE           SITE         NO
  ITSO CA                             CERTAUTH       CERTAUTH     NO
```

Figure 109. RACDCERT LISTRING command output for our key ring: Example 4

3.2.3 RACDCERT REMOVE: Remove a certificate from a key ring

The RACDCERT REMOVE command removes a certificate from a key ring. Basically it is like the RACF REMOVE command that removes a user from the specified group, without deleting that user or group. The same is true for the RACDCERT REMOVE command. It disassociates the certificate from the specified key ring without deleting the certificate or the key ring.The command syntax is shown in Figure 110 on page 83.

```
RACDCERT [ID( userid)]
  REMOVE([ID( userid) | SITE | CERTAUTH]
    LABEL( 'label-name')
    RING( ring-name)
```

Figure 110. RACDCERT REMOVE command syntax

The `ID( userid)` keyword indicates that the certificate being removed is a user certificate, and `userid` is the user ID that is associated with this certificate. If the `ID` keyword is not specified, it defaults to the value that is specified or defaulted to on the RACDCERT command. SITE indicates that this is a site certificate, and CERTAUTH indicates that this is a certificate authority certificate.

We only use one example here, because the RACDCERT REMOVE command is very similar to the RACDCERT CONNECT command.The command we issued is shown in Figure 111.

```
racdcert id(graaff) remove(certauth label('ITSO CA') ring(pauls-keyring))
```

Figure 111. RACDCERT REMOVE CERTAUTH certificate example

When we now list the key ring, we can see that the CERTAUTH certificate is indeed removed, as shown in Figure 112.

```
racdcert id(graaff) listring(pauls-keyring)

Digital ring information for user GRAAFF:

  Ring:
       >pauls-keyring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  LABEL00000002                       ID(GRAAFF)     PERSONAL     NO
  Pauls trusted server                SITE           SITE         NO
```

Figure 112. RACDCERT LISTRING command output for our key ring: Example 5

3.2.4 RACDCERT LISTRING: Listing the content of a key ring

In the previous paragraphs you have seen the usage of the RACDCERT LISTRING command already. We now go into some more detail. The RACDCERT LISTRING lists the content of a key ring or key rings. The command syntax is show in Figure 113.

```
RACDCERT [ID(userid)]

    LISTRING(ring-name / *)
```

Figure 113. RACDCERT LISTRING command syntax

The *ring-name* is the name of the key ring. To list all rings that are associated with a particular user, `LISTRING (*)` must be specified. For each certificate in the ring, the following information is displayed:

- Ring name
- Owner of the certificate (ID(name), CERTAUTH, or SITE)
- Label assigned to the certificate
- DEFAULT status of the certificate within the ring
- Usage within the ring

Note: Since only user IDs can own key rings, neither CERTAUTH nor SITE can be specified with LISTRING.

Example listings are shown in the previous paragraphs, such as Figure 109 on page 82 and Figure 112 on page 83.

3.2.5 RACDCERT DELRING: Deleting a key ring

The RACDCERT DELRING command deletes a key ring. The command syntax for the RACDCERT DELRING command is shown in Figure 114.

```
RACDCERT [ID(userid)]

    DELRING(ring-name)
```

Figure 114. RACDCERT ADDRING command syntax

Note: *Ring-name* is the name of the key ring being deleted.

Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with DELRING.

Note: When a DELUSER command is issued against a user ID, all of the key rings that are owned by that user ID are also deleted.

3.3 RACDCERT authorization

This section describes the authority required to perform the RACDCERT functions we discussed in this chapter.

3.3.1 Authority required for the RACDCERT functions

To issue the RACDCERT command, you must have one of the following authorities:

- SPECIAL
- Sufficient authority to resource IRR.DIGTCERT.function in the FACILITY class, as identified in Table 2.

Table 2. RACDCERT authority checks

FUNCTION	READ	UPDATE	CONTROL
ADD	Add a certificate to one's own user ID.	Add a certificate for someone else.	Add a certificate authority or site certificate.
ADDRING	Create a key ring for one's own user ID.	Create a key ring for another user ID.	Not applicable.
ALTER	Change the trust status or label of one's own certificate.	Change the trust status or label of someone else's certificate.	Change the trust status or label of a certificate authority or site certificate.
CONNECT	See Table 4 on page 86	See Table 4 on page 86	See Table 4 on page 86
DELETE	Delete one's own certificate.	Delete the certificate of someone else.	Delete a certificate authority or site certificate.
DELRING	Delete one's own key ring.	Delete the key ring of someone else.	Not applicable.
EXPORT	Export one's own certificate.	Export the certificate of another user.	Export a SITE or CERTAUTH certificate.
GENCERT	See Table 3 on page 86	See Table 3 on page 86	See Table 3 on page 86
GENREQ	Generate a request based on one's own certificate.	Generate a request based lon the certificate of another user.	Generate a request based on a SITE or CERTAUTH certificate.
LIST	List one's own certificate.	List the certificate of someone else.	List certificate authority or site certificates.
LISTRING	See one's own key ring.	See the key ring of someone else.	Not applicable.
REMOVE	Delete a certificate from one's own key ring.	Delete a certificate authority or site certificate from one's own key ring.	Delete one's own certificate from one's own key ring.

3.3.2 Authority required for the GENCERT function

The RACDCERT GENCERT function allows a certificate to be generated and signed. Effective controls on the user ID that is being associated with the certificate and what certificate is being used to sign the generated certificate are essential.

RACF performs two checks that determine the authority required for the RACDCERT GENCERT command:

1. How the certificate is being signed, specified with the SIGNWITH keyword.
2. What type of certificate is being generated, which is specified with the `ID()`, SITE or CERTAUTH keywords.

Table 3 shows the required authorization, in case you are not RACF SPECIAL when creating a certificate.

Table 3. Authority required to generate a certificate

SIGNWITH	**Own Certificate**	**Someone Else's Certificate**	**SITE or CERTAUTH Certificate**
`SIGNWITH` one's own certificate	READ authority to `IRR.DIGTCERT.ADD` and READ authority to `IRR.DIGTCERT.GENCERT`	UPDATE authority to `IRR.DIGTCERT.ADD` and READ authority to `IRR.DIGTCERT.GENCERT`	CONTROL authority to `IRR.DIGTCERT.ADD` and READ authority to `IRR.DIGTCERT.GENCERT`
`SIGNWITH` a SITE or CERTAUTH certificate	READ authority to `IRR.DIGTCERT.ADD` and CONTROL authority to `IRR.DIGTCERT.GENCERT`	UPDATE authority to `IRR.DIGTCERT.ADD` and CONTROL authority to `IRR.DIGTCERT.GENCERT`	CONTROL authority to `IRR.DIGTCERT.ADD` and CONTROL authority to `IRR.DIGTCERT.GENCERT`
`SIGNWITH` not specified	READ authority to `IRR.DIGTCERT.ADD` and READ authority to `IRR.DIGTCERT.GENCERT`	UPDATE authority to `IRR.DIGTCERT.ADD` and UPDATE authority to `IRR.DIGTCERT.GENCERT`	CONTROL authority to `IRR.DIGTCERT.ADD` and CONTROL authority to `IRR.DIGTCERT.GENCERT`

3.3.3 Authority required for the CONNECT function

The USAGE keyword allows a certificate to be connected to a ring and used in a manner that differs from the certificate's original use. For example, a certificate that is a personal certificate could be used as a certificate authority certificate.

The USAGE keyword is powerful, and must be controlled. The rules for connection are shown in Table 4, which shows the access control checks that are performed when connecting to one's own key ring; and Table 5 on page 87, which shows the access control checks that are performed when connecting to someone else's key ring.

Table 4. Authority required to connect to one's own key ring

USAGE	**Own Certificate**	**Someone Else's Certificate**	**SITE or CERTAUTH Certificate**
Personal	READ authority to `IRR.DIGTCERT.CONNECT`	UPDATE authority to `IRR.DIGTCERT.CONNECT`	UPDATE authority to `IRR.DIGTCERT.CONNECT`

USAGE	Own Certificate	Someone Else's Certificate	SITE or CERTAUTH Certificate
SITE/ CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT

Table 5. Authority required to connect to someone else's key ring

USAGE	Own Certificate	Someone Else's Certificate	SITE or CERTAUTH Certificate
Personal	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT
SITE/ CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT

3.4 Certificate Name Filtering (CNF)

CNF is a new function introduced by RACF APAR OW40129 and SAF APAR 40130 for OS/390 Security Server Version 2 Release 8.

CNF addresses several of the scalability and maintenance issues associated with RACF's current digital certificate support.

CNF allows you to:

- Map multiple digital certificates to a single, shared RACF user ID, while maintaining the security of your system.

  When IDs are shared, less certificate maintenance needs to be performed by your RACF system administrator.

- Map one digital certificate to a single RACF user ID, without the need to have the certificate available in OS/390 data set.

  This addresses the expiration problem of digital certificates, when mapping individual digital certificates to a specific RACF user ID.

- Optionally, add an additional criteria to map a digital certificate to a RACF user ID.

  This allows you to specify an application ID or system ID, to more specifically select the RACF user ID to be used for access to OS/390 resources.

Additionally, this APAR also introduces a new RACF user ID attribute called *restricted.* As the title indicates, this attribute allows a RACF user ID restricted access to OS/390 resources.

The following section addresses the updates required to implement the new certificate filtering support.

3.4.1 RACDCERT enhancements to support Certificate Name Filtering

The RACDCERT command is your primary administrative tool for managing digital certificates using RACF. Enhancements to the RACDCERT command

implement the Certificate Name Filtering functions. These new functions are shown in Figure 115 on page 88.

```
RACDCERT [ID(user-id) | MULTIID]
MAP [(cert-dsn)]
  [SDNFILTER('subject-dist-name-filter')]
  [IDNFILTER('issuer-dist-name-filter')]
  [CRITERIA('criteria-profile-name-template')]
  [WITHLABEL('label-name')]) [TRUST | NOTRUST]

LISTMAP (LABEL('label-name'))

ALTMAP (LABEL('label-name'))
  [NEWCRITERIA('criteria-profile-name-template')]
  [NEWLABEL('label-name')] [TRUST | NOTRUST]

DELMAP (LABEL('label-name'))
```

Figure 115. RACDCERT command functions for CNF

The new “mapping” function creates the linkage between the digital certificates and the RACF user ID, based on the following:

- Subject’s Distinguished Name Filter (SDNFILTER)

  For example, CN=Paul de Graaff, ou=ITSO,o=IBM,c=US

- Issuer’s Distinguished Name Filter (IDNFILTER)

  For example, Issuer's Name: CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=VeriSign Trust Network.O=VeriSign, Inc.

- Criteria

  Specifies the (optional) additional criteria to be used when selecting the RACF user ID to be mapped to.

The new parameters of the RACDCERT command are:

MAP Establishes the linkage between the digital certificate(s) and the RACF user ID

ALTMAP Changes the filter (linkage) defined

DELMAP Deletes the filter (linkage) defined

LISTMAP Lists the filter (linkage) defined

CNF support also introduces new RACF classes and a new anchor user ID called “irrmulti”. The new RACF classes are :

DIGTNMAP Maps a distinguished name to a RACF user ID

DIGTCRIT Maps additional criteria found through the DIGTNMAP class to a RACF user ID

The RACF user ID *irrmulti* is like the other anchor user IDs, *irrcerta* and *irrsitec*, and is used as an anchor for mapping profiles created using the MULTIID parameter of RACDCERT MAP.

3.4.1.1 RACDCERT MAP

The RACDCERT MAP command specifies a certificate name filter. It results in the creation of a profile in the DIGTNMAP class. DIGTNMAP profiles are used as filters when a user attempts to access the system using a digital certificate.

An associated user ID is found by comparing the issuer's distinguished name and the subject's distinguished name from the certificate with the filter values used to create the DIGTNMAP profile. The associated user ID is specified with the `ID` keyword or specified in DIGTCRIT profiles if `MULTIID` is specified. When you specify `MAP`, you must also specify `IDNFILTER`, `SDNFILTER`, or both.

A *data set name* can be specified with the `MAP` keyword. Data-set-name is the name of the data set that contains a certificate. The certificate provides a model for the filter names specified with `SDNFILTER` and `IDNFILTER`. The subject's distinguished name is used beginning with the value specified by `SDNFILTER`. The issuer's distinguished name is used beginning with the value specified by `IDNFILTER`. Using a model certificate is optional, but can reduce the chance of typographical errors when entering long filters for `SDNFILTER` or `IDNFILTER`.

The model certificate used with the `MAP` keyword can have an issuer's distinguished name or a subject's distinguished name that exceeds 255 characters. However, the portion of each used in the filter to associate a user ID with the certificate cannot exceed 255 characters.

The `IDNFILTER('issuer's-distinguished-name-filter')` keyword specifies the significant portion of the issuer's distinguished name that is used as a filter when associating a user ID with a certificate.

When specified without data-set-name, you must specify the entire portion of the distinguished name to be used as a filter.

The `SDNFILTER('subject's-distinguished-name-filter')` keyword specifies the significant portion of the subject's distinguished name. This is the part of the name that will be used as a filter when associating a user ID with a certificate.

When specified without data-set-name, you must specify the entire portion of the distinguished name to be used as the filter.

The `CRITERIA(criteria-profile-name-template)` keyword, when specified with `MULTIID`, indicates a dynamic user ID mapping. The user ID associated with this mapping profile is based not only on the issuer's distinguished name and the subject's distinguished name found in the certificate, but also on additional criteria. The criteria-profile-name-template specifies the additional criteria in the form of a name containing one or more variable names, separated by freeform text. These variable names begin with an ampersand (&) and end with a period. The freeform text should identify the variables contained in the template:

```
variable-name1=-name1.variable-name2=-name2...
```

For example, if the application identity and system identifier are to be considered in determining the user ID associated with this mapping, the CRITERIA keyword should be specified as follows:

```
CRITERIA(APPLID=&APPLID.SYSID=&SYSID)
```

The RACF-defined criteria are the application ID (APPLID) and the system-identifier (SYSID). When a user presents a certificate to the system for identification, the identity of the application being accessed (as well as the system the user is trying to access) becomes part of the criteria. The application passes its identity to RACF, and RACF determines the system-identifier. The system-identifier is the 4-character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. This value is substituted for &APPLID and &SYSID in the criteria.

Once the substitution is made, the fully expanded criteria template is used as a resource name to find a matching profile defined in the DIGTCRIT class using the RDEFINE command.

Attention

Criteria names other than APPLID and SYSID are allowed, but are effective in Certificate Name Filtering if the application supplies these criteria names and their associated values to RACF when the user attempts to access the application using a certificate. SYSID is determined by RACF, but APPLID must be specified with the initACEE callable service. Other criteria names should not be specified on RACDCERT unless you are instructed to do so in documentation for the application.

The `WITHLABEL('label-name')` keyword specifies the label that is assigned to this mapping. If specified, it must be unique to the user ID with which the mapping is associated. If WITHLABEL is not specified, a label is generated in the same manner issuing the WITHLABEL keyword for the RACDCERT ADD command.

Up to 32 characters can be specified for label-name. It can contain embedded blanks and mixed-case characters, and is stripped of lead and trailing blanks. If a single quotation mark is intended to be part of the label-name, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

The TRUST/NOTRUST keyword, when specified with MAP, indicates whether this mapping can be used to associate a user ID to a certificate presented by a user accessing the system. If neither TRUST nor NOTRUST is specified, the default is TRUST.

3.4.1.2 RACDCERT ALTMAP

The RACDCERT ALTMAP command changes the label, trust status, or criteria associated with the mapping identified by label-name. Specifying label name is required if more than one mapping is associated with the user ID. If NEWLABEL, NEWCRITERIA, or TRUST/NOTRUST is not specified, the mapping is not altered.

The command syntax is shown in Figure 115 on page 88.

3.4.1.3 RACDCERT DELMAP

The RACDCERT DELMAP command deletes the mapping identified by label-name for the specified user ID. Specifying the label name is required if more than one mapping is associated with the user ID.

Note: Mappings might also be deleted as part of DELUSER processing.

The command syntax is shown in Figure 115 on page 88.

3.4.1.4 RACDCERT LISTMAP

RACDCERT LISTMAP lists information about the mapping identified by label-name for the user ID specified. Do not specify LABEL if you intend to list all mappings associated with the user ID.

The command syntax is shown in Figure 115 on page 88.

3.4.2 Restricted access user IDs

Users entering the system by supplying a RACF user ID, password, or digital certificate can gain access to any RACF-protected resource that has a global access-checking entry allowing access. Users can also gain access to any resource where the UACC provides sufficient authority, unless the user ID or group name is excluded by being specifically placed on the access list. However, locating and updating a large number of profiles in this manner might cause the user to miss one of the profile's access list, which could result in a security exposure.

To solve this problem, a user ID can be given the restricted access attribute. User IDs with this attribute are ideal for shared user IDs assigned to users who do not identify themselves and for user IDs created for use with RACF's digital certificate name filtering support.

When access checking is performed, global access checking is bypassed for users with the restricted access. Neither ID(*) on the access list nor the UACC can be used to allow access to the resource.

Attention

Note that this attribute does not have an effect on access checking for OS/390 UNIX resources, such as HFS files. This type of resource has permission bits for owner, group and other, and does not have an access list. In this case, the permission bits apply to users with the restricted access attribute.

Be aware that resources that are protected by a profile in *warning* mode are accessible by *restricted* user IDs.

3.4.3 Certificate Name Filtering examples

This section provides some examples of the usage of Certificate Name Filtering when accessing OS/390 resources through the IBM HTTP Server for OS/390.

The examples are :

1. Map a digital certificate to a specific RACF user ID.

   This can also be achieved using the RACDCERT ADD command, but now we no longer need the data set containing the digital certificate and we avoid the need to setup the linkage again when the digital certificate expires. See 3.4.3.1, “Example 1” on page 92.

2. Create filters based on both subject’s and issuer’s distinguished name and map it to a userid.

This example helps us understand how the filter function works, before we make it more complicated. See 3.4.3.2, “Example 2” on page 98.

3. Create a filter based on issuer’s distinguished name and using the MULTIID option and a criteria to determine the RACF user ID.

   This example explains the added value of a criteria, or multiple criteria. See 3.4.3.3, “Example 3” on page 104.

3.4.3.1 Example 1

In this example we are setting up a map between one digital certificate and one RACF user ID. Our intent is to introduce the mapping capability, because the CNF support is really targeted towards mapping multiple (thousands, millions !) digital certificates to one “generic” RACF user ID.

The one-on-one relationship is useful for customers that do not want to use the RACDCERT ADD command. The advantage for these customers is that no longer does the digital certificate have to be provided in a data set, and the expiration of certificate is no longer a problem. It also reduces the size of the RACF database, because the certificate is no longer in the RACF database.

Before we start defining mapping profiles in the new DIGTNMAP class, we have to activate and raclist the DIGTNMAP class, as shown in Figure 116.

```
SETROPTS CLASSACT(DIGTNMAP)

SETROPTS RACLIST(DIGTNMAP)
```

Figure 116. SETR RACLIST command example to activate the DIGTNMAP class

To create a map between our sample certificate and our sample RACF user ID, we issued the command shown in Figure 117.

```
racdcert id(graaff) map(‘graaff.veri0312.cerbin’) idnfilter('CN=') sdnfilter('CN='
withlabel('pauls verisign cert')
```

Figure 117. RACDCERT MAP command example

Note: The data set graaff.veri0312.cerbin contains our sample certificate in a DER-encoded binary format. This data set can be created using the certificate export functions of Microsoft’s Internet Explorer.

The model data set is not really needed, but the way the distinguished name construction is set up it was easier to use the model data set rather then typing it all out (see Figure 118 on page 93).

Next we do a RACDCERT LISTMAP to see the results of the RACDCERT MAP command. Figure 118 on page 93 shows the output of the RACDCERT LISTMAP command. As you can see, it would have been quite cumbersome to type the exact distinguished name for both the issuers and the subject. That is why it might still be useful to have the client certificate available to the RACF administrator for these one on one relationships.

```
racdcert id(graaff) listmap

Mapping information for user GRAAFF:

   Label: pauls verisign cert
   Status: TRUST
   Issuer's Name Filter:
        >CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU<
        >=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=Ver<
        >iSign Trust Network.O=VeriSign, Inc.<
   Subject's Name Filter:
        >CN=Paul M de Graaff.OU=Digital ID Class 1 - Netscape.OU=Persona Not V<
        >alidated.OU=www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(<
        >c)98.OU=VeriSign Trust Network.O=VeriSign, Inc.<
```

Figure 118. RACDCERT LISTMAP output example

After we have defined the mapping profile, we have to issue a SETROPTS RACLIST(DIGTNMAP) REFRESH, to refresh the instorage profiles.

Note: You will not receive a warning message indicating a refresh is needed!

Next we execute our sample scenario. The scenario involves the following steps and will be used in our other examples as well:

1. We use a Web browser to serve an HTML page on our Web server on OS/390 and we use SSL with client authentication enabled. The URL we used was `https://wtsc57.itso.ibm.com/jack/mvsview.html`, as shown in Figure 119.

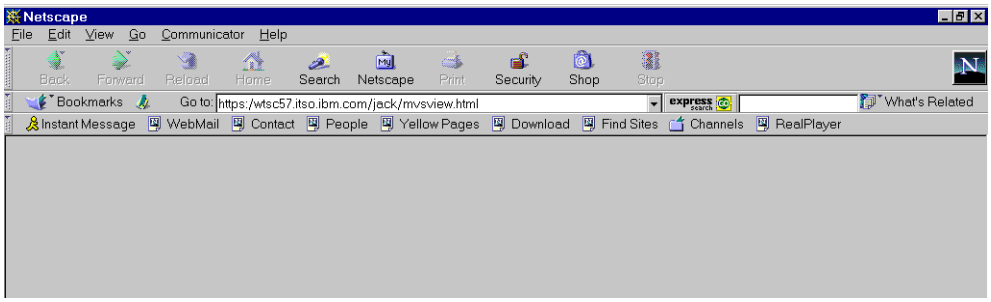


Figure 119. Netscape window to show the sample URL used

The protection setup for this URL in our OS/390 HTTP Server configuration file (`HTTPD.CONF`) is show in Figure 120 on page 94.

```
Protection mvs {
        ServerId        pauls-server
        AuthType        Basic
        Userid          PUBLIC
        SSL_ClientAuth  Client
        Mask            Anybody
}
Protection mvs2 {
        ServerId        pauls-server
        AuthType        Basic
        Userid          %%CERTIF%%
        SSL_ClientAuth  Client
        PasswdFile      %%SAF%%
        Mask            Anybody
}

Protect /jack/*                 mvs
Protect /jack2/readmvs.rexx     mvs2
....
Pass            /jack/*             /u/jjones/*
Exec            /jack2/*            /u/jjones/cgi-bin/*
```

Figure 120. HTTPD.CONF file showing the URL protection for our sample

2. After we press Enter to get the html page (mvsview.html), we are prompted for our client certificate, as shown in Figure 121.

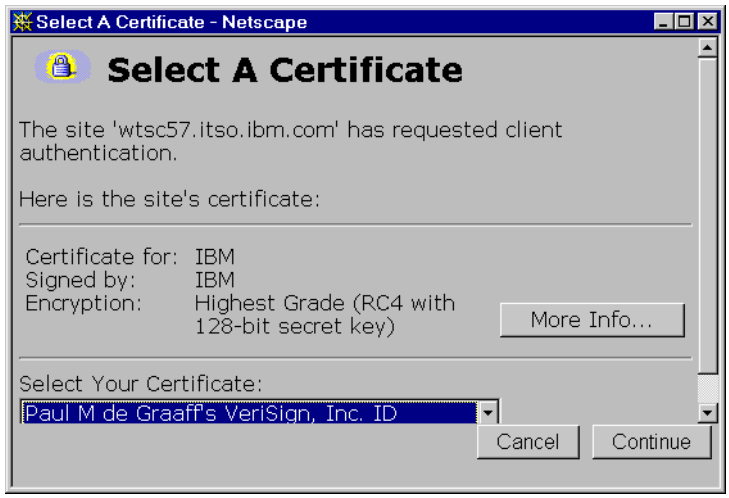


Figure 121. Netscape's prompt for a client certificate

We select the certificate that we used earlier to create the filter in step 1. Next we click **Continue** to get to our page. We have protected our Netscape Certificate Database with a password, so we receive a password prompt as shown in Figure 122 on page 95.

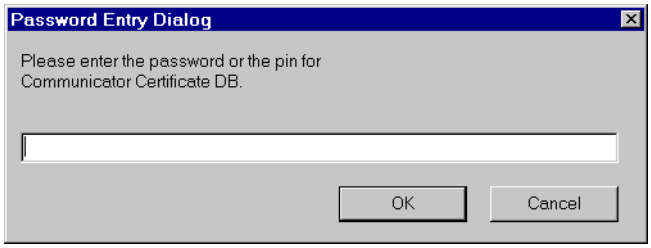


Figure 122. Netscape password prompt for the Certificate Database

After we enter our password and click **OK**, we finally get to our html page (`mvsview.html`) as shown in Figure 123.

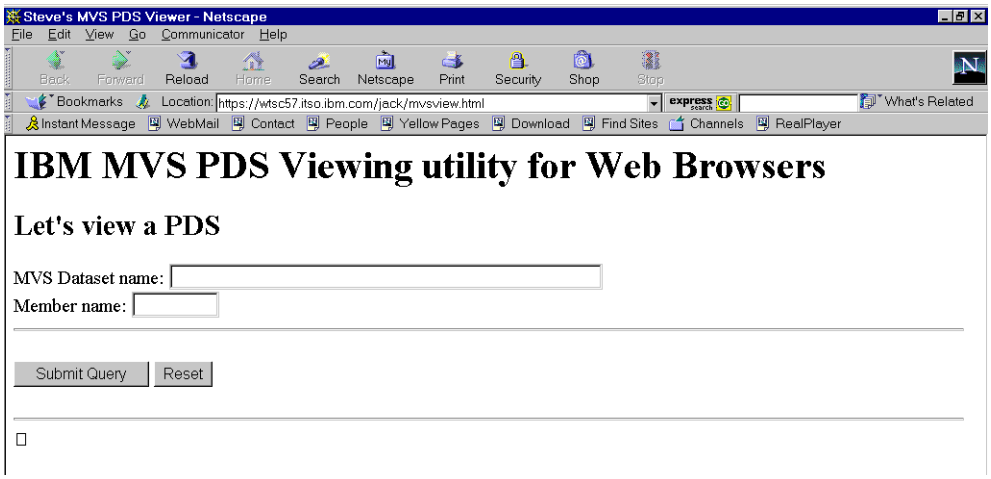


Figure 123. Netscape's window to show our mvsview.html page

If we look into the HTTPD log of the Web server, shown in Figure 124 on page 96, we can see what really happened here from a security perspective:

1. We have a match on the protect statement (1).
2. An authorization check occurs and requires a client certificate (2).
3. Access to the mvsview.html page is done using RACF user ID `PUBLIC` (3) as defined in the protection setup (see Figure 120 on page 94).

```
....
Protect..... /jack/* matched "/jack/mvsview.html" -> "/jack/mvsview.html" 1
Protection.. setup as defined in config file
Pass........ /jack/* matched "/jack/mvsview.html" -> "/u/jjones/mvsview.html"
Passing..... "/u/jjones/mvsview.html"
.....
AuthCheck... Translated path: "/u/jjones/mvsview.html" (method: GET)   2
Client certificate data:
   Common Name = Paul M de Graaff
   Organization = VeriSign, Inc.
   Organizational Unit = Digital ID Class 1 - Netscape
   Issuer Common Name = VeriSign Class 1 CA Individual Subscriber-Persona Not

   Issuer Organization = VeriSign, Inc.
   Issuer Organizational Unit = www.verisign.com/repository/RPA Incorp. By
......
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
Translated.. "/u/jjones/mvsview.html"
HTHandle..... Access as "PUBLIC" for Client "-unknown-" 3
```

Figure 124. HTTPD log: Example 1

4. The html page (as shown in Figure 123 on page 95) gives us the option to specify the PDS name and a member name we want to browse. After entering the PDS name (SYS1.SAMPLIB) and member name (IEFSSN00), the html page calls the REXX procedure MVSREAD.REXX to perform the actual operation.

   The protection setup for the REXX procedure (as shown in Figure 120 on page 94) specifies `%%certif%%` as the parameter for the user ID to be used. This indicates to the Web server to call SAF (initACEE) to determine if this certificate is mapped to a specific RACF user ID.

   Note: It will use profiles in both the DIGTCERT and DIGTNMAP class for that.

   Figure 126 on page 97 shows the Web server HTTPD log indicating the security processing that occurred:

   1. We have a match on the protect statement (1).
   2. An authorization check occurs and requires a client certificate (2).
   3. Access to SYS1.PARMLIB is done using RACF user ID GRAAFF (3) as a result of searching the profiles defined in the RACF classes DIGTCERT and DIGTNMAP.

      The map of the digital certificate to the RACF user ID is also recorded in SMF as a INITOEDP event, as shown in Figure 125.

```
INITOEDP       17:56:50   2000-04-22   GRAAFF
  graaff@us.ibm.com.CN=Paul M de Graaff.OU=Digital ID Class 1 -Netscape.
  OU=Person

  CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.
  OU=www.verisi
```

Figure 125. IRRADU00 SMF unload INITOEDP example

```
.....
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" -> 1
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx" 2
Client certificate data:
   Common Name = Paul M de Graaff
   Organization = VeriSign, Inc.
   Organizational Unit = Digital ID Class 1 - Netscape
   Issuer Common Name = VeriSign Class 1 CA Individual Subscriber-Persona Not

   Issuer Organization = VeriSign, Inc.
   Issuer Organizational Unit = www.verisign.com/repository/RPA Incorp. By
..
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
Translated.. "-null-"
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=GRAAFF       3
......
```

Figure 126. HTTPD log: Example 2

Figure 127 shows the final result of our first example, and as you can see it shows our installation's `IEFSSN00` member of our `SYS1.PARMLIB`.

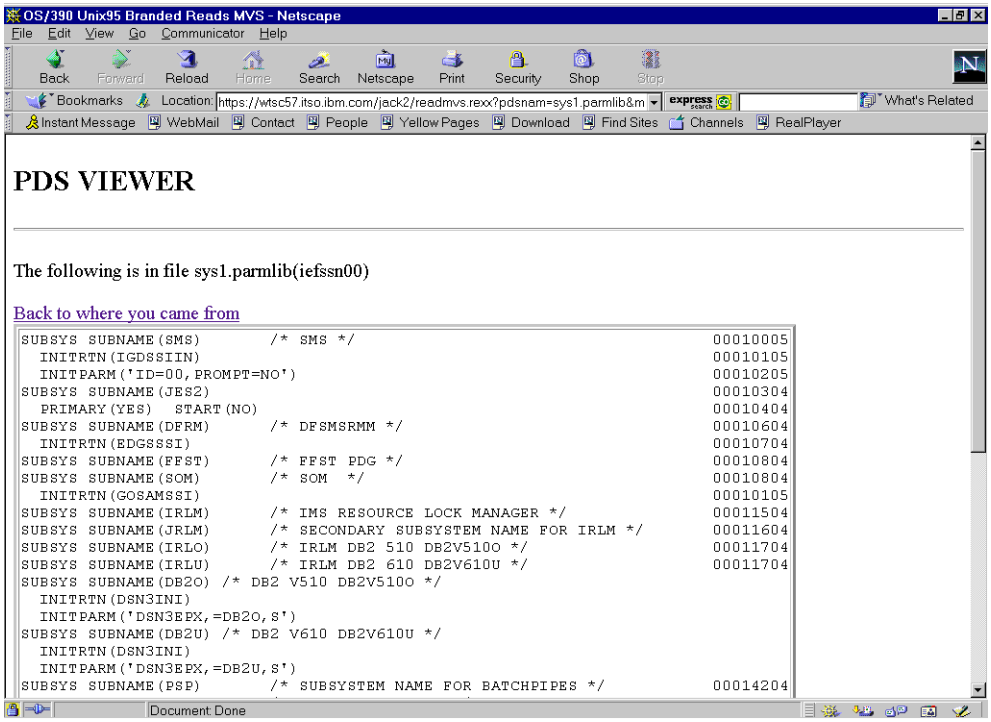


Figure 127. PDS viewer result page

If our certificate had not been defined to RACF, either through a RACDCERT ADD or a RACDCERT MAP, the system log would show the ICH408I error message shown in Figure 128.

```
ICH408I USER(WEBSRV  ) GROUP(IMWEB   ) NAME(ICSS WEB SERVER     ) 108
  DIGITAL CERTIFICATE IS NOT DEFINED.  CERTIFICATE SERIAL NUMBER(77DD0
  FC28DEA887F56121235E11CA09) SUBJECT(graaff@us.ibm.com.CN=Paul de Gra
  ff.OU=Digital ID Class 1 - Microsoft.OU=Persona Not Validated.OU=www
  verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)98.OU=VeriSig
   Trust Network.O=VeriSign, Inc.) ISSUER(CN=VeriSign Class 1 CA Indiv
  dual Subscriber-Persona Not Validated.OU=www.verisign.com/repository
  RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=VeriSign Trust Network.O=VeriSi
  n, Inc.).
```

Figure 128. ICH408I error message indicating an unsuccessful map

Note: If the certificate is not defined, the OS/390 HTTP server regresses from `%%CERTIF%%` to `%%SAF%%`, and prompts for a RACF user ID and password.

3.4.3.2 Example 2

In this example we use the same scenario as in example 1, but now we are using four different client certificates for authentication and, by defining different filters, we can learn how the matching of filters occurs.

We have the following digital certificates with these subjects' distinguished names:

- `CN=Ted Anderson,ou=SMPO,o=IBM,c=US`
- `CN=Paul de Graaff,ou=ITSO,o=IBM,c=US`
- `CN=Pekka Hanninen,ou=IGS,o=IBM,c=FI`
- `CN=Jack Jones,ou=SNTP,o=IBM,C=US`
- `CN=Patrick Kappeler,ou=SNTP,o=IBM,c=FR`

These digital certificates are issued by the following Certificate Authority:

- `cn=Trust Authority CA,ou=Trust Authority,o=Your Organization,c=US`

Note: The distinguished name of the CA is the default name when you install IBM's Trust Authority product, which has been recently renamed to Tivoli SecureWay PKI.

We have a model digital certificate stored in the data set graaff.taja0434.cerbin that contains Jack Jones's digital certificate in a DER-encoded binary format.

We are now going to set up filters based on part of the subject's distinguished name and the full issuer's distinguished name.

We want to map all digital certificates from:

1. Organization IBM , country US and issued by our Trust Authority CA to RACF user ID IBMUS (see Figure 129 on page 99).

```
racdcert id(ibmus) map(taja0423.cerbin) sdnfilter('O=IBM.C=US') idnfilter('CN=

racdcert id(ibmus) listmap

Mapping information for user IBMUS:

Label: LABEL00000001
Status: TRUST
Issuer's Name Filter:
     >CN=Trust Authority CA.OU=Trust Authority.O=Your Organization.C=US<
Subject's Name Filter:
     >O=IBM.C=US<
```

Figure 129. RACDCERT MAP command: Example 1

Note: The issuer's distinguished name is retrieved from the model data set.

2. Organizational unit SNTP, organization IBM, country US and issued by our Trust Authority CA to RACF user ID SNTPUS (see Figure 130).

```
racdcert id(sntpus) map(taja0423.cerbin) sdnfilter('OU=SNTP.O=IBM.C=US')
idnfilter('CN=')

racdcert id(sntpus) listmap

Mapping information for user SNTPUS:

Label: LABEL00000001
 Status: TRUST
 Issuer's Name Filter:
      >CN=Trust Authority CA.OU=Trust Authority.O=Your Organization.C=US<
 Subject's Name Filter:
      >OU=SNTP.O=IBM.C=US<
```

Figure 130. RACDCERT MAP command: Example 2

3. Organizational unit SNTP, organization IBM, country FR and issued by our Trust Authority CA to RACF user ID SNTPFR (see Figure 131).

```
racdcert id(sntpfr) map(tapa0423.cerbin) sdnfilter('OU=SNTP.O=IBM.C=FR')
idnfilter('CN=')

racdcert id(sntpfr) listmap

Mapping information for user SNTPFR:

Label: LABEL00000001
 Status: TRUST
 Issuer's Name Filter:
      >CN=Trust Authority CA.OU=Trust Authority.O=Your Organization.C=US<
 Subject's Name Filter:
      >OU=SNTP.O=IBM.C=FR<
```

Figure 131. RACDCERT MAP command: Example 3

Note: Even though we specify the subject's distinguished name ourselves, a check is made in the model data set, whether the specification is correct. In our case we initially used Jack Jones's certificate, which had a C=US specification, where we needed a C=FR specification.

When we issued the command with Jack's model data set, we received the error shown in Figure 132.

```
IRRD141I The starting point specified for SDNFILTER is not found
in the certificate.
```

Figure 132. IRRD141I error message

We then exported Patrick's certificate in a DER-encoded binary format, stored it in data set graaff.tapa0423.cerbin, and used that data set instead, as shown in Figure 131 on page 99.

4. Country FI (Finland) and issued by any CA to RACF user ID FIUSER (see Figure 133).

```
racdcert id(fiuser) map sdnfilter('C=FI')

Mapping information for user FIUSER:

Label: LABEL00000001
 Status: TRUST
 Issuer's Name Filter:
      ><
 Subject's Name Filter:
      >C=FI<
```

Figure 133. RACDCERT MAP command: Example 4

After we have defined all the filters to be used in our example, we have to issue a SETROPTS RACLIST(DIGTNMAP) REFRESH to refresh the instorage profiles.

We now show the HTTPD log entries for the various digital certificates used and the RACF user ID that got assigned to show how each digital certificate gets a specific one assigned. It is no use showing the other screens because they are all the same as in example 1.

First we use Ted Anderson's digital certificate to see what RACF user ID he gets assigned. See the HTTPD log in Figure 134 on page 101.

.

```
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" ->
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx"
 Client certificate data:     1
    Common Name = Ted Anderson
    Country = US
    Locality = Knapp
    State = Wisconsin
    Organization = IBM
    Organizational Unit = SMPO
    Issuer Common Name = Trust Authority CA
    Issuer Country = US
    Issuer Organization = Your Organization
    Issuer Organizational Unit = Trust Authority
....
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
.....
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=IBMUS     2
```

Figure 134. HTTPD log showing Ted's mapping

1 The client certificate data of Ted Anderson.

2 The RACF user ID *IBMUS* is the selected one. This was a match on the first filter we set up, for anybody with an organization of IBM (O=IBM) and a country of US (C=US) and issued by our Certificate Authority (CA).

Next we use Jack Jones's digital certificate to go through the same scenario.Figure 135 on page 102 shows the HTTPD log for Jack's certificate mapping.

.

```
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" ->
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx"
Client certificate data:   1
   Common Name = Jack Jones
   Country = US
   Locality = Atlanta
   State = Georgia
   Organization = IBM
   Organizational Unit = SNTP
   Issuer Common Name = Trust Authority CA
   Issuer Country = US
   Issuer Organization = Your Organization
   Issuer Organizational Unit = Trust Authority
....
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
.....
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=SNTPUS    2
```

Figure 135. HTTPD log showing Jack's mapping

1 The client certificate data of Jack Jones.

2 The RACF user ID *SNTPUS* is the selected one.

This was a match on the second filter we set up, for anybody with an organizational unit of SNTP (OU=SNTP), an organization of IBM (O=IBM) and a country of US (C=US) and issued by our Certificate Authority (CA).

Note: The selection occurred here from more specific to less specific, although the first filter would have been a match, even if the more specific filter had not been defined.

In the third test, we use Patrick Kappeler's digital certificate to go through the same scenario again. Figure 136 on page 103 shows the HTTPD log for Patrick's certificate mapping.

.

```
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" ->
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx"
Client certificate data:   1
   Common Name = Patrick Kappeler
   Country = FR
   Locality = Montpellier
   Organization = IBM
   Organizational Unit = SNTP
   Issuer Common Name = Trust Authority CA
   Issuer Country = US
   Issuer Organization = Your Organization
   Issuer Organizational Unit = Trust Authority
....
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
.....
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=SNTPFR    2
```

Figure 136. HTTPD log showing Patrick's mapping

1 The client certificate data of Patrick Kappeler.

2 The RACF user ID *SNTPFR* is the selected one.

This was a match on the third filter we set up, for anybody with an organizational unit of SNTP (OU=SNTP), an organization of IBM (O=IBM) and a country of FR (C=FR) and issued by our Certificate Authority (CA).

In the last test, we use Pekka Hanninen's digital certificate to go through the same scenario again. Figure 137 on page 104 shows the HTTPD log for Pekka's certificate mapping.

.

```
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" ->
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx"
Client certificate data:   1
   Common Name = Pekka H\C8Anninen
   Country = FI
   Locality = Helsinki
   Organization = IBM
   Organizational Unit = IGS
   Issuer Common Name = Trust Authority CA
   Issuer Country = US
   Issuer Organization = Your Organization
   Issuer Organizational Unit = Trust Authority
....
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
.....
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=FIUSER    2
```

Figure 137. HTTPD log showing Pekka's mapping

1 The client certificate data of Pekka Hanninen.

2 The RACF user ID *FIUSER* is the selected one.

This was a match on the fourth filter we set up, for anybody with a country of FR (C=FR) and issued by any Certificate Authority (CA).

If we had used Paul de Graaff's digital certificate, we would have seen the same results as for Ted Anderson, because it would have matched as well on the first filter we defined, `O=IBM,C=US`.

3.4.3.3 Example 3

In the last example we use the ability to map a digital certificate to multiple RACF user IDs, depending on so-called "criteria". The criteria we use here are the application identifier (APPLID), also know as the VTAM application ID, and the system identifier (the SMF ID as defined in the SMFPRMxx member of the SYS1.PARMLIB).

Filters that contain criteria are also created with the RACDCERT MAP command, but use the MULTIID parameter rather then the ID parameter.

Our system (SMF) identifier is SC57 and we can use APPLID OMVSAPPL for our Web server. OMVSAPPL is a generic application ID for most OS/390 UNIX applications, so we can not distinguish between multiple web servers on one system.

CICS Transaction Server also supports the mapping of digital certificates to RACF user IDs. Here we can distinguish between various CICS regions, because each region will have it's own (VTAM) application ID.

Next we define a filter based on a digital certificate issued by our CA (used in example 2) with the added criteria of an applid and a system identifier, as shown in Figure 138.

```
racdcert multiid map(tapa0423.cerbin) idnfilter('CN=')
    criteria(APPLID=&APPLID.SYSID=&SYSID)

racdcert multiid listmap

Mapping information for MULTIID:

Label: LABEL00000001
 Status: TRUST
 Issuer's Name Filter:
      >CN=Trust Authority CA.OU=Trust Authority.O=Your Organization.C=US<
 Subject's Name Filter:
      ><
 Criteria:
      APPLID=&APPLID.SYSID=&SYSID
```

Figure 138. RACDCERT MULTIID MAP example

The APPLID and the SYSID criteria are added in the map definition as a variables. However, the APPLID criteria is passed in on the initACEE callable service. Also note that no RACF user ID is assigned in this map. So how does the RACF user ID get assigned? Well, the next step is to define the actual criteria. This is done by setting up profiles in the new RACF DIGTCRIT class using the RACF RDEFINE command.

Figure 139 shows the RACF commands to set up two selection criteria, one based on application ID (OMVSAPPL) and system ID (any *), the other specific for system ID SC57. The selected RACF user ID is specified in the APPLDATA field of the RACF RDEFINE command.

```
rdefine digtcrit APPLID=OMVSAPPL.SYSID=* appldata('webuser')

rdefine digtcrit SYSID=SC57 appldata('sc57user')

setr raclist(digtcrit) refresh
```

Figure 139. Setup of criteria

We use Paul de Graaff's digital certificate to go through the same scenario described in example 1. Figure 140 on page 106 shows the HTTPD log for Paul's certificate mapping.

.

```
Protect..... /jack2/readmvs.rexx matched "/jack2/readmvs.rexx" ->
Protection.. setup as defined in config file
Exec........ /jack2/* matched "/jack2/readmvs.rexx" -> "/u/jjones/cgi-bin/read
AuthCheck... Translated script name: "/u/jjones/cgi-bin/readmvs.rexx"
Client certificate data:   1
   Common Name = Paul de Graaff
   Country = US
   Locality = Poughkeepsie
   State = New York
   Organization = IBM
   Organizational Unit = ITSO
   Issuer Common Name = Trust Authority CA
   Issuer Country = US
   Issuer Organization = Your Organization
   Issuer Organizational Unit = Trust Authority ....
Accepted.... by Mask (no ACL, only Protect)
AA.......... check returned 200
HTHandle..... Access as "%%CERTIF%%" for Client "-unknown-"
.....
Environ..... SERVER_SOFTWARE=IBM HTTP Server/V5R2M0
Environ..... SERVER_NAME=wtsc57.itso.ibm.com
Environ..... _BPX_SPAWN_SCRIPT=YES
Environ..... _BPX_USERID=WEBUSER    2
```

Figure 140. HTTPD log showing Paul's mapping

1 The client certificate data of Paul de Graaff.

2 The RACF user ID *WEBUSER* is the selected one.

This was a match on the first filter we set up for anybody with a digital certificate issued by our Certificate Authority (CA). On the initACEE call the variables SC57 and OMVSAPPL were passed in to determine the RACF user ID. The first filter is the more specific and resulted in the RACF user ID WEBUSER being selected.

When using MULTIID filters, it is probably good security to add the restricted attribute to the RACF user ID that is selected to restrict access to only those resources that the user is meant to access.

For more information on Certificate Name Filtering, see the appropriate APAR information in member IRR40129 in the SYS1.SAMPLIB.

Chapter 4. OS/390 UNIX security enhancements

This chapter describes the new RACF features that help the administrator better protect the OS/390 UNIX environment.

The mapping of the OS/390 UNIX user identifiers (UIDs) and OS/390 UNIX group identifiers (GIDs) from OS/390 Version 2 Release 6 is also described, with suggestions on what values to use.

4.1 Mapping the UIDs and GIDs (UNIXMAP class)

The UNIXMAP mapping profiles are used to provide a cross-reference to RACF user and group profiles. These profiles provide RACF with a performance-sensitive method of returning information for a given UID or GID when requested by OS/390 UNIX or application programs.

We suggest that you *always* use these UNIXMAP mapping profiles when your OS/390 UNIX environment is in production.

RACF *automatically* creates a general resource profile named U*uid* in the UNIXMAP class when you define UID in the OMVS segment of a USER profile. The access list of the U*uid* profile contains the RACF user ID that has been assigned this UID. When you define a GID in the OMVS segment of a GROUP profile, RACF creates a general resource profile named G*gid* in the UNIXMAP class. In the access list of the G*gid* profile is the RACF group that has been assigned this GID.

RACF automatically maintains these mapping profiles when UIDs and GIDs are added, changed, or deleted with the ADDUSER, ALTUSER, DELUSER, ADDGROUP, ALTGROUP, or DELGROUP command. The UNIXMAP class does not have to be active for this to happen.

4.1.1 Assigning the UID and GID values

You can assign the GID and UID values with the following commands.

```
ADDGROUP OGRP11 OMVS(GID(121)) SUPGROUP(OMVSGRP) OWNER(OMVSGRP)

ADDUSER ALICE DFLTGRP(OGRP11) OMVS(UID(358)) OWNER(USERS) NAME('ALICE B')

ADDUSER BRUCE DFLTGRP(OGRP11) OMVS(UID(512)) OWNER(USERS) NAME('BRUCE S')
```

RACF creates a UNIXMAP profile named G121 with OGRP11 contained in the access list, and the profiles U358 with ALICE and U512 with BRUCE in the access list. If you issue the command:

```
ALTUSER BRUCE OMVS(UID(712))
```

RACF deletes the U512 profile and creates a U712 profile with BRUCE contained on the access list. If you try to list the profile U512 with the command:

```
RLIST UNIXMAP U512 ALL
```

you will get the error message (ICH13003I U512 NOT FOUND).

If you list the profile U712, you get the following output.

```
CLASS      NAME
-----      ----
UNIXMAP    U712

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    PEKKAH          NONE              NONE     NO

INSTALLATION DATA
-----------------
NONE

APPLICATION DATA
----------------
NONE

SECLEVEL
--------
NO SECLEVEL

CATEGORIES
----------
NO CATEGORIES

SECLABEL
--------
NO SECLABEL

AUDITING
--------
FAILURES(READ)

NOTIFY
------
NO USER TO BE NOTIFIED

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
BRUCE     NONE        000000

   ID     ACCESS  ACCESS COUNT  CLASS                   ENTITY  NAME
-------- -------  ------------ -------- ---------------------------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

In general, you should not alter these profiles. However, it is possible that they might get deleted inadvertently, or damaged by database corruption. If a profile is

deleted, or if the user is not contained in the profile access list, RACF will not be able to retrieve information for the UID or GID that the profile represented. RACF will then be unable to locate the mapping profile, and sends a return code to OS/390 UNIX indicating that the UID or GID is not known.

If this happens, an authorized user (a user with SPECIAL or CLAUTH(UNIXMAP) authority) needs to repair the damage. For example, if the profile U505 was deleted by accident and RACF user BILL can not login to OS/390 UNIX, you can correct this by entering the following commands:

```
RDEFINE UNIXMAP U505 UACC(NONE)

PERMIT U505 CLASS(UNIXMAP) ACCESS(NONE) ID(BILL)
PERMIT U505 CLASS(UNIXMAP) ID(authorized user-userid) DELETE
```

If the SETROPTS NOADDCREATOR option is in effect, the second PERMIT command is not necessary. For more information, see *OS/390 Security Server (RACF) Security Administrator's Guide,* SC28-1916.

Attention

Users who have a UID in their user profile and whose *default* or *current connect* group has a GID in the group profile can use OS/390 UNIX functions and can access OS/390 UNIX files based on the GID and UID values assigned.

Sometimes it is difficult to know what is a good value for the UID. We suggest that you use values that are unique and already in use, like employee serial numbers. A serial number is easy to find and can be used when you initially populate the UNIXMAP class. When you define UIDs for the technical user IDs, for example, use a range that is outside the serial numbers. Do not assign the same UID to multiple RACF users.

For the GID values we suggest that you do not use organization numbers because they normally change quite often. If you defined functional groups in RACF, they might serve as a starting point. You might start with 100, then 200, and so on. Choose the values so that you do not confuse them with the UID values. Remember that the users must have GID values in their default or current connect group to successfully access the OS/390 UNIX system.

4.1.2 Mapping to multiple user IDs and group names

Although assigning the same UID to multiple users is possible, it is not recommended. However, it may be necessary in some cases, such as superusers.

If you assign the same UID to multiple users, control at an individual user level is lost because the UID is used in OS/390 UNIX security checks. Users with the same UID assignment are treated as a single user during such checks.

You can determine which RACF user IDs are associated with a UID by looking at UNIXMAP profiles. For example, to see the RACF user IDs that are associated with UID 0, enter the command

```
RLIST UNIXMAP U0 ALL
```

Assigning the same GID to multiple RACF groups is not recommended, even though it is possible. The control at an individual group level is lost because RACF groups that have the same GID assignment are treated as a single group during OS/390 UNIX security checks.

Recommendation

When using the default OMVS segments in user and group profiles, multiple users will have the same UID. This should be used only as a migration aid. After the population of the UNIXMAP class, the default OMVS segment should be removed if possible.

4.1.3 Initial population of UNIXMAP class in a UNIX environment

The UNIXMAP class is not used for UID and GID lookups until you activate it at your installation. It should be left inactive until the following steps are performed to initially populate the UNIXMAP class with the information that may already exist in a user and group profiles in the database having OMVS segments. OS/390 UNIX can be active while the initial population takes place.

To populate the UNIXMAP class in an existing UNIX environment, do the following steps:

1. Quiesce administrative activity against users and groups.
2. Run the database unload utility (IRRDBU00).
3. Copy the REXX migration exec (IRR30858) from the SYS1.SAMPLIB to your own library and read the instructions at the beginning of the exec member. After reading and modifying the exec, run it against the database unload utility output. It produces a file containing RDEFINE and PERMIT commands that will populate the UNIXMAP class. A short example follows:

```
PROC 0
CONTROL NOLIST MSG NOCONLIST
RDEFINE UNIXMAP G205
PERMIT G205  CLASS(UNIXMAP) ACCESS(NONE) ID( IMWEB  )
RDEFINE UNIXMAP G121
PERMIT G121  CLASS(UNIXMAP) ACCESS(NONE) ID( OGRP11  )
RDEFINE UNIXMAP U712
PERMIT U712  CLASS(UNIXMAP) ACCESS(NONE) ID( BRUCE  )
```

4. Before executing the commands, issue SETROPTS NOADDCREATOR because you do not want the ID of the user who runs the command file produced in step 3 on the access list of all the profiles in this new class.
5. Execute the command file. When you execute this file, you may see messages ICH408I and ICH10102I, indicating that some profile is already defined to the UNIXMAP class. This occurs if a UID maps to more than one RACF user ID or if a GID maps to more than one RACF group name.

6. If SETROPTS ADDCREATOR was in effect prior to step 4, issue SETROPTS ADDCREATOR now to restore that setting.
7. Activate the UNIXMAP class issuing the command SETR CLASSACT(UNIXMAP) . The new UNIXMAP profiles will now be used to do UID and GID lookups. To maintain the performance, keep the UNIXMAP class active.

Attention

If you have not used OS/390 UNIX, you do not need to perform the previous steps. Instead you should determinate what values to use in the OMVS segment of RACF user ID and group profiles as mentioned before. After adding the OMVS segments, activate the UNIXMAP class.

4.1.4 OS/390 UNIX performance considerations

The Virtual Lookaside Facility (VLF) and the UNIXMAP class are used to map UIDs to RACF user IDs and GIDs to RACF group names. For RACF to begin using VLF for UID and GID mapping, you must define the IRRUMAP and IRRGMAP classes to VLF and VLF must be active.

Using the OS/390 Security Server option to cache additional OS/390 UNIX security information in VLF allows RACF to avoid accessing the RACF databases when called to create a security environment for OS/390 UNIX users. To use the cached user security (USP) packet, the IRRSMAP class must be defined to VLF.

Recommendation

To get the best performance in an OS/390 UNIX environment, always keep the UNIXMAP class and the VLF active.

4.2 OS/390 UNIX user limits

OS/390 UNIX has two types of users: a real person who enters the system with TSO logon or with *rlogin* from a UNIX workstation and a user that is really a server, supporting a large number of real users. The resource limits for the OS/390 UNIX users are specified in the BPXPRMxx member of SYS1.PARMLIB. These limits apply to all users except those with UID 0 (superuser authority).

Prior to OS/390 Version 2 Release 8 you have to assign the user limits in the BPXPRMxx member so that they support the servers' needs. The servers that are supporting a large number of real users have different system requirements. They need to consume more storage, use more CPU time, open more files, and have more tasks than a normal person does. This also allows the average user to consume large amounts of system resources. If everyone does this, it could have disastrous results, possibly requiring the system to be re-IPLed. The other alternative available is to set the system level limits to a reasonable level for the average user, and give the server applications superuser authority so they can exceed those limits. If the server is not entirely trusted code, this will compromise the security of the entire system.

In OS/390 Version 2 Release 8, new keywords have been added to the ADDUSER command to allow the additional fields to be added to the user's OMVS segment. These correspond to a subset of the limits in the BPXPRMxx parmlib member, but apply only to this user ID. ALTUSER has also been enhanced with these new keywords, and corresponding ones to remove the values. When a user limit is removed from a user profile, the system limits will again apply to that user.The r_admin callable service has also been updated to support the new keywords when maintaining the user's OMVS segment.

If the default OMVS segment is used (the profile BPX.DEFAULT.USER in the FACILITY class) to get the UID for the user, then the user limit values present in the default OMVS segment are also used.

Table 6 lists the new limits that may be set in the OMVS user segment, their corresponding names in the BPXPRMxx member in SYS1.PARMLIB, and the default values from the BPXPRMXX member in SYS1.SAMPLIB.

Table 6. The new limits in the OMVS user segment

User limit keyword in OMVS segment	BPXPRMxx parameter	Comments	Default value
CPUTIMEMAX	MAXCPUTIME	The MAXCPUTIME specifies the RLIMIT_CPU hard limit resource value processes receive when they are dubbed a process. RLIMIT_CPU indicates the CPU time, in seconds, that a process can use.	1000
ASSIZEMAX	MAXASSIZE	The MAXASSIZE specifies the RLIMIT_AS hard limit resource value that processes receive when they are dubbed a process. RLIMIT_AS indicates the address space region size.	41943040 (40 MB)
FILEPROCMAX	MAXFILEPROC	Specify the maximum number of files that a single user is allowed to have concurrently active or allocated.	64
PROCUSERMAX	MAXPROCUSER	Specify the maximum number of processes that a single user (that is, with the same UID) is allowed to have concurrently active regardless of origin.	25
THREADSMAX	MAXTHREADS	Specify the maximum number of threads that OS/390 UNIX will allow to be active concurrently in a single process.	200

User limit keyword in OMVS segment	BPXPRMxx parameter	Comments	Default value
MMAPAREAMAX	MAXMMAPAREA	Specify the maximum amount of dataspace storage (in pages) that can be allocated for memory mappings of HFS files. Storage is not allocated until memory mapping is active.	4096

Now you have the possibility to evaluate your installation's BPXPRMxx limits and set them to the level of the average user. For the users who require higher resource limits, you can set the individual values in the user OMVS segment. For example, if you want to give user BRUCE more CPU time and more address space size, you issue the following command:

```
ALTUSER BRUCE OMVS(CPUTIMEMAX(2400) ASSIZEMAX(62914560))
```

If you now use the LISTUSER command to display the OMVS segment information, you get the following output.

```
LISTUSER BRUCE OMVS NORACF

******************************** Top of Data ********************************
USER=BRUCE

OMVS INFORMATION
----------------
UID= 0000000712
HOME= /u/BRUCE
PROGRAM= /u/BRUCE/bin/myshell
CPUTIMEMAX= 0000002400
ASSIZEMAX= 0062914560
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
****************************** Bottom of Data ******************************
```

The initUSP and the initACEE callable services both return the new version of the output area OUSP (mapped by IRRPOUSP) to OS/390 UNIX. The OUSP will contain user limits from the OMVS segment. If the default OMVS segment is used, any user limit values present in the default segment will be returned in the OUSP.

The OUSP has a fixed size of 2074 bytes. When the version number of the OUSP is 1, a flag byte will follow the program path name, indicating if there are user limit values for this user. The second flag byte indicates if there was enough room for the user limits in the OUSP. If both HOME and PROGRAM values are 1023 bytes long, there is no room for the user limits. If only some of the user limit values will fit into OUSP, none will be returned, and the second flag byte will indicate that

there was not enough room. If the user limits do not fit in the OUSP, OS/390 UNIX will ignore them and use the system limit values for the user.

Recommendation

Check the length of the HOME and PROGRAM values in the user OMVS segment before adding the user limit values to make sure they can be used by OS/390 UNIX.

Superuser authority should be removed from users to whom it was assigned only for the purpose of exceeding the system limits.

Note: Before using the maximum CPU time value and the maximum address space size value from the BPXPRMxx member of PARMLIB, the values are compared to the values in the OS/390 System. If the OS/390 System value is greater than the one in BPXPBMxx, the OS/390 System value is used. See the *OS/390 UNIX System Services Planning*, SC28-1890 for more information.

4.3 Protected user IDs

Protected user IDs are protected from being used to log on to the system by any method that uses a supplied password, such as TSO logon, CICS signon, OS/390 UNIX `rlogin` or typical batch job submission. Protected user IDs can not be revoked through malicious or inadvertent incorrect password attempts. If the user attempts to use a protected user ID to enter the system with a password, or through an application that normally supplies a password, that attempt fails.

Recommendation

We suggest that you use protected user IDs for the started procedures associated with OS/390 UNIX, such as the kernel, the initialization started procedure, and important daemons that are critical to the availability of your OS/390 UNIX system.

4.3.1 How to define protected user IDs

You can define a protected user ID by specifying the NOPASSWORD and NOOIDCARD operands on the ADDUSER or ALTUSER commands. A protected user ID will have the PROTECTED attribute displayed in the output of the LISTUSER command. The r_admin callable service has also been updated to support the new keyword NOPASSWORD.

The following example shows a protected user ID being defined for a OMVS started task (`NOOIDCARD` is default).

```
ADDUSER OMVSSTC DFLTGRP(STCGROUP) OWNER(SCADMIN) NAME('UNIX TASK') NOPASSWORD
```

In the output of the LISTUSER command you can see the PROTECTED attribute:

```
USER=OMVSSTC  NAME=UNIX TASK             OWNER=SCADMIN   CREATED=99.173
 DEFAULT-GROUP=STCGROUP  PASSDATE=N/A     PASS-INTERVAL=N/A
 ATTRIBUTES=PROTECTED
 REVOKE DATE=NONE   RESUME DATE=NONE
 LAST-ACCESS=UNKNOWN
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
 -------------------------------------------
 ANYDAY                          ANYTIME
  GROUP=STCGROUP  AUTH=USE      CONNECT-OWNER=SCADMIN   CONNECT-DATE=99.173
    CONNECTS=    00  UACC=NONE     LAST-CONNECT=UNKNOWN
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
```

If you try to do a TSO logon, you will get the error message:

```
ICH408I USER(... ) LOGON/JOB INITIATION - INVALID PASSWORD
```

The administrator can revoke and resume the protected user ID by command. The protected user ID can also become revoked through inactivity. The protected user ID can be used to submit JCL and have the user ID propagated if enabled for the protected user ID. Also, the surrogate user IDs and the execution user IDs can be defined as protected user IDs.

To prevent a protected user ID from being used to log on, RACROUTE REQUEST=VERIFY and RACINIT processing will check if the protected user ID indicator is on in the user profile. If so, RACROUTE / RACINIT will be failed unless keywords such as PASSCHK=NO or START=procname have been specified to indicate that no password is needed on this processing.

The Accessory Environment Element (ACEE), mapped by IHAACEE macro, has a new bit value ACEENPWR, which indicates the protected user ID. The bit is located in ACEEFLG3.

Attention

If you share the RACF database, protected user IDs may be used to attempt logon from systems running OS/390 releases prior to OS/390 Version 2 Release 8. This may result in protected user IDs being revoked through malicious or inadvertent incorrect password attempts from downlevel systems. To avoid this, you should make sure that OS/390 Version 2 Release 8 is installed on all shared systems before defining protected user IDs.

Do not issue ADDUSER and ALTUSER commands specifying the PASSWORD/NOPASSWORD or OIDCARD/NOOIDCARD operands to administer protected user IDs from downlevel systems. If an existing protected user ID is administrated from a downlevel system using the ALTUSER command with these operands, the user ID may lose its PROTECTED attribute.

A `LISTUSER` command issued for a protected user ID from a downlevel system will not display the `PROTECTED` attribute.

4.4 Granularity of superuser privileges (UNIXPRIV class)

Many functions in the OS/390 UNIX environment require superuser authority, which is an “all or nothing” type of authority. In order for a user to perform any function which requires superuser authority, the user needs to have a UID(0) or the user must have READ access to the BPX.SUPERUSER profile in the FACILITY class, which allows the user to switch to UID of 0.

Using the new support in OS/390 Version 2 Release 8, you can give individual users the authority to perform specific superuser functions instead of giving them the authority to perform all superuser functions. This minimizes the number of assignments of superuser authority at the installation and reduces the security risk.

Since performance is a consideration for many OS/390 UNIX requests, there is a new class, UNIXPRIV, in the RACF class descriptor table.The UNIXPRIV class will be RACLISTed, so the profiles will be stored in a dataspace. Global access checking is not used for the authorization checking to UNIXPRIV resources.

Table 7 shows each resource name available in the UNIXPRIV class, the OS/390 UNIX privilege associated with each resource, and the level of access required to grant the privilege.

Table 7. Resource names in the UNIXPRIV class for OS/390 UNIX privileges

Resource name	OS/390 UNIX privilege	Access required
CHOWN.UNRESTRICTED [1]	Allows all users to use the `chown` command to transfer ownership of their own files	NONE

SUPERUSER.FILESYS [2]	Allows user to read any HFS file and to read or search any HFS directory.	READ
	Allows user to write to any HFS file and includes privileges of READ access.	UPDATE
	Allows user to write to any HFS directory and includes privileges of UPDATE access.	CONTROL (or higher)
Resource name	**OS/390 UNIX privilege**	**Access required**
SUPERUSER.FILESYS.CHOWN	Allows user to use the `chown` command to change ownership of any file	READ
SUPERUSER.FILESYS.MOUNT	Allows user to issue the `mount` command with the `nosetuid` option and to unmount a file system mounted with the `nosetuid` option.	READ
	Allows user to issue the `mount` command with the `setuid` option and to unmount a file system mounted with the `setuid` option.	UPDATE
SUPERUSER.FILESYS.QUIESCE	Allows user to issue the `quiesce` and `unquiesce` commands for a file system mounted with the `nosetuid` option.	READ
	Allows user to issue the `quiesce` and `unquiesce` commands for a file system mounted with the `setuid` option.	UPDATE
SUPERUSER.FILESYS.PFSCTL	Allows user to use the `pfsctl()` callable service.	READ
SUPERUSER.FILESYS.VREGISTER [3]	Allows a server to use the `vreg()` callable service to register as a VFS file server.	READ
SUPERUSER.IPC.RMID	Allows user to issue the `ipcrm` command to release IPC resources.	READ
SUPERUSER.PROCESS.GETPSENT	Allows user to use the `w_getpsent` callable service to receive data for any process.	READ
SUPERUSER.PROCESS.KILL	Allows user to use the `kill()` callable service to send signals to any process.	READ
SUPERUSER.PROCESS.PTRACE [4]	Allows user to use the `ptrace()` function through the dbx debugger to trace any process. Allows users of the **ps** command to output information on all processes. This is the default behavior of **ps** on most UNIX platforms.	READ
SUPERUSER.SETPRIORITY	Allows user to increase own priority.	READ

Notes:

1. See 4.4.2.1, "Using the CHOWN.UNRESTRICTED profile," on page 120.
2. Authorization to the SUPERUSER.FILESYS resource provides privileges to access only local Hierarchical File system (HFS) files. No authorization to access Network File System (NFS) files is provided by access to this resource.
3. The SUPERUSER.FILESYS.VREGISTER resource authorizes only servers, such as NFS servers, to register as file servers. Users who connect as clients through file server systems, such as NFS, are not authorized through this resource.
4. Authorization to the resource BPX.DEBUG in the FACILITY class is also required to trace processes that run with APF authority or BPX.SERVER authority. For more information about administering BPX profiles, see *OS/390 UNIX System Services Planning*, SC28-1890.

To get the new profiles checked, OS/390 UNIX invokes RACF callable services for authority checking. The callable services are using RACROUTE REQUEST=FASTAUTH calls. To increase performance, the FASTAUTH service is called directly from each callable service. As a result, the SAF router exit (ICHRTX00) will not be called. The pre-processing exit (ICHRFX03) and post-processing exit (ICHRFX04) will always be called, if present, instead of ICHRFX01 and ICHRFX02, even in non-cross-memory mode. Other exits, such as the SAF Callable Service Router Installation Exit (IRRSXT00), will be called as usual.

There are several auditing considerations for the UNIXPRIV class. If you use SETROPTS LOGOPTIONS for the UNIXPRIV class, the settings will be ignored. This is because RACROUTE REQUEST=FASTAUTH invocations do not honor SETROPTS LOGOPTIONS settings. Also, you can only audit successful accesses of UNIXPRIV resources. If you audit other successful OS/390 UNIX events with RACF classes such as PROCACT, FSOBJ, and IPCOBJ, you may see multiple records for the same operation if you also audit successes in the UNIXPRIV class.

Recommendation

Superuser authority should be removed from users if UNIXPRIV profiles can instead be used to grant specific superuser privileges.

4.4.1 Examples of authorizing superuser privileges

The following examples apply to the superuser privileges shown in Table 2, except the privilege associated with the CHOWN.UNRESTRICTED resource, which is discussed in 4.4.2.1, "Using the CHOWN.UNRESTRICTED profile," on page 120. For example, if you want to separate the operators' work, allowing one group to totally supervise the file-system and only one person to kill processes, you would perform the following steps:

1. If you have not defined the UNIXPRIV class to accept generic definitions, you first issue a SETROPTS command. Then define a profile to protect the resource called SUPERUSER.FILESYS.** and a profile to protect the resource called SUPERUSER.PROCESS.KILL, both in the UNIXPRIV class. For example:

```
SETROPTS GENERIC(UNIXPRIV)

RDEFINE UNIXPRIV SUPERUSER.FILESYS.** UACC(NONE)

RDEFINE UNIXPRIV SUPERUSER.PROCESS.KILL UACC(NONE)
```

2. Authorize the selected group and user as appropriate:

```
PERMIT SUPERUSER.FILESYS.** CLASS(UNIXPRIV) ID(FSOPGRP) ACCESS(CONTROL)

PERMIT SUPERUSER.PROCESS.KILL CLASS(UNIXPRIV) ID(HEADOP) ACCESS(READ)
```

3. Activate the UNIXPRIV class, if it is not currently active at your installation.

```
SETROPTS CLASSACT(UNIXPRIV)
```

Note: If you do not activate the UNIXPRIV class and activate SETROPTS RACLIST processing for the UNIXPRIV class, only the superusers will be allowed to supervise the file-system and kill processes.

4. You *must* activate SETROPTS RACLIST processing for the UNIXPRIV class, if it is not already active:

```
SETROPTS RACLIST(UNIXPRIV)
```

Note: If SETROPTS RACLIST processing is already in effect for the UNIXPRIV class, you must refresh SETROPTS RACLIST processing in order for new or changed profiles in the UNIXPRIV class to take effect:

```
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

4.4.2 Allowing OS/390 UNIX users to change file ownership

On OS/390 UNIX systems superusers can change the ownership of any file to any UID or GID on the system. Other users can only change the ownership of files that they own and only to one of their own associated GIDs. You can authorize all OS/390 UNIX users to transfer ownership of files they own to any

UID or GID on the system or you can authorize selected users to transfer ownership of any file to any UID or GID.

To allow selected OS/390 UNIX users to transfer ownership of any file to any UID or GID, create a profile in the UNIXPRIV class protecting a resource called SUPERUSER.FILESYS.CHOWN. Authorize selected users or groups to the profile. Activate the UNIXPRIV class if it is not currently active, then activate SETROPTS RACLIST processing. For example:

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHOWN UACC(NONE)

PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV) ID(CHFSGRP) ACCESS(READ)

SETROPTS CLASSACT(UNIXPRIV)

SETROPTS RACLIST(UNIXPRIV)
```

If you have UNIXPRIV class and the SETROPTS RACLIST processing already activated, you must refresh SETROPTS RACLIST processing in order for changes to take effect.

```
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

4.4.2.1 Using the CHOWN.UNRESTRICTED profile

To allow all OS/390 UNIX users to transfer ownership of files they own to any UID or GID on the system, create a discrete profile in the UNIXPRIV class called CHOWN.UNRESTRICTED. When this profile is defined on your system, all OS/390 UNIX users will be allowed to issue the `chown` command to transfer ownership of files they own. No access list is needed for this profile. RACF will check only for the existence of this profile.

To allow OS/390 UNIX users to transfer ownership for files they own, issue the following commands.

1. Define the discrete profile called CHOWN.UNRESTRICTED:

```
RDEFINE UNIXPRIV CHOWN.UNRESTRICTED
```

2. Activate the UNIXPRIV class, if it is not currently activated at your installation:

```
SETROPTS CLASSACT(UNIXPRIV)
```

Note: If you do not activate the UNIXPRIV class and activate SETROPTS RACLIST processing for the UNIXPRIV class, only the superusers will be allowed to transfer ownership of files to others.

3. You *must* activate SETROPTS RACLIST processing for the UNIXPRIV class, if it is not already active:

```
SETROPTS RACLIST(UNIXPRIV)
```

Note: If SETROPTS RACLIST processing is already in effect for the UNIXPRIV class, you must refresh SETROPTS RACLIST processing in order for the CHOWN.UNRESTRICTED profile to take effect:

```
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

4.5 OS/390 UNIX MOUNT with NOSECURITY keyword

For hierarchical file systems, you can use the MOUNT command to logically mount, or add, a mountable file system to the file system hierarchy. A mount user must have UID 0 or at least have READ access to the BPX.SUPERUSER FACILITY class in OS/390 Security Server Version 2 Release 6 or to the SUPERUSER.FILESYS.MOUNT resource in UNIXPRIV class in OS/390 Security Server Version 2 Release 8.

In OS/390 Version 2 Release 7 the MOUNT command has a new keyword, NOSECURITY, which means that security checking will not be enforced for files in this file system. A user may access or change any file or directory in any way, if the user has read authority to the mount point. Security auditing will still be performed if the installation is auditing successes.

Note: This support was made available in OS/390 Version 2 Release 7, but the RACF APAR OW33566 is for Version 2 Release 6, since RACF has no FMID for Version 2 Release 7. This APAR is included in Version 2 Release 8.

Chapter 5. LDAP Server

This chapter describes the system requirements and installation procedures for the LightWeight Directory Access Protocol (LDAP) Server. We assume that the reader has a basic understanding of LDAP; these concepts will not be repeated here. If a review of introductory LDAP concepts is needed, several recommended resources are as follows:

- *Understanding LDAP,* SG24-4986, which is a generic view of LDAP
- *Ready for e-business: OS/390 Security Server Enhancements,* SG24-5158, which is a S/390 implementation of LDAP at OS/390 2.4
- *LDAP Implementation Cookbook SG24-5110*, which is a non-S/390 implementation of LDAP

There are several major enhancements to the LDAP Server with OS/390 2.7 and OS/390 2.8. With these new releases of OS/390, there have been some changes in system requirements and, therefore, the installation procedures. This chapter covers three main topics:

- System requirements
- Installation of the system requirements
- Optional requirements and installation procedures based on the functions desired for your company's customized implementation

We also show you the enhancements introduced by APAR `OW41326` , such as the support for RACF key-rings and support for encrypting password values if stored in an OS/390 LDAP Server.

5.1 System requirements

A full description of the system requirements for the LDAP Server is in *OS/390 Security Server LDAP Server Administration and Usage Guide,* SC24-5861. For this discussion we will divide the system requirements for LDAP into three parts:

1. What is required to get the LDAP Server working in its most basic mode (the basic system requirements)
2. What is required for directory or database storage (the backend store requirements)
3. What is required for some of the optional features of the OS/390 LDAP Server

5.1.1 Basic OS/390 system requirements

There are four major requirements for the LDAP Server. They are:

1. UNIX System Services - The UNIX System Services must be at the OS/390 2.5 level or higher. It must be running in full function mode. The implementation of UNIX System Services is described in *OS/390 V2R6 UNIX System Services Implementation and Customization,* SG24-5178.
2. TCP/IP - This must be fully configured to support *tcp* traffic from the LDAP client to LDAP Server on the OS/390. The redbooks that describes the TCP/IP implementation are: *OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing,* SG24-5227;

OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 2: UNIX Applications, SG24-5228; *OS/390 eNetwork Communications Server Volume 3: MVS Applications,* SG24-5229.

3. Security product (RACF) - This is used to assign a user ID for the UNIX daemon (MVS STC) and protect the programs that the daemon will use.
4. The last major requirement is the backend store, the database that will be used as the LDAP directory. One of the major enhancements with OS/390 Security Server 2.7 and 2.8 is that there is a choice for the backend store. It can be either DB2 V5 or RACF, or both of them can be used together. The discussion of backend store is in 5.1.2, “Backend store requirements” on page 134.

5.1.1.1 Setting the RACF protection for the LDAP Server

Attention

The following steps are requirements to get the LDAP Server installed, not the steps that will make RACF the backend store for the LDAP Server. These steps could be implemented with any of the major security products that are available on the OS/390 platform today. These instructions use RACF as an example.

The LDAP Server must be associated with a RACF user and be authorized to certain OS/390 facilities such as the UNIX System Services. To do this with our RACF example, use the following steps:

1. If it is not already available, define a RACF GROUP that will be used for this UNIX daemon. The following command has the basic information required:

   ```
   AG ostc DATA('daemons') OMVS(GID(22)) OWNER(stcgroup) SUPGROUP(stcgroup)
   ```

   where the italic words are variables that can be set by the installation.
2. Create the RACF user ID for the LDAP Server. The following command can be used for the basic information:

   ```
   AU ldapsrv DATA('unix daemon') DFLTGRP(ostc) NAME('ldap server')
   OWNER(ostc) NOPASSWORD OMVS(uid(0) HOME('/'))
   ```

   where the italic words are variables that can be set by the installation.
3. Verify the user and group information is correct. Issue the following command and check for the appropriate information:

   ```
   LG ostc OMVS
   ```

   In the following output listing, note that the LDAP Server’s RACF user ID is listed as part of the RACF Group. Also note that the RACF GROUP has an OMVS segment and a GID has been assigned to the GROUP.

```
INFORMATION FOR GROUP OSTC
    SUPERIOR GROUP=STCGROUP     OWNER=STCGROUP
    INSTALLATION DATA=UNIX DAEMONS
    NO MODEL DATA SET
    TERMUACC
    NO SUBGROUPS
    USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
      LDAPSRV USE           000000               NONE
         CONNECT ATTRIBUTES=NONE
         REVOKE DATE=NONE                RESUME DATE=NONE
OMVS INFORMATION
----------------
GID= 0000000022
```

Then issue the following command:

```
LU ldapsrv OMVS
```

In the following sample output listing, note the default group for the LDAP Server's RACF user ID. Also, the LDAP Server should have the PROTECTED attribute assigned to it. Check the OMVS segment for the LDAP Server. The most important item is that the UID be set to 0(zero). The other field in the OMVS segment should be set according your individual installation requirements and standards.

```
 USER=LDAPSRV NAME=LDAP SERVER           OWNER=OSTC      CREATED=99.179
  DEFAULT-GROUP=OSTC      PASSDATE=N/A     PASS-INTERVAL=N/A
  ATTRIBUTES=PROTECTED
  REVOKE DATE=NONE   RESUME DATE=NONE
  LAST-ACCESS=UNKNOWN
  CLASS AUTHORIZATIONS=NONE
  INSTALLATION-DATA=UNIX DAEMON
  NO-MODEL-NAME
  LOGON ALLOWED    (DAYS)          (TIME)
  ---------------------------------------------
ANYDAY                           ANYTIME
  GROUP=OSTC      AUTH=USE      CONNECT-OWNER=OSTC      CONNECT-DATE=99.179
    CONNECTS=    00  UACC=NONE     LAST-CONNECT=UNKNOWN
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

OMVS INFORMATION
----------------
UID= 0000000000
HOME= /
```

4. The RACF user ID and GROUP that was defined for the LDAP Server needs to be associated to the LDAP Server Started Task (STC). This is done through the RACF STARTED class. Issue the following RACF command:

   ```
   RDEF STARTED LDAPSRV.** UACC(NONE) STDATA(USER(ldapsrv) GROUP(ostc))
   ```

   where *LDAPSRV* is the member name in the PROC library where the LDAP Server JCL is stored; *ldapsrv* is the RACF user ID and *ostc* is the RACF group that were created previously. The STARTED class is RACLISTed so the in

storage profiles will have to be refreshed. To refresh the STARTED profiles, issue the following RACF command:

```
SETR RACLIST(STARTED) REFRESH
```

5. For proper functioning and protection, the LDAP Server must be identified as a UNIX daemon and server. This is accomplished with the following RACF commands. If the *BPX.DAEMON* and *BPX.SERVER* profiles have not been set up, issue the following commands:

   ```
   RDEF FACILITY BPX.DAEMON UACC(NONE)
   RDEF FACILITY BPX.SERVER UACC(NONE)
   ```

   This will create the appropriate profiles in the RACF FACILITY class. Now the LDAP Server needs to be permitted to these profiles. To do this, issue the following commands:

   ```
   PE BPX.DAEMON CLASS(FACILITY) ACC(READ) ID(ldapsrv)
   PE BPX.SERVER CLASS(FACILITY) ACC(UPDATE) ID(ldapsrv)
   ```

   where *ldapsrv* is the RACF user ID created previously. Finally, if the FACILITY class is RACLISTed, the profiles will need to be refreshed. To do this, issue the following RACF command:

   ```
   SETR RACLIST(FACILITY) REFRESH
   ```

   BPX.DAEMON indicates that all programs executed by the LDAP Server must be RACF protected and fetched from a RACF-controlled data set. BPX.SERVER indicates the level of trust that is to be placed on the LDAP Server; READ indicating little trust (both server and client must have access to the resources that are being used) and UPDATE indicating a trusted server (only the client must have access to the resources that are being used).

6. The last part of the security setup is to insure that the executable programs which are required by the LDAP Server are protected and authorized correctly. The LDAP Server will always have to have access to its own DLLs. These are stored in *ldaphlq*.SGLDLNK. To protect the programs in this library, issue the following RACF command:

   ```
   RDEF PROGRAM ** ADDMEM('ldaphlq.SGLDLNK'//NOPADCHK)
   ```

   where *ldaphlq* is the high level qualifier (hlq) for the LDAP Server data sets. Refresh these RACF profiles with the command:

   ```
   SETR WHEN(PROGRAM) REFRESH
   ```

   There are a few other data sets that may need to be program controlled, based on the planned customization of the LDAP Server. Some of these data sets are:

   – C Runtime Library (default name - CEE.SCEERUN)

   – System SSL Load Library (default name - GSK.SGSKLOAD)

   – System Load Library (default name - SYS1.LINKLIB)

   – TCP/IP Load Library (default name - TCPIP.SEZALINK)

   The DB2 load library (default name of DSN510.SDSNLOAD) will also have to be program protected if the LDAP Server is using the RDBM. This is discussed in detail in a later section.

Attention

The RACF user ID, LDAPSRV in this case, will also need GRANT access to the DB2 plans and resources. Be sure to allow this user access to the DB2 CLI plans.

5.1.1.2 Basic LDAP Server system requirement planning

There are several basic questions that must be answered before the LDAP Server can be set up, configured, and started. Among the basic setup and operational issues that must be addressed for your LDAP Server are the following:

- Will the configuration files be stored in HFS or PDS? What is the HLQ? How is it protected?
- Will the server be started from PROC (auto ops) or /etc/rc (cron)?
- Are executables placed in LNKLST or STEPLIB? If STEPLIB, in JCL or ENVVARS?
- What are the names and number of LDAP servers? Multi/single server?
- What backend store will be used: DB2, RACF, or both?
- Are there referral LDAP servers? What are the referral configurations? What types of servers?
- Will the server be replicated?

There are some more advanced questions that also have to be considered as well, such as:

- How is the schema going to be defined?
- How are the ACLs going to be set up and maintained?

5.1.1.3 Storing the LDAP Server configuration files

During the customization of the UNIX System Services, we recommend that you move the configuration files out of the HFS that IBM ships with the system. The configuration files for the LDAP Server, SLAPD.CONF and SLAPD.ENVVARS, should be moved from the /usr/lpp/ldap/etc directory into a separate file or data set. How this is accomplished is determined by where you want to store the files.

A nice feature of the LDAP Server is that the configuration files can be stored in either UNIX files or in MVS data sets.

Configuration files stored in HFS

To move the configuration files to a separate HFS and store them as UNIX files:

1. Allocate a separate HFS and mount this within the UNIX file structure.

   Use the sample JCL in SYS1.SAMPLIB(BPXISHFS); use the example for the ETC HFS. The space requirements are very small unless more files are to placed in this HFS. Either identify the DFSMS classes in the JCL or let the ACS routines assign them.

2. Mount the new HFS on the UNIX file structure.

   – Make sure that the mountpoint already exits before trying to MOUNT the data set allocated above. This can be done using the following example:

```
mkdir /etc/ldap
```

– MOUNT the HFS. This can either be done with the sample JCL mentioned previously or by issuing the MOUNT command.

– Be sure that the person doing the MOUNT has one of the following:

- A UID of 0 (not recommended)
- READ access to the BPX.SUPERUSER profile in the RACF FACILITY class (and has switched themselves to UID of 0)
- If OS/390 Security Server R8 is installed, access to the SUPERUSER.FILESYS.MOUNT profile in the RACF UNIXPRIV class (see Chapter 4, “OS/390 UNIX security enhancements” on page 107 for more details on this).

3. Copy the configuration files into the new HFS, using the following commands as an example. This example uses the default directories; be sure to use your own directory names:

```
cp /usr/lpp/ldap/etc/slapd.conf /etc/ldap/slapd.conf
cp /usr/lpp/ldap/etc/slapd.envvars /etc/ldap/slapd.envvars
```

4. Optionally, the schema definitions can be moved to the installation’s separate HFS file too. This might be done if changes are being made to the schema. To do this, issue the following commands:

```
cp /usr/lpp/ldap/etc/slapd.* /etc/ldap/sldap.*
cp /usr/lpp/ldap/etc/schema.* /etc/ldap/schema.*
```

Note

The */etc/ldap* directory is the default directory set in the LDAP Server. If this (the default directory) needs to be changed to a different directory, an environment variable must be changed as well. This environment variable is LDAP_SLAPD_ENVVARS_FILE.
You can store and access files without changing the default directory by identifying the full path name.

Configuration files stored in PDS

To move the configuration files to an MVS data set:

1. Allocate (or identify) the MVS data set that will hold the configuration files. This can be either members within a PDS or a few sequential data sets. Following is the sample used in our tests. An LRECL of 80 and a RECFM of FB could have been used, but since the configuration files shipped with the LDAP Server were an LRECL of 255 and RECFM of VB, they were used.

```
//ALLOC   JOB  .... valid job statement stuff ....
//ETC     EXEC PGM=IEFBR14
//LDAPETC DD   DSN=JJONES.LDAP.ETCPDS,DISP=(,CATLG),
//  SPACE=(255,(300,100,10)),LRECL=255,RECFM=VB
//* This is enough space to copy of the configuration files into the
//* PDS without going into secondary extends.  If just the conf and
//* envvars files are being copied into the PDS, the SPACE can be
//* reduced.
```

2. Copy the UNIX files into the MVS data set. This can be done by:

– Using the ISHELL
Change to the /usr/lpp/ldap/etc directory and enter the `c` command next to the files that need to be copied into the MVS data set. When prompted, indicate that the file is to be moved to a data set and enter the data set name.

– Using TSO
Issue the following TSO command from the ISPF option 6 panel for each file that is to be copied:

```
OGET '/usr/lpp/ldap/etc/sladp.conf' 'JJONES.LDAP.ETCPDS(CONF)'
```

3. Be sure to edit the MVS data set after the copy. In several cases, unprintable characters are copied into the MVS data set. Find them with the `f P'.'` command under ISPF edit and change them to a blank.
4. If these data sets are set up with a new or different HLQ for the LDAP Server, be sure that these are protected with a RACF data set profile.

5.1.1.4 Setting up the LDAP Server JCL procedure

The LDAP Server can be started in one of two ways: it can be controlled from within the UNIX System Services or it can be controlled from the OS/390 started task procedure libraries. There are advantages and disadvantages to either method. This book will not discuss the automated methods of starting the LDAP Server, that is, AOC, cron, and so forth.

Starting the LDAP Server within UNIX System Services

To start the LDAP Server from UNIX System Services, use any of the following three methods:

1. Log on as the LDAPSRV user and issue the command:

   ```
   /usr/bin/slapd -f /etc/ldap/slapd.conf &
   ```

   The user must be defined with a valid password and will be in use for as long as the server is running. The directory paths may have to be changed to match your environment.

2. Add the following lines to the /etc/rc file

   ```
   echo LDAP Server starting, `date`
   # Start the OS/390 LDAP Server at initialization time
   _BPX_JOBNAME='LDAPSRV' /usr/bin/slapd -f /etc/ldap/slapd.conf &
   ```

   In this example, the LDAP Server will run with the jobname of LDAPSRV, but it will run under the RACF user ID of the OMVS kernal (usually OMVSKERN or OMVS). This might be the desired user ID.

3. Set up the `cron` daemon and start the LDAP Server as a daemon there. We will not describe this setup here.

The previous example might have to be changed to run in your environment. For example, the file name might have to be changed to point to the correct directory for the SLAPD.CONF file. Finally, it is the opinion of the writers that these examples are not the best methods for running the OS/390 LDAP Server in a production environment.

Starting the LDAP Server from the OS/390 procedure library

To start the LDAP Server from a started task procedure library, copy the sample LDAP Server JCL procedure from the installation samplib into the system started task procedure library. The LDAP Server ships a sample JCL procedure in the

hlq.SGLDSAMP. It is in member LDAPSRV. Starting the LDAP Server from a started task procedure library can be done using your favorite OS/390 job scheduler. The startup for the LDAP Server has to be after TCP/IP has completed its initialization.

After the JCL procedure is copied, the LDAP Server procedure has to be updated. The following sections will describe the options, using the listing in Figure 141 as an example. This is a copy of the sample JCL mentioned previously, without the comments except for key DD statements.

```
GLD.SGLDSAMP(LDAPSRV)
//LDAPSRV  PROC REGSIZE=64M,
//  PARMS='',
//  GLDHLQ='XXXXXX',
//  OUTCLASS='A'
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('/&PARMS >DD:SLAPDOUT 2>&1')
//STEPLIB   DD DSN=&GLDHLQ..SGLDLNK,DISP=SHR
//*CONFIG   DD DSN=<CONFIG.FILE.DATASET>,DISP=SHR
//*ENVVAR   DD DSN=<ENVVAR.FILE.DATASET>,DISP=SHR
//*DSNAOINI DD DSN=<DSNAOINI.DATASET>,DISP=SHR
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
```

Figure 141. Sample LDAPSRV JCL for the LDAP server started task

Positioning the LDAP Server data sets

The JCL that is used to start the LDAP Server should be updated to point to the new location of these configuration files.

Be sure that LDAP Server DLLs (that is, the load modules in the hlq.SGLDLNK data set) are either in your LNKLST or in a STEPLIB. The recommended method is to put the data set in the LNKLST. The other method of identifying the LDAP Server's DLLs is through the environment variables in the envvar file.

It is also recommended that the configuration files that are shipped in be moved from their installation directory. They can be moved to either an MVS data set or an HFS file.

If they are moved to an HFS file, for example /etc/ldap/slapd.conf, identify them in the appropriate DD statement using the `path` keyword in the DD statement. For example:

```
//CONFIG DD PATH='/etc/ldap/slapd.conf',...
```

If they are moved to an MVS data set, they can be moved to either a member of a PDS or a sequential data set. The source is a LRECL of 121 with a RECFM of V. If the target data set is the normal LRECL of 80 with a RECFM of F, you might get the normal message about difference in sizes. There is only one line that is longer than 72 and that is a comment line, so just use NONUM and the copy should be fine. Then update the JCL procedure to refer to the MVS data sets.

The last two DD statements to mention are:

- DSNAOINI points to the DB2 definitions for its Call Level Interface (CLI). This should be copied from the DB2 SAMPLIB, which is member DSNAOINI in the db2hlq.SDSNSAMP data set. There are unprintable characters in this file — do not change these characters or this file will not work. Also, this file is only needed if DB2 is being used. There are more details on this file later.
- SYSTCPD points to the TCPIP data file. This might not be required, depending on how you have the TCPIP naming conventions set up. The LDAP Server will require access to the TCPDATA member of your TCPIP STC whichever method is used.

Our customized JCL procedure for the LDAP Server is in Figure 142.

```
ITSO - SYS1.PROCLIB(LDAPPDS)
//LDAPPDS  PROC REGSIZE=90M,OUTCLASS='S',DEBUG='-d 65519'
//* CONFIG='-f /u/graaff/slapd.conf'
//*-------------------------------------------------------------------
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('&DEBUG')
//*        PARM=('/&DEBUG &CONFIG')
//*-------------------------------------------------------------------
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(STDCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//*DSNAOINI DD DSN=JJONES.LDAP.DSNAOINI,DISP=SHR
//*-------------------------------------------------------------------
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

Figure 142. Our LDAPPDS JCL to start the LDAP server as a started task

In this example, the DEBUG parm can be changed as needed, the DSNAOINI DD statement can be uncommented based upon whether DB2 is being used as a backend store, and the configuration file location can be altered as needed.

Customizing the LDAP Server files

This section discusses some of the base customization that might be needed in the three configuration files. More detailed configuration information can be found in the *OS/390 Security Server LDAP Server Administration and Usage Guide*, SC24-5861.

1. ENVVARS

```
/usr/lpp/ldap/etc/slapd.envvars
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
LANG=En_US.IBM-1047
```

Figure 143. Our SLAPD.ENVVARS file

As shown in Figure 142, we copied the slapd.envvars file to a member of our PDS `JJONES.LDAP.ETCPDS` called `STDENV`.

Note: The ENVVARS file is allocated in our started task JCL as a ddname of ENVVAR.

There is no reason to change this file, although we did change some environment variables. To do this, change the LEPARM in the JCL procedure, as described in the user's guide. We found the syntax of this very difficult and poorly documented. Fortunately, there is very little need to change anything in this area.

2. DSNAOINI

```
ITSO dataset name - 'DSN510.SDSNSAMP(DSNAOINI)'
; This is a comment line...
; Example COMMON stanza
ÝCOMMON"
MVSDEFAULTSSID=V51A

; Example SUBSYSTEM stanza for V42A subsystem
ÝV51A"
MVSATTACHTYPE=RRSAF
PLANNAME=DSNACLI

; Example DATA SOURCE stanza for STLEC1 data source
ÝSTLEC1"
AUTOCOMMIT=0
CONNECTTYPE=2

; Example DATA SOURCE stanza for STLEC1B data source
ÝSTLEC1B"
CONNECTTYPE=2
CURSORHOLD=0
```

Figure 144. Sample DSNAOINI file as supplied by DB2

This file is documented in the DB2 manuals; someone with DB2 skills is probably required. The limited knowledge that is required is:

- V51A in the example is the DB2 subsystem name.
- We changed RRSAF to CAF (Call Attach Facility).
- DSNACLI is the DB2 plan that must be used.
- STLEC1 is the DB2 location (DDF).

To find the DB2 location name, run the DB2 utility, DSNJU004, with the same data sets listed in the BSDS1 and BSDS2 DD statements in the DB2 started task JCL procedure. This will list the DB2 location name at the very bottom of the output.

3. CONFIG

```
/usr/lpp/ldap/etc/slapd.conf - global parms
include         /etc/ldap/slapd.at.system
include         /etc/ldap/slapd.cb.at.conf
include         /etc/ldap/slapd.at.conf
include         /etc/ldap/slapd.oc.system
include         /etc/ldap/slapd.cb.oc.conf
include         /etc/ldap/slapd.oc.conf
port             389
securePort       636
security         none
sslKeyRingFile   key.kdb
sslKeyRingFilePW none
sslCipherSpecs   12288
# maxthreads     nnn
# maxconnections nnn
# waitingthreads nnn
timelimit        3600
sizelimit        500
#adminDN        yourAdminDN
#adminPW        yourAdminPW
```

Figure 145. Sample SLADP.CONF file

Figure 145 shows the global parameters in the configuration file. These are the parameters that impact the general setup of the LDAP Server. The more specialized ones will be discussed later. The basic ones that will need to be uncommented and set up are those dealing with the threads (maxthreads, maxconnections, and waitingthreads). These parameters are described in the LDAP Server User's Guide.

```
/usr/lpp/ldap/etc/slapd.conf - database parms
database        rdbm GLDBRDBM
servername      yourDBserver
databasename    yourDBname
dbuserid        yourDBuserid
tbspaceentry    yourEntryTablespace
tbspace32k      your32KTablespace
tbspace4k       your4KTablespace
tbspacemutex    yourMUTEXTablespace
dsnaoini        yourCLIInitializationFile
suffix         "cn=localhost"
#suffix        yourCompanySuffix
index           cn              eq
index           ou              eq
index           sn              eq
index           telephoneNumber eq
index           title           eq
readOnly        off
```

Figure 146. Sample SLAPD.CONF file (continued)

Figure 146 shows the database parms contained in the sample SLAPD.CONF . They describe one of the backend stores for your LDAP Server. There can be one or more of these sets of database parms. The example above, which is sent with the LDAP Server, has the parms needed for a DB2 backend store. They are discussed in the following section, along with the RACF database parms.

5.1.2 Backend store requirements

The backend store is the database requirements for the LDAP Server, that is, where the contents of the directory are stored. The system requirements for the backend store changed with the introduction of OS/390 2.7. In OS/390 2.5 and OS/390 2.6, there was a requirement for DB2 V5 because this was the only backend store that the LDAP Server supported.

Starting with OS/390 2.7, RACF can be used as the backend store. The LDAP Server can run without DB2 being the backend store, if only RACF user and group information is needed. Or the LDAP Server can use both the RACF and DB2 databases as the backend store if information from both sources is needed. Of course, the LDAP Server can use just the DB2 database as the backend store as it always has, if RACF information is not needed.

Note: Starting with OS/390 2.8, both the LDAP client and server code is included in the SecureWay Security Server for OS/390 (formerly known as the OS/390 Security Server). Both the LDAP client and server code will come *enabled* so that a license for the Security Server is not required to use the LDAP code. This does not mean that databases required for the backend store are free to use. A license is required for either RACF or DB2 or both if they are to be used as the backend store.

This section describes how the LDAP Server is configured to use RACF and DB2 separately as the backend store. If both RACF and DB2 are used as the backend store, combine the steps described in this section. The LDAP Server can be configured to know if it has one or both of the RACF and DB2 databases based upon the database statements in the configuration file. The LDAP Server will access the appropriate database based on the Distinguished Name (DN) used in the LDAP request. An example of this will be given at the end of the section.

5.1.2.1 RACF backend store (SDBM)

When the LDAP Server is running with this configuration, there is no copying of information. The RACF database is being accessed from the LDAP Server. The appropriate RACF authority is still requested to access the RACF information. The correct authority includes making sure that the programs used in the RACF access are APF-authorized. When using both the RDBM and SDBM, the data sets requiring APF authorization are:

gldhlq.SGLDLNK The LDAP load library containing, among others, the LDAP-RACF interfaces

db2hlq.SDSNLOAD The DB2 load library containg the DB2 CLI interface

Note: The DB2 load library does *not* require APF authorization, when only the SDBM (RACF) backend is used.

The JCL procedure to start the LDAP Server with just the RACF backend store is shown in Figure 147. Note that it does not reference the DNSAOINI DD

statement. The configuration file will also have no reference to the DNSAOINI file. This is because this file describes the DB2 interface which is not needed if only the RACF backend store is used. Also, the gldhlq.SGLDLNK load module data set was in the LNKLST so the JCL procedure does not need a STEPLIB to this data set. The configuration and envvars files have been moved into the PDS, 'JJONES.LDAP.ETCPDS'. They are also listed below. Finally, the SYSTCPD DD statement is also used because of the TCPIP data set naming convention used at the ITSO.

```
//LDAPPDS  PROC REGSIZE=90M,OUTCLASS='S',DEBUG='-d 65519'
//*-------------------------------------------------------------
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('&DEBUG')
//*-------------------------------------------------------------
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(STDCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//*-------------------------------------------------------------
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

Figure 147. ITSO - LDAPPDS PROC

The envvars file was moved to the member, STDENV, in the 'JJONES.LDAP.ETCPDS' data set. It was *not* changed from the sample that was shipped with the LDAP Server.

```
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
LANG=En_US.IBM-1047
```

Figure 148. ITSO - 'JJONES.LDAP.ETCPDS(STDENV)'

The configuration file was moved to 'JJONES.LDAP.ETCPDS(STDCONF)' and updated to support the RACF only backend store.

```
include         /usr/lpp/ldap/etc/slapd.at.racf
include         /usr/lpp/ldap/etc/slapd.at.conf
include         /usr/lpp/ldap/etc/slapd.at.system
include         /usr/lpp/ldap/etc/slapd.oc.system
include         /usr/lpp/ldap/etc/slapd.oc.conf
include         /usr/lpp/ldap/etc/slapd.oc.racf
port            389

maxthreads      500
maxconnections 256
waitingthreads  100
timelimit       3600
sizelimit       500
adminDN         "racfid=JJONES,profiletype=user,sysplex=LOCAL"
database        sdbm GLDBSDBM
suffix          "sysplex=LOCAL"
```

Figure 149. ITSO - 'JJONES.LDAP.ETCPDS(STDCONF)'

In the configuration file, the `include` statements identify the information needed to define the LDAP schema. Initially we used the `slapd` files, which are the older definition files. Later we will show our example of using the newer `schema` files which are introduced in OS/390 2.8.

We used all the include files even though DB2 is not being used. It was our experience that GA OS/390 2.8 code, when we tried using just the RACF definitions, gave us an error message that a certain objectclass or attribute was not defined. The reason behind this is that the base schema definitions required by the RACF schema are in the other include files.

Note: You do *not* need to include the CB attributes and objectclass files when using only the RACF backend.

The other global parms used in this example indicate the port and capacity limitations to be used by this LDAP Server. These are explained in the LDAP Server User's Guide.

The last global parm is the `adminDN` user ID. In this case, we used a RACF user ID although a LDAP DN could have been used. Since a RACF user ID was used, the adminPW is not used since the password is stored in the RACF database. This is a more secure environment than the current handling of the LDAP passwords. We used and recommend use of the RACF passwords as much as possible. LDAP passwords by default are not encrypted, but RACF passwords are. A new parameter `pwEncryption` in the database section indicates how passwords are to be encrypted. For more information on storing encrypted passwords in LDAP see 5.4, "Encryption support for password values stored in LDAP" on page 188.

The database parms indicate that RACF is being used: `sdbm` indicates that a security database manager is going to be used and that the `GLDBSDBM` load module is to be used to handle the request. Finally, the `suffix` parm is the indicator in the DN that this is a RACF request. A requirement in the DN of RACF ID is the keyword of `sysplex=`; this is an OS/390 LDAP requirement. In our example, that is

the only item used in the suffix. There can be more items used in the suffix. For example, a suffix of `sysplex=smfid,ou=companyname,c=country` is valid.

When the LDAP Server is started with the following console command:

```
s ldappds
```

the following messages appear in the joblog of the `LDAPPDS` started task:

```
GLD0022I OS/390 Version 2 Release 8 Security Server LDAP Server
 Starting slapd.
GLD0010I Reading configuration file //DD:CONFIG.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.at.racf.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.at.conf.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.at.system.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.oc.system.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.oc.conf.
GLD0010I Reading configuration file /usr/lpp/ldap/etc/slapd.oc.racf.
GLD0002I Configuration file successfully read.
GLD0056I Non-SSL port initialized to 389.
GLD0122I Slapd is ready for requests.
```

Figure 150. Sample JOBLOG of the LDAP Server with an SDBM backend

Essentially this states that the LDAP Server is ready for non-secure requests on port 389. Also, this states that the configuration files were read from the DD statement in your JCL and lists the include statements within the configuration file. If the RACF include statements are reviewed, the objectclass and attributes for the RACF schema are defined.

```
objectclass racfUserOmvsSegment
 requires
  objectClass
 allows
  racfOmvsUid,
  racfOmvsHome,
  racfOmvsInitialProgram
```

Figure 151. Sample of RACF objectclass - RACF OMVS segment

This sample from the `/usr/lpp/ldap/etc/slapd.oc.racf` shows the objectclass definitions for the RACF OMVS segment. Each of these must be defined in the attribute file.

```
attribute racfOmvsUid           cis _nocreate 10   sensitive
attribute racfOmvsHome          cis _nocreate 1023 sensitive
attribute racfOmvsInitialProgram cis _nocreate 1023 sensitive
```

Figure 152. Sample of RACF attributes - RACF OMVS segment

The sample from the `/usr/lpp/ldap/etc/slapd.at.racf` shows the attributes for the objectclasses of the RACF OMVS segment. The three important items are:

- The _nocreate field, which tells LDAP not to allocate any storage for this field because it is in the RACF database.
- The length field, which matches up with the RACF database layout.
- The security attribute, which in the above example is always listed as `sensitive`. Other possible security attributes are `normal` and `critical`.

To verify that the LDAP Server is correctly accessing the RACF database, a couple of LDAP client command examples are shown. First, log on to TSO and get into OMVS, then issue the following UNIX command:

```
ldapsearch -h wtsc57.itso.ibm.com
-D  "racfid=JJONES,profiletype=user,sysplex=LOCAL" -w ????????
-b  "racfid=jjones,profiletype=user,sysplex=local"
"objectclass=*"
```

Figure 153. LDAPSEARCH example from USS accessing the RACF backend

As the LDAP client, this command searches the LDAP Server on the host identified in the `-h` parm. The client runs with the authority of the user (DN) in the `-D` parm. This is validated by the password in the `-w` parm. The `-b` parm identifies the start of the search or the item to be searched for, and the `objectclass=` indicates what is to be listed in the out. In this example, the RACF profile for a user (all the fields) is to be listed.

If everything is working correctly, you should get something that looks like the output shown in Figure 154 on page 139.

```
racfid=JJONES,profiletype=USER,sysplex=LOCAL
objectclass=racfUser
objectclass=racfBaseCommon
objectclass=racfBaseUserSegment
objectclass=racfUserOmvsSegment
objectclass=SAFTsoSegment
racfid=JJONES
racfProgrammerName=JACK JONES
racfOwner=racfid=SYS1,profiletype=USER,sysplex=LOCAL
racfAuthorizationDate=98.097
racfDefaultGroup=racfid=SYS1,profiletype=GROUP,sysplex=LOCAL
racfPasswordChangeDate=99.258
racfPasswordInterval=180
racfAttributes=SPECIAL
racfAttributes=AUDITOR
racfRevokeDate=NONE
racfResumeDate=NONE
racfLastAccess=99.299/21:02:04
racfClassName=NONE
racfInstallationData=NO-INSTALLATION-DATA
racfDatasetModel=NO-MODEL-NAME
racfLogonDays=ANYDAY
racfLogonTime=ANYTIME
racfConnectGroupName=racfid=SYS1,profiletype=GROUP,sysplex=LOCAL
racfConnectGroupName=racfid=OMVSGRP,profiletype=GROUP,sysplex=LOCAL
racfSecurityLevel=NONE SPECIFIED
racfSecurityCategoryList=NONE SPECIFIED
racfSecurityLabel=NONE SPECIFIED
racfOmvsUid=0000666666
racfOmvsHome=/u/jjones
racfOmvsInitialProgram=/bin/sh
racfomvsmaximumcputime=NONE
racfomvsmaximumaddressspacesize=NONE
racfomvsmaximumfilesperprocess=NONE
racfomvsmaximumprocessesperuid=NONE
racfomvsmaximumthreadsperprocess=NONE
racfomvsmaximummemorymaparea=NONE
SAFAccountNumber=ACCNT#
SAFDefaultLoginProc=IKJACCNT
SAFLogonSize=00000000
SAFMaximumRegionSize=00000000
SAFDefaultSysoutClass=X
SAFUserdata=0000
SAFDefaultCommand=%ispjcj
```

Figure 154. Sample LDAPSEARCH output from a RACF user ID

Note: The OMVS segment information might not show correctly in your installation. APAR `OW42613` is required to get the information shown in Figure 154.

The other LDAP client example is listed in Figure 155 on page 140. This example is run from TSO (ISPF option 6). To use these REXX commands, you need to have access to the *gldhlq*.SGLDEXEC.

```
ldapsrch -h wtsc57.itso.ibm.com -D "racfid=jjones,profiletype=user,sysplex
=local" -w ???????? -b "profiletype=group,sysplex=local" "objectclass=*" dn
```

Figure 155. LDAPSRCH command example from a TSO environment

This example will list the DNs for all the RACF groups as long as the RACF user ID used in the command has the RACF authority to display the group information, like RACF `SPECIAL` or `GROUP SPECIAL`. The output of our `LDAPSRCH` command is shown in Figure 156.

```
racfid=ADSM,profiletype=GROUP,sysplex=LOCAL
racfid=ALLUSERS,profiletype=GROUP,sysplex=LOCAL
racfid=AMS,profiletype=GROUP,sysplex=LOCAL
racfid=ANF,profiletype=GROUP,sysplex=LOCAL
racfid=ANO,profiletype=GROUP,sysplex=LOCAL
racfid=AOCCICS,profiletype=GROUP,sysplex=LOCAL
racfid=AOF,profiletype=GROUP,sysplex=LOCAL
racfid=APL2,profiletype=GROUP,sysplex=LOCAL
....
racfid=YELLWGRP,profiletype=GROUP,sysplex=LOCAL
profiletype=group,sysplex=LOCAL
```

Figure 156. LDAPSRCH results example

The example lists the RACF group names with the RACFID part of the DN and identifies them as RACF groups with the PROFILETYPE part of the DN. The rest of the DN is the suffix that was entered in the configuration file. The SYSPLEX part of the DN is required. The last line of the output identifies the search argument.

5.1.2.2 DB2 backend store (RDBM)

When the LDAP Server is running with this configuration, there are some DB2 requirements, as follows:

- There has to be a DB2 subsystem up and running; this section does not cover the steps to get this subsystem set up.
- Some DB2 tablespaces and tables must be set up for the LDAP Server; this section describes the allocation steps for these items.
- The CLI (Call Level Interface) must be set up; this section does not go through the setup of ODBC or CLI but does show the JCL to connect the LDAP Server to the CLI plan. The DB2 environment with ODBC and CLI and support for 32K tablespaces must be set up to proceed with the steps in this section.

As a reminder, the *db2hlq*.SDSNLOAD data set has to be RACF program protected since the LDAP Server is mentioned in the RACF FACILITY profile of BPX.DAEMON. This data set contains the DB2 CLI interface which the LDAP Server uses to communicate with the DB2 subsystem. If both RACF and DB2 are used as the backend store, then this data set must also be APF-authorized.

The following JCL procedure was used to start the LDAP Server with just the DB2 backend store. Note that it does reference the DNSAOINI DD statement. The configuration file could have had a reference to where the DNSAOINI file is. Also, the gldhlq.SGLDLNK load module data set was in the LNKLST, so the JCL

procedure does not need a STEPLIB to this data set. The configuration and envvars files have been moved into the PDS, 'JJONES.LDAP.ETCPDS'. They are also listed below. Finally, the SYSTCPD DD statement is also used because of the TCPIP data set naming convention used at the ITSO.

```
//LDAPSRV  PROC REGSIZE=64M,P1='-d 65519',
//  OUTCLASS='S'
//*-------------------------------------------------------
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('/&P1 ')
//*-------------------------------------------------------
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(NEWCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//DSNAOINI DD DSN=JJONES.ICF.DSNAOINI,DISP=SHR
//*-------------------------------------------------------
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

Figure 157. Sample JCL procedure to start our LDAP server

The envvars file was moved to the member, STDENV, in the 'JJONES.LDAP.ETCPDS' data set. It was *not* changed from the sample that was shipped with the LDAP Server, as shown in Figure 158.

```
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
LANG=En_US.IBM-1047
```

Figure 158. LDAP ENVVARS example

The configuration file was moved from /usr/lpp/ldap/etc/slapd.conf to 'JJONES.LDAP.ETCPDS(NEWCONF)' and updated to support the DB2 only backend store, as shown in Figure 159 on page 142.

```
include           /etc/ldap/slapd.at.system
include           /etc/ldap/slapd.at.conf
include           /etc/ldap/slapd.oc.system
include           /etc/ldap/slapd.oc.conf

port              389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         500

adminDN           "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US"
adminPW           paddle

database          rdbm GLDBRDBM
servername        CENTDB2
databasename      LDAP28
dbuserid          JJONES
tbspaceentry      LDAPTENT
tbspace32k        BIGTBLSP
tbspace4k         SMLTBLSP
tbspacemutex      MUTEXTBL
suffix            "cn=localhost"
suffix            "o=IBM_US,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
readOnly off
```

Figure 159. SLAPD.CONF example

In the configuration file, the `include` statements identify the information needed to define the LDAP schema. Initially we used the `slapd` files which are the older definition files. Later we will show our example of using the newer `schema` files which are introduced in OS/390 2.8.

The other global parms used in this example indicate the port and capacity limitations to be used by this LDAP Server. These are explained in the LDAP Server User's Guide.

The last global parm is the `adminDN` user ID. In this case, we used a standard LDAP DN. The configuration file also identified the `adminPW`. This is not the recommendation for a secured LDAP Server. If RACF is not being used, then the password can be stored in the LDAP Server once the administrator is defined and the LDAP password is stored in the LDAP directory, which would be recommended over storing the password in the configuration file.

APAR `OW41326` introduces encryption support for password values stored in the LDAP Server. See 5.4, "Encryption support for password values stored in LDAP" on page 188. Depending on your installation security policy, you can either use RACF or LDAP to identify and authenticate the administrator user ID.

The database parms indicate that DB2 is being used as the backend store, `rdbm` indicates that a relational database manager is going to be used and that the

GLDBRDBM load module is to be used to handle the request. All the parms between this database statement and the next database statement, if one exists, describe the DB2 environment for this LDAP directory. In this example, only one DB2 database was used, but more than one could have been used. For example:

Multiple databases

```
global parms
... include statments ...
... connection statements ...
... administration statements ...
database rdbm GLDBRDBM
... description for first DB2 database ...
database rdbm GLDBRDBM
... description for second DB2 database ...
database sdbm GLDBSDBM
... description for RACF database ...
```

Detailed descriptions of the database parameters are in the OS/390 Security Server LDAP Server Administration and Usage Guide. The DB2 parameters are here briefly described:

servername DB2 server location name
This is the DB2 DDF name and must match a DATA SOURCE stanza in the DSNAOINI file (an example of how to find this name is given later in this section).

databasename The name of the database this backend will use to store directory data (an example of how to find this name is given later in this section).

dbuserid An OS/390 user ID that will be the owner of the tables.

tbspace*xxx* These parameters are the labels for the various table spaces within the DB2 database.

suffix Denotes the root of a subtree in the namespace managed by this server within this backend. The `cn=localhost` suffix is used for parameters for the server (such as replica parameters). The other suffix is used to indicate the DNs that are within the LDAP directory for this server. In our example, all DNs start with `o=IBM_US,c=US`.

index This option specifies the indexing for fast retrieval to be maintained for the specified attribute or attributes. Indexing must be activated and `RUNSTATS` (in DB2) must run for this to take effect.

readonly This parameter sets the database to read only mode or update mode.

dsnaoini This parameter identifies the file that contains the CLI parameters. In our example, this parameter is not used because the DSNAOINI DD statement is used in the JCL procedure.

The last file that is required to set up the DB2 backend store is the DSNAOINI file. In our example, this is within the OS/390 data set JJONES.ICF.DSNAOINI. This is referenced in the JCL in the procedure, instead of being indicated by the dsnaoini parameter in the SLAPD.CONF file.

```
ÝCOMMON"
MVSDEFAULTSSID=DB51

ÝDB51"
MVSATTACHTYPE=CAF
PLANNAME=DSNACLI

ÝCENTDB2"
AUTOCOMMIT=0
CONNECTTYPE=1
```

Figure 160. Our sample DSNAOINI file

Most of this is a copy of the sample DSNAOINI file. The DB2 subsystem name and location will be needed to complete the information in this file correctly. Below is an example of how to find the required DB2 information.

To identify the DB2 subsystem name, if unknown, look in SYS1.PARMLIB(IEFSSNxx) for the SUBSYS SUBNAME(xxxx) that is the DB2 subsystem that is going to be used with the LDAP Server. The 'xxxx' indicates the subsystem name. In our example the subsystem name is `DB51`, shown in Figure 161. The figure also shows the delimiter that is used for that DB2 subsystem when MVS console commands are needed.

```
SUBSYS SUBNAME(DB51)
  INITRTN(DSN3INI)
  INITPARM('DSN3EP,=DB51,S')
SUBSYS SUBNAME(PSP)
SUBSYS SUBNAME(BP01)
SUBSYS SUBNAME(RACF)
  INITRTN(IRRSSI00)
  INITPARM('#,M')
```

Figure 161. Sample IEFSSNxx member

To get the DB2 server location, use the DSNJU004 program supplied by DB2. An example of the JCL we used to run DSNJU004, is shown in Figure 162.

```
//JJONESC JOB (999,POK),'Jack Jones',CLASS=A,REGION=4M,
//     MSGCLASS=Z,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//PRNT EXEC PGM=DSNJU004
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=S
//SYSUT1 DD DSN=DSN510.BSDS01,DISP=SHR
//SYSUT2 DD DSN=DSN510.BSDS02,DISP=SHR
```

Figure 162. Sample JCL for DSNJU004

This will produce a report that contains the DDF record information required by the LDAP Server. The top and bottom parts of the report are shown in Figure 163.The location is at the bottom of the report.

```
*************************************************************************
*
*             LOG MAP OF THE BSDS DATA SET BELONGING TO MEMBER 'NO NAME ' OF
*
*************************************************************************
RELEASE LEVEL OF BSDS -  ACTIVE=2.3 AND ABOVE   ARCHIVE=2.3 AND ABOVE   DDNAME
   LOG MAP OF BSDS DATA SET COPY 1, DSN=DSN510.BSDS01
   LTIME INDICATES LOCAL TIME, ALL OTHER TIMES ARE GMT.
         DATA SHARING MODE IS OFF
         SYSTEM TIMESTAMP   - DATE=1999.182  LTIME=15:57:50.74
         UTILITY TIMESTAMP  - DATE=1998.015  LTIME=16:14:53.30
         VSAM CATALOG NAME=DSN510
         HIGHEST RBA WRITTEN       000001A36C30  0000.000  00:00:00.0
         HIGHEST RBA OFFLOADED     000000000000
         RBA WHEN CONVERTED TO V4  000000000000
       THIS BSDS HAS MEMBER RECORDS FOR THE FOLLOWING MEMBERS:
         HOST MEMBER NAME:
           MEMBER ID:                 0
           GROUP NAME:
           BSDS COPY 1 DATA SET NAME:
           BSDS COPY 2 DATA SET NAME:
ACTIVE LOG COPY 1 DATA SETS
......
......
......
**** DISTRIBUTED DATA FACILITY ****
                        COMMUNICATION RECORD
                      20:18:00 JULY 02, 1999
LOCATION=CENTDB2 LUNAME=SC57DB51 PASSWORD=(NULL) GENERICLU=(NULL) PORT=446
DSNJ200I DSNJU004 PRINT LOG UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Figure 163. Sample DSNJU004 output

The DDF location name is required for the setup of the LDAP Server; in our example, this is CENTDB2. This is inserted into the DSNAOINI file at the appropriate place.

To create the tablespaces required for the LDAP Server, use the sample DB2 commands shipped with the LDAP Server in ***hlq***.SGDLSAMP(LDAPSPFI). We used a batch job in our example to create the tablespaces, as shown in Figure 164 on page 146.

```
// .... valid job statement .....
//DSNTIJR EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
     DSN SYSTEM(DB51)
        RUN PROGRAM(DSNTIAD)  PLAN(DSNTIA51) -
            LIBRARY('DB2V510O.RUNLIB.LOAD')
        END
//SYSIN    DD  *

        CREATE DATABASE LDAP28;

        CREATE LARGE TABLESPACE LDAPTENT IN LDAP28
             NUMPARTS 1 BUFFERPOOL BP32K;

        CREATE TABLESPACE SMLTBLSP IN LDAP28 SEGSIZE 4 BUFFERPOOL BP0;
        CREATE TABLESPACE BIGTBLSP IN LDAP28 SEGSIZE 4 BUFFERPOOL BP32K;
        CREATE TABLESPACE MUTEXTBL IN LDAP28 LOCKSIZE TABLESPACE
             BUFFERPOOL BP0;
//*
```

Figure 164. Sample JCL to create the LDAP tablespace

Figure 165 shows the JCL used to drop the LDAP tablespaces when needed.

```
//DSNTIJR EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB51)
  RUN PROGRAM(DSNTIAD)  PLAN(DSNTIA51) -
       LIBRARY('DB2V510O.RUNLIB.LOAD')
  END
//SYSIN    DD  *

  DROP TABLESPACE LDAP28.LDAPTENT;
  DROP TABLESPACE LDAP28.SMLTBLSP;
  DROP TABLESPACE LDAP28.BIGTBLSP;
  DROP TABLESPACE LDAP28.MULEXTBL;
  DROP DATABASE   LDAP28;
```

Figure 165. Sample JCL to drop the LDAP tablespaces

When the LDAP Server is started with the console command:

```
s ldappds
```

the messages shown in Figure 166 on page 147 will appear on the console and within the JESLOG of the started task.

```
GLD0022I OS/390 Version 2 Release 8 Security Server LDAP Server
 Starting slapd.
GLD0010I Reading configuration file //DD:CONFIG.
GLD0010I Reading configuration file /etc/slapd.oc.conf.
GLD0010I Reading configuration file /ldap/etc/slapd.at.conf.
GLD0010I Reading configuration file /ldap/etc/slapd.at.system.
GLD0010I Reading configuration file /ldap/etc/slapd.oc.system.
GLD0002I Configuration file successfully read.
GLD0056I Non-SSL port initialized to 389.
GLD0122I Slapd is ready for requests.
```

Figure 166. Sample JOBLOG of the LDAP server with a RDBM backend

Essentially this states that the LDAP Server is ready for non-secure requests on port 389. Also, it states that the configuration files was read from the DD statement in your JCL and lists the include statements within the configuration file.

A quick method to check the installation of the LDAP Server with this configuration is to issue the following LDAP query from the UNIX shell.

```
ldapsearch -h wtsc57.itso.ibm.com -D "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US"
-w paddle -b cn=localhost objectclass=*
```

Figure 167. Sample LDAPSEARCH command to check the LDAP installation

Using the definitions and configurations above, the result from the LDAP query are shown in Figure 168.

```
CN=LOCALHOST
objectclass=container
cn=localhost
```

Figure 168. Sample output of the LDAPSEARCH command example

This is the container for the replicaObjects. If it is specified in the `SLDAP.CONF` it can be queried.

5.1.2.3 An example of using both SDBM and RDBM

The LDAP Server at OS/390 R7 can use both DB2 and RACF as the backend store at the same time. This does not mean copying of data from one backend store to the other. The individual data is stored within its own backend store; that is, RACF data is stored within the RACF database and DB2 data is stored within the DB2 database.

To use both RACF and DB2 as the backend store, follow the same steps listed above for the RDBM (DB2) backend store. A few changes to the SLAPD.CONF file are necessary; they are shown in bold print in Figure 169 on page 148.

```
include           /etc/ldap/slapd.at.system
include           /etc/ldap/slapd.at.conf
include           /etc/ldap/slapd.oc.system
include           /etc/ldap/slapd.oc.conf
include /etc/ldap/slapd.at.racf
include /etc/ldap/slapd.oc.racf

port              389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         500

adminDN           "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US"
adminPW           paddle

database          rdbm GLDBRDBM
servername        CENTDB2
databasename      LDAP28
dbuserid          JJONES
tbspaceentry      LDAPTENT
tbspace32k        BIGTBLSP
tbspace4k         SMLTBLSP
tbspacemutex      MUTEXTBL
suffix            "cn=localhost"
suffix            "o=IBM_US,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
readOnly off

database        sdbm GLDBSDBM
suffix          "sysplex=LOCAL"
```

Figure 169. Sample SLAPD.CONF with both RDBM and SDBM defined

5.2 Optional OS/390 LDAP Server features

This section details OS/390 LDAP Server features that are optional, but are important features to have. We discuss the following features:

- Access control lists
- Secure Sockets Layer (SSL) support
- MultiServer support
- Referrals
- New schemas
- Replication

5.2.1 Access control lists

The LDAP Server allows for protection of the LDAP directory (that is, the DB2 database) and its entries. This is done through access control lists (ACLs). These

ACLs are associated with directory entries. The ACL protects that entry and possibly everything below that entry.

To understand how ACLs work, a good understanding of the LDAP hierarchical naming conventions is necessary. Every schema starts with a ‘root’ that is a beginning point for the directory. A basic schema (using predefined objectclasses and attributes) for IBM might be:

1. At the highest level, to separate into countries:

```
attribute c countryName cis c 128 normal (in the attribute file)
objectclass country (in the objectclass file)
   requires
      objectclass,
       c
   allows
   ...
```

2. Then, to define major organization within each country:

```
attribute o organizationName  is o 128 normal
objectclass organization
   requires
      objectclass,
       o
   allows
   ...
```

3. Next, to define organizational units within the major organization:

```
attribute ou organizationalUnit  is ou 128 normal
objectclass organizationUnit
   requires
      objectclass,
       ou
   allows
   ...
```

4. Finally, to identify each individual person:

```
attribute cn commonName cis cn 128 normal
objectclass person
   requires
      objectclass,
      sn,
      cn
   allows
   ...
```

Putting these parts of the schema together, a valid DN (Distinguished Name) could be *cn=Jack, ou=SNTC, o=IBM, c=US.* Under this DN there might be lots of information about this person, but, for our example, we are going to talk about three items that might be stored under this DN. They are:

```
attribute telephoneNumber tel  telephoneNumber  32 normal
attribute homePhone tel homePhone 32 sensitive
attribute userPassword bin userPassword 128 critical
```

The last attribute in each of these definitions is the sensitivity level of the data. *TelephoneNumber* is listed as normal, which usually means this is public information; *homePhone* is listed as sensitive, which usually means that only selective personnel can access this data. And *userPassword* is defined as

critical, which usually means that only the owner, administrator, and system can access this data. Only normal, sensitive, and critical are valid as access classes.

Once the data is identified by its level of sensitivity (done at schema definition time), identify which users or groups can have access (and the level of access) to each level. This is done with either the ldapcp utility or the ldif2db program. This section discusses and shows examples of the ldapcp utility.

The ldapcp utility is a UNIX executable that is shipped in the /bin directory. This is an interactive command. To use it you must either explicitly identify the path or be sure that it is in your `PATH` environment variable. Figure 170 shows an example of how to start an ldapcp session.

```
ldapcp -h wtsc57.itso.ibm.com -d "racfid=jjones,profiletype=user,sysplex=local"
-w ???????? -Z -k /u/jjones/secure/LdapClient.kdb -P racf <enter>

<enter security commands here>

quit
```

Figure 170. LDAPCP command example

This invokes the ldapcp environment in interactive line-mode. The highlighted parameters indicate the communication will be using SSL (the setup for SSL is discussed in 5.2.2, “SSL support” on page 153). The user ID and password used for authentication is a RACF user ID and password. This could be a non-RACF DN and password. If a user ID and password is not provided, you will be prompted for one.

This does not imply that ldapcp works with the RACF backend (SDBM). Indeed, ldapcp does not build ACLs for the SDBM (RACF), nor will it display information from the SDBM (RACF), such as user IDs in a RACF group. But RACF user IDs can be used in ACLs to protect access to data stored in the RDBM (DB2).

To end the ldapcp session, enter the `exit` or `quit` command.

We show some examples for protecting the *ou=SNTC, o=IBM, c=US* LDAP directory entry in Figure 171 on page 151. The commands shown are entered between the `ldapcp` command and the `quit` command displayed in Figure 170.

```
acl create "ou=SNTC, o=IBM, c=US" "true" \      1
"access-id:cn=SecurityAdmin, ou=Security,o=IBM, c=US:\
object:da:normal:rwcs:sensitive:rsc:
group:cn=Anybody:normal:rsc"

acl  create "ou=SNTC,o=IBM,C=US" \   2
"access-id:racfid=pekka,profiletype=user,sysplex=local:normal:r"

acl create "cn=Jack,ou=SNTC,o=IBM, C=US" "false" "access-id\   3
:cn=Paul,ou=ITSO,o=IBM, c=US:object:ad:normal:rwsc:\
sensitive:rwsc:critical:rwsc"

acl create "ou=SNTC, o=IBM, c=US" "false" \    4
"group:cn=SecurityGrp, ou=Security, o=IBM, c=US: \
object:da:normal:rwsc:sensitive:rwsc:critical:rsc"

acl query object "ou=SNTC, o=IBM, c=US"     5
```

Figure 171. LDAPCP command examples to create ACLs

In example 1, the entry protected is `ou=SNTC, o=IBM, c=US`. The "true" option indicates that this protection is to be propagated to entries below this directory entry. Therefore, `cn=Jack, ou=SNTC, o=IBM, c=US` is also protected by this ACL unless something special is done. It also identifies a user ID called `cn=SecurityAdmin, ou=Security,o=IBM,c=US` who is allowed to:

- Delete and add objects (object:da)
- Read, write, search, and compare objects that have an access class of normal (normal:rwsc)
- Read, search, and compare objects with an access class of sensitive (sensitive:rsc)

Attention

A backslash (`\`) is the continuation character for a UNIX command that is too long.

`cn=SecurityAdmin, ou=Security, o=IBM, c=US` has no access to critical data in this entry. Access must be specifically granted, that is, read is not granted just because write is granted. Read, write, search, and compare are the only accesses allowed on the access classes. To allow "creation" of objects, only add and delete is allowed.

The last item in example 1 is the way to identify a group, `cn=Anybody`, in an ACL. `cn=Anybody` is a special group that is similar to `UACC` in RACF. If the user is not listed on the ACL, and is not a member of any listed access group, then the user receives the permissions listed under the group: `cn=Anybody` ACL entry. If this ACL entry does not exist, access to the entry is completely denied.

Similar to RACF, every entry has an owner. The owner can be a user or a group and can be propagated to "sub entries". Owners have full access to the entry and, along with the administrator, are the only ones who can change the ACLs.

Example 2 shows how to add a RACF user ID, pekka, to the ACL. This user can only read normal data.

Example 3 covers just the `cn=Jack,ou=SNTC,o=IBM, C=US` entry and is not propagated (because of the "false"). This example gives `cn=Paul, ou=ITSO, o=IBM, c=US` complete control of the entry.

Example 4 shows how to add a group, `cn=SecurityGrp, ou=Security, o=IBM, c=US` to the ACL. This group needs to be defined with the `ldapcp` command. This is not to be confused with a RACF group, although RACF user IDs can be defined within the LDAP group.

The last statement, 5 in Figure 171 on page 151, is a request for a listing of the ACL that protects `ou=SNTC,o=IBM, C=US` . The output, written to `stdout`, is shown in Figure 172.

```
object = ou=SNTC,o=IBM, c=US
aclSource = ou=SNTC,o=IBM, c=US
aclPropagate = TRUE
acl = access-id:cn=SecurityAdmin, ou=Security,o=IBM, c=US:
object:da:normal:rwcs:sensitive:rsc
acl = access-id:racfid=pekka,profiletype=user,sysplex=local:normal:r
acl = group:cn=SecurityGrp, ou=Security, o=IBM, c=US:
object:da:normal:rwsc:sensitive:rwsc:critical:rsc
acl = group:cn=Anybody:normal:rsc
```

Figure 172. LDAPCP ACL QUERY command example output

The `ldapcp` command is also used to create groups within the LDAP directory. Figure 173 shows an example of how to create and modify a group definition.

Note: This example assumes the `ldapcp` session is already started.

```
group create \   1
"cn=SecurityGrp,ou=Security,o=IBM,c=US" \
"cn=Jack,ou=Security,o=IBM,c=US" \
"cn=Pekka,ou=Security,o=IBM,c=US"  \
"cn=Ted,ou=Security,o=IBM,c=US"

group delete member \  2
"cn=SecurityGrp,ou=Security,o=IBM,c=US" \
"cn=Ted,ou=Security, o=IBM, c=US"

group add member \ 3
"cn=SecurityGrp,ou=Security,o=IBM,c=US" \
"cn=Paul,ou=Security, o=IBM, c=US"

group list group "ou=Security, o=IBM, c=US"   4

group list member \    5
"cn=SecurityGrp,ou=Security, o=IBM, c=US"
```

Figure 173. LDAPCP command example to create groups

Example 1 creates a group called SecurityGrp with members Jack, Pekka, and Ted in it.

Command 2 deletes member Ted from the group. Command 3 adds Paul as a member to the group. Finally, all the groups within the organization unit, Security, are listed by command 4, and all members within the SecurityGrp group are listed by command 5. The output of these group list commands is shown in Figure 174.

```
group list
suffix = ou=Security, o=IBM, c=US
count = 1
groups = cn=SecurityGrp, ou=Security, o=IBM, c=US

member list
group = cn=SecurityGrp,ou=Security, o=IBM, c=US
count = 3
members = cn=Jack, ou=Security, o=IBM, c=US
________= cn=Pekka, ou=Security, o=IBM, c=US
________= cn=Paul, ou=Security, o=IBM, c=US
```

Figure 174. LDAPCP group list commands output

5.2.2 SSL support

The OS/390 LDAP Server supports Secure Sockets Layer (SSL) for secured communication between an LDAP client and an LDAP Server. To do this, digital certificates must be defined and stored correctly. Client certificate support is explained in a later section. This section explains the steps to set up a server certificate and utilize SSL in LDAP commands.

5.2.2.1 SSL setup using a UNIX key ring

A server certificate can be obtained from a certificate authority and stored in a key ring on the OS/390 system. This key ring is identified in SLAPD.CONF as the LDAP key ring and the desired certificate is marked as the default certificate within that key ring.

This example establishes a server certificate for the LDAP Server using System SSL. The UNIX System Services utility that uses System SSL is GSKKYMAN. Its functionality and setup instructions are explained in the *OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference*, SC24-5877. When you enter the GSKKYMAN command, you get the menu shown in Figure 175. Enter option **1** to create a new key ring for the LDAP Server.

```
JJONES:/etc/ldap/secure: >gskkyman

            IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number:
 ===> 1
```

Figure 175. GSKKYMAN menu option display

You will be prompted for the key ring database name and the password for the key ring. For production key rings, the password should not expire or should be created and maintained by RACF. The support for RACF key rings is introduced by APAR OW41326 and is described in 5.2.2.2, “SSL setup using a RACF key ring” on page 160.

```
Enter key database name or press ENTER for "key.kdb": LdapServer.kdb
Enter password for the key database.......>
Enter password again for verification.....>
Should the password expire? (1 = yes, 0 = no) Ý1¨: 0

The database has been successfully created, do you want to continue to work
with the database now? (1 = yes, 0 = no) Ý1¨:
 ===> 1
```

Figure 176. GSKKYMAN command example to create a key ring

Now that the key ring is created, it is used to set up the public-private key pair for the LDAP Server and to store the certificate request and digital certificates for the LDAP Server. The options that are available when working with a key ring are listed in Figure 177.

In a production environment, a digital certificate generated by a recognized Certificate Authority is probably desirable. In this case, option **3** would be appropriate for the new key ring. This would generate a public-private key pair and a request for a digital certificate would be built. The certificate request would then be shipped to the Certificate Authority for signing and shipped back to the requester (that is, the LDAP Server). The signed digital certificate would then be stored in the created key ring using option **4** and marked as the default certificate for this key ring. At that time, the digital certificate could be used by the LDAP Server for SSL support (that is, LDAP Server’s configuration file could point to that key ring).

```
Key database menu

Current key database is /etc/ldap/secure/LdapServer.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu):
 ===>
```

Figure 177. GSKKYMAN - key database menu options

In our test case, we are creating a self-signed certificate, using option **5**.

This option will ask several questions about the strength of the public-private keys and the DN for the digital certificate.

```
Enter option number (or press ENTER to return to the parent menu): 5
Enter version number of the certificate to be created (1, 2, or 3) Ý3¨:  3
Enter a label for this key.................> RacfServer
Select desired key size from the following options (512):
    1:    512
    2:    1024
Enter the number corresponding to the key size you want: 1
Enter certificate subject name fields in the following.
    Common Name (required).................> LDAP RACF Server
    Organization (required)................> IBM
    Organization Unit (optional)..........> ITSO
    City/Locality (optional)..............> Poughkeepsie
    State/Province (optional).............> NY
    Country Name (required 2 characters)..> US
Enter number of valid days for the certificate Ý365¨:  900
Do you want to set the key as the default in your key database?
(1 = yes, 0 = no) Ý1¨: 1
Do you want to save the certificate to a file? (1 = yes, 0 = no) Ý1¨: 1
Should the certificate binary data or Base64 encoded ASCII data be saved?
(1 = ASCII, 2 = binary) Ý1¨: 1
Enter certificate file name or press ENTER for "cert.arm": LdapServer.arm

Please wait while self-signed certificate is created...

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
 ===>
```

Figure 178. GSKKYMAN - creating a self-signed certificate

The question at 1 to save the certificate in an armored file (.arm) is not required and can be used for backup purposes.

To view the results of these actions, return to the GSKKYMAN - key database menu, as shown in Figure 179. Start GSKKYMAN and open the created key ring using option **2**, then enter option **1**, List/manage keys and certificates.

```
Key database menu

Current key database is /etc/ldap/secure/LdapServer.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 1
```

Figure 179. GSKKYMAN - key database menu

This results in a list of the labels for all the certificates and keys within the key ring, as shown in Figure 180.

```
Key and certificate list

Key database name is /etc/ldap/secure/LdapServer.kdb

Please choose one of the following keys to work with.

    1  - RacfServer
    2  - Integrion Certification Authority Root
    3  - IBM World Registry Certification Authority
    4  - Thawte Personal Premium CA
    5  - Thawte Personal Freemail CA
    6  - Thawte Personal Basic CA
    7  - Thawte Premium Server CA
    8  - Thawte Server CA
    9  - Verisign Test CA Root Certificate

Enter a key number or press ENTER for more labels:
 ===> 1
```

Figure 180. GSKKYMAN - key and certificate list

The one that was just created was *RacfServer*. In this case, select option **1** to select our certificate.

```
 Key Menu

Currently selected key: RacfServer

Choose one of the following options to proceed.

    1  - Show key information
    2  - Set the selected key as default
    3  - View certificate of the key
    4  - Remove trust root status
    5  - Copy the certificate of this key to a file
    6  - Delete the key
    7  - Export the key to another database
    8  - Export the key to a file

    0  - Exit program

Enter option number (or press ENTER to return to the parent menu):
 ===> 3
```

Figure 181. GSKKYMAN - key menu

Note: The selected label appears on the Key Menu, along with all the options that are available for that key. Since we want to view the certificate information, option **3** is selected.

Figure 182 on page 157 shows the certificate information for the RacfServer key. It contains the DNs for the requester and the signer. Since this is a self-signed certificate, these DNs are the same. The key is marked as the default certificate and is marked trusted.

```
Basic information of the currently selected key

                     Unique ID:    14
                        Label:    RacfServer
          Chosen as default key:    true
                      Key size:    512
                Set as trusted:    true
           Private key existence:    true
  User defined field existence:    false
               Private key type:    Software generated
            Private key version:    0
     Algorithm used to encrypt
private key information: PBEWithMD5AndDESCBC


     Certificate information for the selected key

                   Version:    3
             Serial number:    1f05096cf4415153
               Issuer name:
                               LDAP RACF Server
                               ITSO
                               IBM
                               Poughkeepsie, NY
                               US
              Subject name:
                               LDAP RACF Server
                               ITSO
                               IBM
                               Poughkeepsie, NY
                               US
            Effective date:    06/30/99
           Expiration date:    12/17/01
    Signature algorithm OID:    md5WithRSAEncryption
          Issuer unique ID:    false
         Subject unique ID:    false
       Number of extensions:    0

Press ENTER to return to the previous menu
```

Figure 182. GSKKYMAN - certificate view

Before leaving the GSKKYMAN utility, you have the ability to store the key ring password in a so-called “stash file”. From the main menu, enter option **11** and follow the instructions to create a stash file, as shown in Figure 183 on page 158. If you do not create a stash file, you can code the key ring password in the LDAP configuration file, using the `sslKeyRingFilePW` parameter. The directory and the stash file should be appropriately protected so that only authorized persons can access them.

Note: If you are using RACF to create the key ring and store the digital certificate within a RACF-protected key ring, these actions are not required.

```
 Key database menu

Current key database is /etc/ldap/secure/LdapServer.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 11

The encrypted password has been stored in file
/etc/ldap/secure/LdapServer.sth

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0":
```

Figure 183. GSKKYMAN - key database menu

Since this is a self-signed digital certificate that the LDAP Server is using, the client must have access to a copy of the digital certificate. This copy should be in PKCS12 format and stored in a location that is accessible by the clients. To create the digital certificate in the correct format, use option **8** (Export the key to a file) on the key menu of user ID (shown in Figure 184 on page 159) and export a copy of the digital certificate in a PKCS12 format into a file that has the permission bits set to 755. We use this file later when we create the key ring for the LDAP client.

```
 Key Menu

Currently selected key: RacfServer

Choose one of the following options to proceed.

    1  - Show key information
    2  - Set the selected key as default
    3  - View certificate of the key
    4  - Remove trust root status
    5  - Copy the certificate of this key to a file
    6  - Delete the key
    7  - Export the key to another database
    8  - Export the key to a file

    0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 8
Enter output PKCS12 file name or press ENTER for "key.p12": ldapserver.p12
Enter a password to protect the output PKCS12 file:
Enter password again for verification.....>

Please wait while the key is exported to a file...

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
```

Figure 184. user ID - key menu - export a key example

To complete the implementation of SSL for the LDAP Server, the LDAP configuration file, SLAPD.CONF, has to be updated. The SSL directives are global parameters and are positioned before the database directives.

```
..... place these in with the global directives .....
securePort      2636
sslAuth         serverAuth
security        ssl
sslKeyRingFile  /usr/lpp/ldap/etc/secure/LdapRacfServer.kdb
# sslKeyRingFilePW racf
sslKeyRingPWStashFile /usr/lpp/ldap/etc/secure/LdapRacfServer.sth
sslCipherSpecs  15104
..... database directives should follow the global directives .....
```

Figure 185. SLAPD.CONF - SSL directives

Note: The default SSL port for LDAP is 636.

The key ring password has been commented out because we use a password stash file. It can be used if desired, but the more secure method is using a password stash file.

When the LDAP Server is started, the messages shown in Figure 186 on page 160 should appear.

```
.....
GLD0052I Configuration read securePort 2636.
GLD0053I Configuration read security of ssl.
GLD0054I Value connectionsAllowed is set to 3.
GLD0002I Configuration file successfully read.
GLD0056I Non-SSL port initialized to 1389.
GLD0057I SSL port initialized to 2636.
.....
```

Figure 186. LDAP Server JOBLOG messages indicating SSL usage

Since this is a self-signed certificate, there will have to be some additional work done on the client side before the certificate can be trusted. The objective here is to get the OS/390 LDAP Server's certificate into the client's key ring. See 5.2.2.3, "SSL setup for the LDAP client" on page 162 for a description of this procedure.

5.2.2.2 SSL setup using a RACF key ring

APAR OW41326 for OS/390 Version 2 Release 8 introduces the use of a RACF key ring for the OS/390 LDAP Server. The RACF key ring is created using the RACDERT ADDRING command, described in 3.2.1, "RACDCERT ADDRING: Creating a key ring" on page 78.

Note: This is to replace the key-ring created with the user ID utility.

The user ID under which the LDAP Server runs (started task ID) must be authorized by RACF to use RACF key rings. In our example we used the following RACF commands:

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LISTRING
  CLASS(FACILITY) ID(LDAPSRV) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LIST
  CLASS(FACILITY) ID(LDAPSRV) ACCESS(CONTROL)
```

LDAPSRV is our started task user ID for the OS/390 LDAP Server.

Note: Remember to refresh the RACF FACILITY class after doing the permissions, using the SETROPTS RACLIST(FACILITY) REFRESH command.

Now we have to define the RACF key ring with the RACDCERT ADDRING command. The command used in our example is:

```
RACDCERT ID(LDAPSRV) ADDRING(ldap-wtsc57)
```

The owner of the RACF key ring has to be the user ID ldapsrv of the started task of the OS/390 LDAP Server. This is achieved by specifying the ID parameter on the RACDCERT ADDRING command.

This creates an empty key ring; we have to add the required certificates to the key ring next. Depending on your previous deployment of LDAP on OS/390, you may already have had a key ring (non-RACF) with certificates in it. To migrate these certificates over to RACF, you have to export them out of the UNIX key ring using the GSKKYMAN utility and then import them into RACF using the RACDCERT ADD command.

Figure 184 on page 159 shows an example of the export process for the certificate we need to add to the RACF database as well, using the RACF RACDCERT command.

Figure 187 shows the steps to add the exported certificate to RACF and connect it to the RACF keyring.

```
racdcert id(ldapsrv) add(racfserv.p12bin) withlabel('racfserv') password('paul')

racdcert id(ldapsrv) list

Digital certificate information for user LDAPSRV:

Label: racfserv
Status: TRUST
Start Date: 1999/06/30 16:23:45
End Date:   2001/12/17 16:23:45
Serial Number:
     >1F05096CF4415153<
Issuer's Name:
      >CN=LDAP RACF Server.OU=ITSO.O=IBM.L=Poughkeepsie.SP=NY.C=US<
 Subject's Name:
      >CN=LDAP RACF Server.OU=ITSO.O=IBM.L=Poughkeepsie.SP=NY.C=US<
 Private Key Type: Non-ICSF
 Private Key Size: 512
 Ring Associations:
 *** No rings associated ***

racdcert id(ldapsrv) connect (id(ldapsrv) label('racfserv') ring(ldap-wtsc57)
usage(personal) default

racdcert id(ldapsrv) listring(ldap-wtsc57)

Digital ring information for user LDAPSRV:

Ring:
     >ldap-wtsc57<
Certificate Label Name             Cert Owner     USAGE      DEFAULT
--------------------------------   ------------   --------   -------
racfserv                           ID(LDAPSRV)    PERSONAL     YES
```

Figure 187. RACF keyring setup example

Note: Both the key ring and the certificate must be owned by the started task user ID.

Now that the RACF key ring is set up and authorized, we need to update the LDAP configuration file SLAPD.CONF to specify the RACF key ring name for sslKeyRingFile (item 1 in Figure 188). The two other SSL key ring parameters, sslKeyRingFilePW (item 2) and sslKeyRingPWStashFile (item 3), should be commented out, so they revert to a NULL value, as shown in Figure 188.

```
..... place these in with the global directives .....
securePort      2636
sslAuth         serverAuth
security        ssl
sslKeyRingFile  ldap-wtsc57  1
#sslKeyRingFilePW           2
#sslKeyRingPWStashFile      3
sslCipherSpecs  15104
..... database directives should follow the global directives .....
```

Figure 188. SLAPD.CONF - SSL directives for RACF key rings

We are now ready to start our OS/390 LDAP Server with our RACF key ring. The messages that appear are no different then those in Figure 186 on page 160.

To see if LDAP actually uses the RACF key ring, we use the RACTRACE tool that was developed during another ITSO project. It can be downloaded from `ftp://www.redbooks.ibm.com/redbooks/GG243984`.

The RACTRACE entries show the opening of the RACF keyring (1) and the selection of the server certificate (2).

```
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTRING LDAPSRV.ldap-wtsc57     (1)
SEGM=CERTDATA
RACF  ROUTER3  FIELD=CERTCT   CERTRING
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTCERT 00.CN=WTSC57.ITSO.IBM.COM.OU=ITSO.O=IBM.
L=POUDATA
..............
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTCERT 866E7C8778E1A94B.CN=wtsc57.itso.ibm.com.
OU=i
RACF  ROUTER3  FIELD=UACC     APPLDATA
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTCERT 866E7C8778E1A94B.CN=wtsc57.itso.ibm.com.
OU=iTDATA
RACF  ROUTER3  FIELD=CERT     CERTPRVK CERTPRVT CERTPRVS
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTCERT 866E7C8778E1A94B.CN=wtsc57.itso.ibm.com.
OU=i
RACF  ROUTER3  FIELD=UACC     APPLDATA
RACF *ROUTER1  EXTRACT  REQSTOR=**NONE**       SUBSYS=**NONE**
RACF  ROUTER2  CLASS=DIGTCERT 866E7C8778E1A94B.CN=wtsc57.itso.ibm.com.
OU=iTDATA
RACF  ROUTER3  FIELD=CERT     CERTPRVK CERTPRVT CERTPRVS
RACF  ROUTER2  CLASS=DIGTCERT 1F05096CF4415153.CN=LDAP¢RACF¢Server.OU=
ITSOTDATA    (2)
RACF  ROUTER3  FIELD=CERT     CERTPRVK CERTPRVT CERTPRVS
```

Figure 189. RACTRACE entries for selection of RACF keyring and certificate

5.2.2.3 SSL setup for the LDAP client

Create a private directory for the client. Since this is on OS/390 Unix System Services, use the MKDIR command. Since this is a private key ring, a separate directory was created, /U/JJONES/SECURE, with the permission bits set so only the individual user could access it. Then issue the GSKKYMAN command. Figure 190 on page 163 shows the screen that appears within your UNIX shell.

```
 IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 1
Enter key database name or press ENTER for "key.kdb": LdapClient.kdb
Enter password for the key database.......>
Enter password again for verification.....>
Should the password expire? (1 = yes, 0 = no) Ý1¨: 0

The database has been successfully created, do you want to continue to work with
 the database now? (1 = yes, 0 = no) Ý1¨:
 ===> 1
```

Figure 190. GSKKYMAN main menu

This creates the certificate key ring and the password to protect the file. When we start working with this key, the main menu will appear. The objective is to get the LDAP Server's certificate into the client's key ring. To do this, use option **8**, Import Keys, as shown in Figure 191.

```
Key database menu

Current key database is /u/jjones/secure/LdapClient.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu):
 ===> 8
```

Figure 191. GSKKYMAN - key database menu

Import the certificate from the publicly accessible directory where the server has stored it, as shown in Figure 192 on page 164.

```
           Import Key Menu

Current key database is /u/jjones/secure/LdapClient.kdb

    1  - Import keys from another key database
    2  - Import keys from a PKCS12 file

    0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 2
Enter the PKCS12 file name: /etc/ldap/public/LdapServer.p12
Enter the password which protects the PKCS12 file:

Please wait while keys are imported from the file to database......

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
 ===>
```

Figure 192. GSKKYMAN - import key menu

Note: The approved clients have to know the location and the password for the LDAP Server's keys. Once the LDAP server has been added to the key ring, its keys can be viewed for validation and to be sure that they are marked as trusted.

Before the user exits GSKKYMAN, the user's stash file should be created as depicted in Figure 193.

```
 Key database menu

Current key database is /u/jjones/secure/LdapClient.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 11

The encrypted password has been stored in file /u/jjones/secure/LdapClient.sth

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨:
 ===>
```

Figure 193. GSKKYMAN - key database menu

Note: If you are using RACF for storage of the LDAP Server's certificate and the certificate was created using the RACDDCERT command, there is currently no way to export that certificate in a PKCS#12 format.

Now when the client wants to use SSL for an LDAP query, the appropriate parms must be coded. Figure 194 shows the LDAP query used before, but now we use the SSL parameters. The `-Z` and `-p` indicate that we are using SSL and the secure port. The `-K` parm identifies the client key ring, so we can verify the LDAP Server's digital certificate. And the `-P` parm is the key ring password so we can verify that this client is authorized to use the key ring.

```
ldapsearch -h wtsc57.itso.ibm.com -Z -p 1636
-D "racfid=jjones,profiletype=user,sysplex=local" -w ????????
-b "racfid=jjones,profiletype=user,sysplex=local"
-K /u/jjones/secure/LdapClient.kdb -P racf "objectclass=*"
```

Figure 194. LDAPSEARCH command example using SSL

Note: When ICSF services are RACF-protected using the `CSFSERV` class, the following permissions need to be granted:

```
permit CSFKGN class(csfserv) id(ldapsrv) access(r)
permit CSFPKI class(csfserv) id(ldapsrv) access(r)
permit CSFPKD class(csfserv) id(ldapsrv) access(r)

permit CSFDSV class(csfserv) id(user) access(r)
permit CSFPKE class(csfserv) id(user) access(r)

Note: ldapsrv is the User ID of the LDAP started task
Note: user is the User ID performing the LDAP-SSL session
```

When the `LDAPSEARCH` command in Figure 194 is issued and the debug option is used for the LDAP Server, some of the messages shown in Figure 195 on page 166 will be displayed in the LDAP Server's JOBLOG.

Note the following items:

- The server is listening on two sockets (1).
- SSL is set up before the user ID and password are validated (2).
- The data returned to the client is encrypted under SSL before it is passed back (3).
- Finally, the SSL socket is closed and the LDAP Server goes into wait mode (4).

```
 Listening on 2    (1)
 Listening on 3    (1)
ConnectionThread F56C288 taking new assignment for connection F53F998 (sd 4)
DataEncrypted entered, sd = 4
DataEncrypted routine, recv_rc (length of data returned) = 54
DataEncrypted YES
SSLSupport_SocketInit called (2)
Going to select on socket!!
SSLSupport_SocketRead called
SSLSupport_SocketRead called
SSLSupport_SocketRead called
Operation 1 added
do_bind       (3)
Bind operation requested by racfid=jjones,profiletype=user,sysplex=local.
=> dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
<= dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
do_bind:conn 6 version 2 dn (racfid=jjones,profiletype=user,sysplex=local) meth
Entered select_and_verify_backend
entered select_backend for dn=racfid=jjones,profiletype=user,sysplex=local
select_backend: selected sysplex=LOCAL
select_and_verify_backend: backend found
Exit select_and_verify_backend: be=F53D878, rc = 0, msg=NULLSTRBUF
Calling backend routine
entering sdbm_back_bind...
=> dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
<= dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
validate_binddn entered.
=> dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
<= dn_normalize "racfid=jjones,profiletype=user,sysplex=local"
sdbm_back_bind: username = JJONES

.......(lots of other messages from the LDAP Server).........

Sending msg to client
SSLSupport_SocketWrite called
conn=6 RESULT err=0 nentries=0
do_bind: exit conn->c_dn=racfid=jjones,profiletype=user,sysplex=local, conn->c_
Going to select on socket!!
SSLSupport_SocketRead called
SSLSupport_SocketRead called
SSLSupport_SocketRead called

.......(lots of other messages from the LDAP Server).........

Sending msg to client
SSLSupport_SocketWrite called
conn=6 RESULT err=0 nentries=0
sdbm_back_search exiting.
backend routine successful
Going to select on socket!!
SSLSupport_SocketRead called
SSLSupport_SocketRead called
SSLSupport_SocketRead called
Operation 3 added
do_unbind conn=6 op=2 fd=4
entering sdbm_back_unbind...
conn=6 op=2 closed errno=0
TerminateConn: Connection 4  being terminated
SSLSupport_SocketClose called     (4)
ConnectionThread F56C288 has finished ProcessConnection
ConnectionThread F56C288 waiting
```

Figure 195. LDAP Server's JOBLOG messages

5.2.3 MultiServer

Starting with OS/390 2.7, the LDAP Server supports the Parallel Sysplex environment. This means that two or more LDAP Servers can be using the same backend store. This is a major enhancement, especially in the area of availability. This can be set up in several different ways, but the two major methods are:

1. The different LDAP Servers are on the same OS/390 image as the shared backend store.
2. The different LDAP Servers are on different OS/390 images, in which case WLM and data sharing must be implemented for the shared backend store.

The third valid option is to run a combination of the two options.

Setting up a multiserver environment was not within the scope of this redbook project, so we have nothing to add to documentation provided by the LDAP manuals. However, there is an impact on our examples and discussion because of this major enhancement. Two of the items that must be mentioned are:

- The configuration file, SLDAP.CONF, now has some parameters that are only for the multiserver environment. These are *sysplexGroupName* and *sysplexServerName*, and to a lesser extent, *multiserver* and *tbspacemutex*. If these parameters are present in the configuration file, multiserver support might be turned on by mistake. This is especially true if sysplexGroupName and sysplexServerName are added to SLDAP.CONF with values. In this case, multiserver support is turned on, no matter what the other parameters indicate. This is true even if multiserver is set to **N** or **n**.
- If multiserver support is turned on, replication is not supported. LDAP Server instances operating in multiserver mode, either with or without dynamic workload management enabled, will not perform replication even if replication objects are present in the RDBM database. If replication is required, single-server mode must be used.

5.2.4 Referrals

You can use referrals to distribute the RDBM (DB2) backend store or to refer to other LDAP servers. In our test environment we did referrals from one OS/390 LDAP server to another OS/390 LDAP server to distribute the DB2 backend store. Both LDAP servers were on the same OS/390 image, but this could have been across a network without any changes to the configuration, except for the IP address.

In our example, we used the LDAPSRV server described previously as the root server. The setup is described in 5.1.2.2, “DB2 backend store (RDBM)” on page 140. Another LDAP server was added which was called LDAPNEW. The setup for LDAPNEW was similar to LDAPSRV. The JCL procedure for LDAPNEW started task is shown in Figure 196 on page 168.

```
//LDAPNEW PROC REGSIZE=90M,OUTCLASS='S',DEBUG='-d 0 '
//*-------------------------------------------------------------
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('&DEBUG')
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(NEWCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//DSNAOINI DD DSN=JJONES.REP.DSNAOINI,DISP=SHR
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

Figure 196. LDAPNEW JCL for the started task

Since these two LDAP servers were on the same OS/390 system, they used the same environment variables files, which are pointed to by the `ENVVAR DD` statement.

The DSNAOINI file is slightly different in that a different DB2 subsystem is used and different DB2 tables are built. This is indicated in the DSNAOINI file. The same SPUFI jobs are used as we described in building the LDAPSRV server, with changes to the DB2 subsystem name and tablespace names. The final result is a DSNAOINI file which is shown in Figure 197. This is pointed to by the DSNAOINI DD statement in the LDAPNEW JCL procedure shown in Figure 196.

```
ÝCOMMON"
MVSDEFAULTSSID=DB2O

ÝDB2O"
MVSATTACHTYPE=CAF
PLANNAME=DSNACLI

ÝDB2O"
AUTOCOMMIT=0
CONNECTTYPE=1
```

Figure 197. DSNAOINI file example

Finally, the SLDAP.CONF file for LDAPNEW has to be built. Here again, it looks similar to the configuration for LDAPSRV except for the references to the DB2 subsystem, which had to point to the new DB2 subsystem. The configuration file, which is pointed to by the `CONFIG DD` statement, is shown in Figure 198 on page 169.

```
include           /usr/lpp/ldap/etc/slapd.at.system
include           /usr/lpp/ldap/etc/slapd.at.conf
include           /usr/lpp/ldap/etc/slapd.oc.system
include           /usr/lpp/ldap/etc/slapd.oc.conf

port             1389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         500

adminDN           "cn=LDAPAdmin,ou=RACF_DEV,o=IBM,c=US"
adminPW           RACFDEV

database          rdbm GLDBRDBM
servername        DB2O
databasename      LDAPREP
dbuserid          JJONES
tbspaceentry      LDAPTREP
tbspace32k        BIGREP
tbspace4k         SMALLREP
tbspacemutex      MUTEXREP
suffix            "cn=localhost"
suffix            "ou=RACF_DEV,o=IBM,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
```

Figure 198. SLAPD.CONF file for LDAPNEW

At this point, the two LDAP servers have been configured and run as separate servers. Now the referral configuration needs to be set up. In this example, the design is very simple, but as more servers are added and/or the network becomes more complex, this design needs to be well architected. In this example, LDAPSRV is the root server and covers all the data with a suffix of `o=IBM,c=US`. LDAPNEW is the subordinate server, just covering the section of the data tree with a suffix of `ou=RACF_DEV,o=IBM,c=US`.

To set up the referral configuration, the root or higher-order LDAP server must understand where and what the subordinate LDAP server is. To do this, the referral objectclass is used. Figure 199 shows the LDAPADD command used to add the LDAPNEW referral information to the LDAPSRV server.

```
ldapadd -h wtsc57.itso.ibm.com -p 389
-D  "cn=LDAP Admin,ou=ITSO,o=IBM,c=US" -w paddle
-f /u/jjones/addref.mods


/u/jjones/addref.mods contains:

dn: o=IBM,c=US
objectclass: referral
ref: ldap://wtsc57.itso.ibm.com:1389/ou=RACF_DEV,o=IBM,c=US
```

Figure 199. LDAPADD command example to referral object

This `ref` directive indicates the subordinate LDAP server's IP address and the information that the subordinate LDAP server covers. In our example, this would be the LDAPNEW procedure and its data, `ou=RACF_DEV,o=IBM,c=US`.

The subordinate LDAP server should point back to a higher or more knowledgeable LDAP server. To do this the configuration file for the LDAPNEW server needs to be updated. Figure 200 shows the updated SLAPD.CONF file for the LDAPNEW server.

```
include           /usr/lpp/ldap/etc/slapd.at.system
include           /usr/lpp/ldap/etc/slapd.at.conf
include           /usr/lpp/ldap/etc/slapd.oc.system
include           /usr/lpp/ldap/etc/slapd.oc.conf

port             1389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         500

referral ldap://wtsc57.itso.ibm.com:389

adminDN           "cn=LDAPAdmin,ou=RACF_DEV,o=IBM,c=US"
adminPW           RACFDEV

database          rdbm GLDBRDBM
servername        DB2O
databasename      LDAPREP
dbuserid          JJONES
tbspaceentry      LDAPTREP
tbspace32k        BIGREP
tbspace4k         SMALLREP
tbspacemutex      MUTEXREP
suffix            "cn=localhost"
suffix            "ou=RACF_DEV,o=IBM,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
```

Figure 200. SLAPD.CONF file with referral update for LDAPNEW

The highlighted `referral` directive points back to the IP address of the LDAPSRV server.

Note: The port number is not required in this case since it was the default port for the LDAP server.

When the LDAP servers are started, they will refer to each other. An LDAP command such as:

```
ldapsearch -h wtsc57.itso.ibm.com -d "cn=LDAPAdmin,ou=RACF_DEV,o=IBM,c=US"
-w RACFDEV -b "ou=RACF_DEV,o=IBM,c=US" objectclass="*"
```

would be redirected by the LDAPSRV server to the LDAPNEW server (DB2 backend store) to get the appropriate data.

5.2.5 Using the schema files

With OS/390 R8, the LDAP Server comes with two sets of files that pre-define a default IBM schema. The first set is the files that have *slapd* in the file name, such as slapd.at.system and slapd.oc.system. These are the files that we have used in the include statements of our SLAPD.CONF examples. These slapd files were introduced with the OS/390 R5 version of the LDAP Server.

With OS/390 R8, these slapd files are included with the LDAP Server. There are also a series of attribute and objectclass definitions that are a superset of these `slapd` files. These definitions are in a series of files with *schema* in their file names, such as schema.system.at and schema.system.oc. With these additional definitions, you might be able to avoid having to customize the schema. But these new files require more DB2 space, so they are not set up as the default files.

Migrating to these new schema files is not difficult, but you will have to read and follow the instructions in the *OS/390 Security Server LDAP Server Administration and Usage Guide*, SC24-5861, to avoid an out-of-space condition within DB2.

More than likely, the space for DB2 will have to be increased to use the schema files because the extra definitions will create more tables. Figure 201 is the SPUFI example that was used to set up the ITSO's test environment.

```
EDIT       JJONES.SPUFI.CMDS(MAKEEM2) - 01.01            Columns 00001 00072
****** ***************************** Top of Data ******************************
000001
000002 CREATE DATABASE LDAPREP;
000003
000004 CREATE LARGE TABLESPACE LDAPTREP IN LDAPREP
000005        NUMPARTS 1 BUFFERPOOL BP32K;
000006
000007 CREATE TABLESPACE SMALLREP IN LDAPREP SEGSIZE 4 BUFFERPOOL BP0
000008        USING STOGROUP SYSDEFLT PRIQTY 720 SECQTY 720;
000009 CREATE TABLESPACE BIGREP   IN LDAPREP SEGSIZE 4 BUFFERPOOL BP32K;
000010 CREATE LARGE TABLESPACE LARGEREP IN LDAPREP
000011        NUMPARTS 1 BUFFERPOOL BP32K
000012        USING STOGROUP SYSDEFLT PRIQTY 720 SECQTY 720;
000013 CREATE TABLESPACE MUTEXREP IN LDAPREP LOCKSIZE TABLESPACE
000014        BUFFERPOOL BP0;
000015
****** **************************** Bottom of Data ****************************
```

Figure 201. SPUFI example to create the DB2 tables for the schema files

The `USING STOGROUP` on the `CREATE` statement allows for extra storage to be used for these DB2 tables. Ask your DB2 database administrator which `BUFFERPOOL` to use for these allocations.

If there is not enough space to allocate the required DB2 tables, an error message will be issued, as shown in Figure 202 on page 172.

```
GLD2003I Error code -1 from odbc string:  " SQLExecDirect " CREATE TABLE JJONES
NTGROUPTYPE VARCHAR(64) NOT NULL ) IN LDAPREP.SMALLREP .
native retcode = -904; state = "57011"; message = "{DB2 for OS/390}{CLI Driver}
 DSNT408I SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN
          UNAVAILABLE RESOURCE. REASON 00D70014, TYPE OF RESOURCE 00000220, AND
          RESOURCE NAME DB2V510O.DSNDBC.LDAPREP.SMALLREP.I0001.A001
 DSNT418I SQLSTATE   = 57011 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNXICTB SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = 190  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'000000BE'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 202. DB2 error message when enough space is not available

This error message indicates that the SMALLREP tablespace was the resource that needed to be increased. Another method to check how your DB2 space and extensions are being used is to list out the DB2 tablespaces using ISPF option 3.4, as shown in Figure 203.

```
DSLIST - Data Sets Matching DB2V510O.DSNDBD.L*                  Row 1 of 16
Command ===>                                                 Scroll ===> PAGE

Command - Enter "/" to select action               Tracks %Used XT  Device
------------------------------------------------------------------------------
        DB2V510O.DSNDBD.LDAPREP.ACLRINDE.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.BIGREP.I0001.A001       70    ?  18  3390
        DB2V510O.DSNDBD.LDAPREP.CPTRINDE.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LARGEREP.I0001.A001     30    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LDAPRDES.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LDAPRENT.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LDAPTREP.I0001.A001      2    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LDAP1HT5.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LDAP1OU5.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LONGCHAN.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.LONGENTR.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.MUTEXREP.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.OWNRINDE.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.RCTRIDRI.I0001.A001      1    ?   1  3390
        DB2V510O.DSNDBD.LDAPREP.SMALLREP.I0001.A001    225    ?   8  3390
        DB2V510O.DSNDBD.LDAPREP.SRCRINDE.I0001.A001      1    ?   1  3390
```

Figure 203. ISPF option 3.4 example -listing tablespace usage

Note the large number of extents in the BIGREP and SMALLREP tablespaces. These numbers are from the sample server that comes with the OS/390 LDAP Server with a few extra records. This is a rather small amount of data, so plan for the capacity of your data. As a general rule of thumb, using a large primary space with a medium size secondary space provides the best results.

After ensuring that there is enough space for the extra tablespaces, the SLAPD.CONF must be changed to point to the schema files, as shown in Figure 204 on page 173.

```
EDIT       JJONES.LDAP.ETCPDS(SCHCONF) - 01.07            Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 include         /etc/ldap/schema.user.at
000002 include         /etc/ldap/schema.user.oc
000003 include         /etc/ldap/schema.system.at
000004 include         /etc/ldap/schema.system.oc
000005 include         /etc/ldap/schema.IBM.at
000006 include         /etc/ldap/schema.IBM.oc
000007
000008 port             1389
000009 .....
       .....
000016 sslport 1636
       .....

000026 .....
000027 database          rdbm GLDBRDBM
000028 servername        DB2O
000029 databasename      LDAPREP
000030 dbuserid          JJONES
000031 tbspaceentry      LDAPTREP
000032 tbspace32k        BIGREP
000033 tbspace4k         SMALLREP
000034 tbspacemutex      MUTEXREP
000035 suffix            "cn=localhost"
000036 suffix            "o=IBM_US,c=US"
000037 index  cn   eq,sub
000038 index  ou   eq,sub
000039 index  sn   eq,sub
000040 readOnly off
       ......
```

Figure 204. SLAPD.CONDF example containing the SCHEMA files

All that is left is to reload the database. We use the `DB2LDIF` sample JCL shipped in *hlq*.SGLDSAMP, as shown in Figure 205, to unload the data in the “old” database in an LDAP Data Interchange Format (LDIF).

```
..... standard JOB statement .....
//DB2LDIF  EXEC PGM=GLDDB2LD,REGION=0M
//*-------------------------------------------------------
//DSNAOINI DD DSN=JJONES.ICF.DSNAOINI,DISP=SHR
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(OLDCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//*-------------------------------------------------------
//SYSPRINT DD DSN=JJONES.LDAPREP.LDIF,DISP=(,CATLG),
//   SPACE=(TRK,(1,1)),LRECL=255,RECFM=VB
//*-------------------------------------------------------
//CEEDUMP  DD SYSOUT=S
//SYSERR   DD SYSOUT=S
//STDOUT   DD SYSOUT=S
```

Figure 205. DB2LDIF JCL example

The DSNAOINI and CONFIG files should point to the old LDAP environment; that is, the old DB2 database that is to be unloaded with the old configuration definitions which point to the slapd files. The output will be placed in the SYSPRINT data set in LDIF . Be sure to provide enough space for all the data.

Figure 206 shows a sample JOBLOG from the DB2LDIF job.

```
..... messages from the job .....
GLD2054I No -o parameter supplied, writing to standard output.
GLD0010I Reading configuration file //DD:CONFIG.
GLD0010I Reading configuration file /etc/ldap/slapd.at.system.
GLD0010I Reading configuration file /etc/ldap/slapd.at.conf.
GLD0010I Reading configuration file /etc/ldap/slapd.oc.system.
GLD0010I Reading configuration file /etc/ldap/slapd.oc.conf.
GLD2062E The LDAP Server will operate in single-server mode.
GLD2065E Current DB schema version is 5.0.
```

Figure 206. DB2LDIF sample joblog

Figure 207 shows an example of what your output might look like in an LDIF format.

```
..... data in LDIF format .....
dn: CN=LOCALHOST
objectclass: container
cn: localhost

dn: cn=Hilding Landen, ou=Widget Division, ou=Austin, o=IBM_US, c=US
objectclass: organizationalPerson
cn: Hilding Landen
sn: Landen
telephonenumber: 1-914-433-1439
internationalisdnnumber: 293-1439
facsimiletelephonenumber: 1-914-433-0123
title: Certified Solutions Architect, PSS Sweden
postalcode: 1234
seealso: cn=Linda Carlesberg, ou=Austin, o=IBM_US, c=US
```

Figure 207. Example output of the DB2LDIF

Now the LDIF data needs to be loaded into the new DB2 environment with the new schema. To do this, run the LDIF2DB sample JCL, as shipped in *hlq*.SGDLSAMP. Figure 208 shows our example of the JCL that we used to load the data into the new database. The output from the unload, that is, the data set used as the `SYSPRINT` statement in Figure 207, is used as the input data set in the `SYSIN` statement here. The `SYSIN` statement that is commented out is an example of using a UNIX file to hold the LDIF data.

```
//LDIF2DB  EXEC PGM=GLDLD2DB,REGION=0M,
//         PARM=('/-d 65519')
//DSNAOINI DD DSN=JJONES.ICF.DSNAOINI,DISP=SHR
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(NEWCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//SYSIN    DD DSN=JJONES.LDAPREP.LDIF,DISP=SHR
//*SYSIN    DD PATH='/u/jjones/initload.ldif'
//SYSPRINT DD SYSOUT=S
//CEEDUMP  DD SYSOUT=S
//SYSERR   DD SYSOUT=S
//STDOUT   DD SYSOUT=S
```

Figure 208. LDIF2DB sample JCL

Figure 209 shows an example of the messages from a successful execution of the LDIF2DB program. The most important message in this case is the last one, which has the results of the LDIF2DB operation (1).

```
..... messages from the job .....
GLD2054I No -o parameter supplied, writing to standard output.
GLD0010I Reading configuration file //DD:CONFIG.
GLD0010I Reading configuration file /etc/ldap/slapd.at.system.
GLD0010I Reading configuration file /etc/ldap/slapd.at.conf.
GLD0010I Reading configuration file /etc/ldap/slapd.oc.system.
GLD0010I Reading configuration file /etc/ldap/slapd.oc.conf.
GLD2062E The LDAP Server will operate in single-server mode.
GLD2004I ldif2db: 55 entries have been successfully added out of 55 attempted.(1)
```

Figure 209. LDIF2DB JOBLOG example

5.2.6 Replication

Replication is one of the major enhancements introduced at this level of the LDAP Server. Replication provides availability and performance features. Availability is provided by having a concurrent running backup and performance is obtained because requests for information can be directed to either the master or the backup.

Single-Server Mode

It should be noted that the LDAP Server has to be running in single-server mode to support replication. The sysplexGroupName and sysplexServerName directives should not be added to the SLAPD.CONF file and the multiserver directive, if listed in the SLAPD.CONF file, should be set to N or n. If running in multiserver mode, similar benefits can be obtained without replication.

To set up replication, there must be:

- A master server that contains the master database. The master server is aware of all replicas. All changes are made through the master server and the master database. The master server propagates all changes to the replica servers.
- One or more replica objects. This is a different LDAP Server running with a *replicaObject* defined in it. The replica server knows about the master server. Its database is identical to the master database.

The master and replica can be separated across a network or they can be within the same image of the operating system, but they both must be running in single-server mode.

To set up a replication environment, we set up the LDAPSRV that we defined previously as the master. This was running exactly as described in 5.1.2.2, “DB2 backend store (RDBM)” on page 140. Another LDAP Server was set up with the same definitions. We called this LDAP Server LDAPRDB. The JCL for the procedure is shown in Figure 210 on page 176.

```
//LDAPRDB  PROC REGSIZE=90M,OUTCLASS='S',DEBUG='-d 0 '
//*-------------------------------------------------------------
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('&DEBUG')
///*------------------------------------------------------------
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(RDBCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//DSNAOINI DD DSN=JJONES.REP.DSNAOINI,DISP=SHR
//*-------------------------------------------------------------
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT  DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

Figure 210. LDAPRDB JCL example

The content of the configuration file JJONES.LDAP.ETCPDS(RDBCONF) listed in the `CONFIG DD` statement above is shown in Figure 211.

```
include           /usr/lpp/ldap/etc/slapd.at.system
include           /usr/lpp/ldap/etc/slapd.at.conf
include           /usr/lpp/ldap/etc/slapd.oc.system
include           /usr/lpp/ldap/etc/slapd.oc.conf

port             1389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         500

adminDN           "cn=LDAPRDB Admin,ou=ITSO,o=IBM,c=US"
adminPW           replica

database          rdbm GLDBRDBM
servername        DB2O
databasename      LDAPREP
dbuserid          JJONES
tbspaceentry      LDAPTREP
tbspace32k        BIGREP
tbspace4k         SMALLREP
tbspacemutex      MUTEXREP
suffix            "cn=localhost"
suffix            "o=IBM_US,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
```

Figure 211. Config file used by LDAPRDB

Note: The DB2 environment is different.

Since this is a different DB2 subsystem, the configuration file has to be updated. Since this could have been on a different system, the DB2 subsystem and tables could have been the same, if desired.

Note: If the servers are running on the same system, the TCP/IP ports used must be different.

Since these two servers are going to be replicas of one another, the *include* statements should reference the same type of schema definitions. For example, the config file in our example includes the slapd files in the LDAPSRV configuration. Then the SLAPD.CONF file for LDAPRDB should include the same slapd files instead of the schema files.

The DSNAOINI file for the LDAPRDB server is shown in Figure 212.

```
ÝCOMMON¨
MVSDEFAULTSSID=DB2O

ÝDB2O¨
MVSATTACHTYPE=CAF
PLANNAME=DSNACLI

ÝDB2O¨
AUTOCOMMIT=0
CONNECTTYPE=1
```

Figure 212. DSNAOINI file LDAPSRB

In our case, we were running both LDAPSRV and LDAPRDB on the same OS/390 image. But they are both going to run in single-server mode with their own RDBM environment.

Note: The SSID for the LDAPRDB is `DB20` and that LDAPSRV's DSNAOINI uses a SSID of `DB51`.

Since both servers are on the same system, we used the same ENVVARS file for them.

At this point, we should be able to successfully start and run both LDAP Servers at the same time. They are not replicas yet, but this is a good test to see if the configuration is set up correctly.

We designated LDAPSRV as the master LDAP server and we have to indicate to this server who is the replica server. We need to add this information to the LDAPSRV `replicaObject`. This is stored in the `cn=localhost` entry. It can be added using the LDAPADD command or by running the LDIF2DB JCL. Figure 213 on page 178 shows the LDAPADD command used and the LDIF data to define the replicaObject. This command is targeted to the LDAPSRV started task, which we defined at the beginning of this chapter.

```
ldapadd -h wtsc57.itso.ibm.com -p 389
-D  "cn=LDAP Admin,ou=ITSO,o=IBM,c=US" -w paddle
-f /u/jjones/addrep.mods

/u/jjones/addrep.mods contains:

dn: cn=LDAPRDB,cn=localhost
cn: LDAPRDB
objectclass: replicaObject
replicaHost: wtsc57.itso.ibm.com
replicaBindDN: racfid=JJONES,profiletype=USER,sysplex=LOCAL
replicaCredentials: ????????
replicaPort:  1389
replicaUseSSL:  FALSE
description: "LDAPRDB is the replica for LDAPSRV."
```

Figure 213. LDAPADD command example to add replicaObject

The replica LDAP Server required definitions to point to the master LDAP Server. The following statements are added to the `slapd.conf` file for this server to identify it as the *replica* and who is the *master server*:

masterServer Specifies the location of the replica's master server in LDAP URL format.

masterServerDN Specifies the DN allowed to make changes to the replica. The presence of this option indicates that the server instance using this configuration file is a slave.

masterServerPW Specifies the password for the masterServerDN that will be allowed to make the updates.

If the master server is using SDBM (RACF) backend then the masterServerDN can be a RACF user ID and no password will be needed in the SLAPD.CONF file. If RACF is being used, remember that SDBM(RACF) information is not be replicated. Only RDBM information is replicated.

Note: Replication of RACF information is achieved by using the RACF RRSF function.

Figure 214 on page 179 shows the additions we made for our configuration.

```
...
the rest of the slapd.conf file
...
database          rdbm GLDBRDBM
servername        DB2O
databasename      LDAPREP
dbuserid          JJONES
tbspaceentry      LDAPTREP
tbspace32k        BIGREP
tbspace4k         SMALLREP
tbspacemutex      MUTEXREP
suffix            "cn=localhost"
suffix            "o=IBM_US,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
masterServer      ldap://wtsc57.itso.ibm.com:389
masterServerDN    "racfid=JJONES,profiletype=USER,sysplex=LOCAL"
# masterServerPW - not needed if RACF is used
```

Figure 214. SLAPD.CONF file for the slave LDAP Server

The key to set this up is to make sure that in the master server, the replicaObject contains the valid replicaHost (the IP address of the replica), the replicaPort, and a valid replicaBindDN and replicaCredential to access the required data. For the replica server, a valid masterServer (the IP address of the master) and masterServerDN with password must be provided.

Now for the final setup step, be sure that both servers are not running. Copy the master database into the replica database using the DB2LDIF and LDIF2DB JCL to be sure that they are starting with the same data. Figure 215, and Figure 216 on page 180, show JCL examples of using the LDAPSRV and LDAPRDB configuration files when running DB2LDIF and LDIF2DB.

```
..... standard JOB statement .....
//DB2LDIF  EXEC PGM=GLDDB2LD,REGION=0M
//*-------------------------------------------------------
//DSNAOINI DD DSN=JJONES.ICF.DSNAOINI,DISP=SHR
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(STDCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//*-------------------------------------------------------
//SYSPRINT DD DSN=JJONES.LDAPREP.LDIF,DISP=(,CATLG),
//   SPACE=(TRK,(1,1)),LRECL=255,RECFM=VB
//*-------------------------------------------------------
//CEEDUMP  DD SYSOUT=S
//SYSERR   DD SYSOUT=S
//STDOUT   DD SYSOUT=S
```

Figure 215. DB2LDIF example using LDAPSRV config files

```
..... standard JOB statement .....
//LDIF2DB  EXEC PGM=GLDLD2DB,REGION=0M,
//         PARM=('/-d 65519')
//DSNAOINI DD DSN=JJONES.REP.DSNAOINI,DISP=SHR
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(RDBCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//SYSIN    DD DSN=JJONES.LDAPREP.LDIF,DISP=SHR
//SYSPRINT DD SYSOUT=S
//CEEDUMP  DD SYSOUT=S
//SYSERR   DD SYSOUT=S
//STDOUT   DD SYSOUT=S
```

Figure 216. LDIF2DB example using LDAPRDB config files

Now start both LDAPSRV and LDAPRDB. To test the replication setup, issue an LDAPSEARCH command against either LDAPSRV or LDAPRDB. For example, the following search is against the master server. By changing the port number, the same results should appear.

```
ldapsearch -h wtsc57.itso.ibm.com -p 389
-D "racfid=jjones,profiletype=user,sysplex=local" -w ????????
-b "cn=Pekka Hanninen,ou=Austin,o=IBM_US,c=US"
"objectclass=*"

with the following results ...

cn=Pekka Hanninen,ou=Austin,o=IBM_US,c=US
cn=Pekka
sn=Pekka
objectclass=organizationalperson
objectclass=person
title=ITSO Redbook Writer and FrameMaker Expert
```

Figure 217. LDAPSEARCH command example

In this way, availability and performance can be added to the LDAP environment.

The concern with this arrangement might seem to be insuring that the data is always matching. When an object is added, it is added first to the master and then replicated. So, executing the command shown in Figure 218 did exactly what is expected: it added the information to the master database and then replicated the information to the replica server. LDAPSEARCH commands were used to show the information that was added was the same on both databases.

```
ldapadd -h wtsc57.itso.ibm.com -p 389
-D "racfid=jjones,profiletype=user,sysplex=local" -w ????????
-f /u/jjones/repadd1.mod
```

Figure 218. LDAPADD command example to make replication happen to slave

The LDAPADD command, shown in Figure 219 on page 181, adds something to the LDAP replica server. Following the trace information, this was rerouted to the master server (via the masterServer information), added to the master database and then replicated.

```
ldapadd -h wtsc57.itso.ibm.com -p 1389
-D "racfid=jjones,profiletype=user,sysplex=local" -w ????????
-f /u/jjones/repadd2.mod
```

Figure 219. LDAPADD command example to make replication happen to master

Testing with one or the other LDAP server being down or unavailable was done as well. In these cases, the information was saved and when the missing LDAP server was recycled, the data was updated without intervention from the operators. During the time of the attempted update, there will be warning messages issued to indicate the server not available.

5.3 Migrating from previous LDAP releases

This section discusses migration from prior releases of LDAP. If you had LDAP running already with an RDBM backend, your DB2 databases need to be migrated to the new LDAP V2R7 or V2R8 schema before they can be used by the new LDAP Server.

To migrate your DB2 database to the new schema, execute the `LDAPSPMG` member located in *HLQ.*SGLDSAMP. To execute this file you need to access the DB2 Interactive panels. The DB2 Interactive panel, as shown in Figure 220, allows for execution of the LDAP migration script by using option 1, SPUFI.

```
DB2I PRIMARY OPTION MENU                                    SSID: DB51
COMMAND ===>  1

Select one of the following DB2 functions and press ENTER.

 1  SPUFI                  (Process SQL statements)
 2  DCLGEN                 (Generate SQL and source language declarations)
 3  PROGRAM PREPARATION    (Prepare a DB2 application program to run)
 4  PRECOMPILE             (Invoke DB2 precompiler)
 5  BIND/REBIND/FREE       (BIND, REBIND, or FREE plans or packages)
 6  RUN                    (RUN an SQL program)
 7  DB2 COMMANDS           (Issue DB2 commands)
 8  UTILITIES              (Invoke DB2 utilities)
 D  DB2I DEFAULTS          (Set global parameters)
 X  EXIT                   (Leave DB2I)









PRESS:                     END to exit      HELP for more information
```

Figure 220. DB2 Interactive Selection panel

Select option **1**, **SPUFI**, on the DB2 Interactive selection panel, to execute the LDAP migration script. As shown in Figure 222 on page 182, you specify the input data set containing the LDAP migration script LDAPSPMG at option 1. We copied the migration script to a private library, called *graaff.pdg.exec,* where we made

the required changes. Initially the migration script executes a SQL command to determine the level of the DB2 database, as shown in Figure 221.

```
select DB_VERSION from hilding.LDAP_NEXT_EID;
```

Figure 221. LDAPSPMG example

This script is executed under the authority of the user that initally installed LDAP on your system. In our case that was user HILDING.

Next, we execute the LDAP migration script, as shown in Figure 222.

```
                          SPUFI                              SSID: DB51
===>

Enter the input data set name:        (Can be sequential or partitioned)
 1  DATA SET NAME ... ===> 'GRAAFF.PDG.EXEC(LDAPSPMG)'
 2  VOLUME SERIAL ... ===>            (Enter if not cataloged)
 3  DATA SET PASSWORD ===>            (Enter if password protected)

Enter the output data set name:       (Must be a sequential data set)
 4  DATA SET NAME ... ===> OUTPUT

Specify processing options:
 5  CHANGE DEFAULTS   ===> YES        (Y/N - Display SPUFI defaults panel?)
 6  EDIT INPUT ...... ===> YES        (Y/N - Enter SQL statements?)
 7  EXECUTE ......... ===> YES        (Y/N - Execute SQL statements?)
 8  AUTOCOMMIT ...... ===> YES        (Y/N - Commit after successful run?)
 9  BROWSE OUTPUT ... ===> YES        (Y/N - Browse output data set?)

For remote SQL processing:
10  CONNECT LOCATION  ===>


PRESS:  ENTER to process    END to exit             HELP for more information
```

Figure 222. DB2 SPUFI panel

When we press Enter, we are asked to confirm the current SPUFI defaults, as shown in Figure 223 on page 183.

```
                          CURRENT SPUFI DEFAULTS            SSID: DB51
===>

Enter the following to control your SPUFI session:
 1  ISOLATION LEVEL   ===> RR         (RR=Repeatable Read, CS=Cursor Stability)
 2  MAX SELECT LINES  ===> 250        (Maximum number of lines to be
                                        returned from a SELECT)
Output data set characteristics:
 3  RECORD LENGTH ... ===> 4092       (LRECL=Logical record length)
 4  BLOCK SIZE ...... ===> 4096       (Size of one block)
 5  RECORD FORMAT ... ===> VB         (RECFM=F, FB, FBA, V, VB, or VBA)
 6  DEVICE TYPE ..... ===> SYSDA      (Must be DASD unit name)

Output format characteristics:
 7  MAX NUMERIC FIELD ===> 33         (Maximum width for numeric fields)
 8  MAX CHAR FIELD .. ===> 80         (Maximum width for character fields)
 9  COLUMN HEADING .. ===> NAMES      (NAMES, LABELS, ANY or BOTH)




PRESS:  ENTER to process    END to exit              HELP for more information
```

Figure 223. DB2 SPUFI Defaults Panel

To confirm these defaults we press Enter again. We are now in ISPF EDIT mode, as shown in Figure 224 on page 184, where we can change the input if required.

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT       GRAAFF.PDG.EXEC(LDAPSPMG) - 01.00              Columns 00001 00072
000007 --//*************************************************************//
000008
000009 -- Use the following statements to migrate your EXISTING Release 5 or
000010 -- Release 6 LDAP Server DB2 database to the Release 7 schema.
000011 -- After running this script in SPUFI you should be able to run the
000012 -- Release 7 LDAP Server using your newly-migrated database, with no
000013 -- loss of existing data.  The newly-created tablespace name will
000014 -- be used to update the database section of the LDAP Server
000015 -- configuration file.
000016 --
000017 -- First, to determine whether you need to migrate your existing
000018 -- database, uncomment the SQL query following this paragraph
000019 -- and run the query under SPUFI. Change the uuuuuuuu to the
000020 -- name of the user which created your original tables. This
000021 -- will be the name associated with "dbuserid" in your server
000022 -- configuration file.  If the result of the query
000023 -- is '5.0', you must uncomment and run the remainder of this SPUFI
000024 -- script; if the result of the query is '5.1', your database has
000025 -- already been migrated and no further changes are needed to run
000026 -- the Release 7 LDAP Server.
000027 --
000028    select DB_VERSION from hilding.LDAP_NEXT_EID;
000029 --
000030 -- If the result of the preceding query was '5.0', uncomment the
000031 -- remaining statement in this script, change the dddddddd to the
000032 -- name of the database in which your LDAP tables reside.  This
000033 -- will be the name associated with "databasename" in your server
Command ===>                                                Scroll ===> CSR
```

Figure 224. LDAPSPMG in ISPF EDIT mode

We already changed the name of the table to HILDING.LDAP_NEXT_EID . In case you have not done so, you can change this here. Next we press PF3 to save the changes, if any were made. In case you had already changed the query, as we did, you will receive an informational message (DSNE0803A) on the top of the screen, as shown in Figure 225 on page 185. To then execute the query, we press Enter. The results of the query should either be 5.0 or 5.1, where 5.0 means your DB2 databases need to be migrated to the Release 7 or 8 schema and 5.1 or greater means your databases are already migrated to the new schema.

```
                          SPUFI                                SSID: DB51
===>
DSNE803A INPUT FILE WAS NOT CHANGED. PRESS ENTER TO CONTINUE
Enter the input data set name:        (Can be sequential or partitioned)
 1  DATA SET NAME ... ===> 'GRAAFF.PDG.EXEC(LDAPSPMG)'
 2  VOLUME SERIAL ... ===>            (Enter if not cataloged)
 3  DATA SET PASSWORD ===>            (Enter if password protected)

Enter the output data set name:       (Must be a sequential data set)
 4  DATA SET NAME ... ===> OUTPUT

Specify processing options:
 5  CHANGE DEFAULTS   ===> *          (Y/N - Display SPUFI defaults panel?)
 6  EDIT INPUT ...... ===> *          (Y/N - Enter SQL statements?)
 7  EXECUTE ......... ===> YES        (Y/N - Execute SQL statements?)
 8  AUTOCOMMIT ...... ===> YES        (Y/N - Commit after successful run?)
 9  BROWSE OUTPUT ... ===> YES        (Y/N - Browse output data set?)

For remote SQL processing:
10  CONNECT LOCATION  ===>


PRESS:  ENTER to process    END to exit             HELP for more information
```

Figure 225. DB2 SPUFI panel

The result of the query is shown in Figure 226, where you can see the result is 5.0. We need to execute the next step of the LDAPSPMG script to do the actual migration.

```
 Menu  Utilities  Compilers  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 BROWSE    GRAAFF.OUTPUT                          Line 00000010 Col 001 080
--
   select DB_VERSION from hilding.LDAP_NEXT_EID;
---------+---------+---------+---------+---------+---------+---------+---------+
DB_VERSION
---------+---------+---------+---------+---------+---------+---------+---------+
5.0
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+
 Command ===>                                              Scroll ===> CSR
```

Figure 226. Results of the LDAPSPMG script

The actual migration involves creating a new tablespace, as shown in Figure 227.

```
--
-- create tablespace tttttttt in dddddddd locksize tablespace
--             bufferpool BP0;
--
```

Figure 227. LDAPSPMG Create Tablespace contents

Uncomment the remaining statement in the LDAPSPMG script, change the dddddddd to the name of the database in which your LDAP tables reside. This will

be the name associated with databasename in your server configuration file (slapd.conf). Change `tttttttt` to a name of your choosing, which will be assigned to the newly-created tablespace. This name will be associated with the tbspacemutex keyword in the database section of the server configuration file.

Attention

This statement *must* be run by the user ID used to create your database originally. This will be the name associated with the “dbuserid” in your server configuration file (SLAPD.CONF).

We now go back to the DB2 SPUFI panel to actually create the new tablespace, as shown in Figure 228.

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT       GRAAFF.PDG.EXEC(LDAPSPMG) - 01.01              Columns 00001 00072
000043    create tablespace ldapv27t in ldapdb locksize tablespace
000044                bufferpool BP0;
000045 --
000046 --
****** **************************** Bottom of Data ****************************
```

Figure 228. DB2 SPUFI Edit panel

We call our new tablespace LDAPV27T, to reflect that this is a LDAP release 7 tablespace. The database we use is LDAPDB, which is the database we originally used in our Release 5 implementation of LDAP on OS/390.

Figure 229 shows the output of the execution of the LDAPSPMG script.

```
 Menu  Utilities  Compilers  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 BROWSE    GRAAFF.OUTPUT                              Line 00000042 Col 001 080
-- server configuration file (slapd.conf).
--
   create tablespace ldapv27t in ldapdb locksize tablespace
               bufferpool BP0;
---------+---------+---------+---------+---------+---------+---------+---------+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+---------+
---------+---------+---------+---------+---------+---------+---------+---------+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+---------+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
DSNE621I NUMBER OF INPUT RECORDS READ IS 46
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 58
******************************** Bottom of Data ********************************
```

Figure 229. LDAPSPMG output

To verify your definition you can run the SQL query shown in Figure 230.

```
select * from sysibm.systablespace order by name
```

Figure 230. SQL Query to check definition made

The relevant part of the output of this query is shown in Figure 231.

```
 Menu  Utilities  Compilers  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 BROWSE    GRAAFF.OUTPUT                          Line 00000000 Col 001 080
******************************** Top of Data ********************************
---------+---------+---------+---------+---------+---------+---------+---------+
select * from sysibm.systablespace order by name;                    00010019
---------+---------+---------+---------+---------+---------+---------+---------+
NAME     CREATOR   DBNAME      DBID    OBID    PSID  BPOOL     PARTITIONS  LOCK
---------+---------+---------+---------+---------+---------+---------+---------+
DSNRGFTS  DAV       DSNRGFDB     257       1       2  BP0               0  A
DSNRLS01  DAV       DSNRLST      256       1       2  BP0               0  A
DSN32K01  GRAAFF    DSNDB07        7       3       4  BP32K             0  A
DSN4K01   GRAAFF    DSNDB07        7       1       2  BP0               0  A
LDAPTBSP  HILDING   LDAPDB       258       1       2  BP32K             1  A
LDAPV27T  HILDING   LDAPDB       258     182     183  BP0               0  S
LDAP32K   HILDING   LDAPDB       258       5       6  BP32K             0  A
LDAP4K    HILDING   LDAPDB       258       3       4  BP0               0  A
SYSCOPY   SYSIBM    DSNDB06        6       7      16  BP0               0  A
SYSDBASE  SYSIBM    DSNDB06        6       1       9  BP0               0  A
```

Figure 231. Query results

If the definition is correct, you then make the necessary changes to the server configuration file (SLAPD.CONF), as stated in the LDAPSPMG script.

The parameter tablespaceentry needs to reflect the new tablespacename, as shown in the relevant part of our SLAPD.CONF file in Figure 232.

```
#########################################################################
# rdbm database definitions
#########################################################################
database rdbm GLDBRDBM
# The following options must be filled in with appropriate values
# for your DB2 setup, prior to attempting to run with the DB2 backend.
servername CENTDB2
databasename LDAPDB
dbuserid HILDING
#tbspaceentry LDAPTBSP
tbspaceentry LDAPV27T
tbspace32k LDAP32K
tbspace4k LDAP4K
dsnaoini HILDING.ICF.DSNAOINI
```

Figure 232. SLAPD.CONF file containing the tablespaceentry

You can now start your LDAP Server, as shown in Figure 233 on page 188.

```
S LDAPSRV
IRR812I PROFILE LDAPSRV.** (G) IN THE STARTED CLASS WAS USED 163
        TO START LDAPSRV WITH JOBNAME LDAPSRV.
$HASP100 LDAPSRV  ON STCINRDR
IEF695I START LDAPSRV  WITH JOBNAME LDAPSRV  IS ASSIGNED TO USER
LDAPSRV , GROUP LDAPGRP
$HASP373 LDAPSRV  STARTED
IEF403I LDAPSRV - STARTED - TIME=13.41.10
GLD0022I OS/390 Security Server Version 2 Release 7 - LDAP Server 168
 Starting slapd.

GLD2062I The LDAP Server will operate in single-server mode. 169

GLD2065I Current DB schema version is 5.0. 170

GLD0122I Slapd is ready for requests. 171
```

Figure 233. LDAPSRV Release 7 startup messages

Message GLD2065I suggests the migration was unsuccessful, but that is not true.

5.4 Encryption support for password values stored in LDAP

APAR OW41326 introduces storing user passwords in some encrypted form in an OS/390 LDAP Server. This APAR applies to OS/390 Release 7 and 8.

To start encrypting or hashing user passwords, you have to add a statement to the database section of your LDAP configuration, called `pwEncryption`. The `pwEncryption` option supports the following parameters:

none Specifies no encryption.

crypt Specifies that userPassword attribute values are encoded by the UNIX crypt hash algorithm before they are stored in the directory. These passwords are stored in the RDBM (DB2) prefixed with the tag{crypt}.

MD5 Specifies that userPassword attribute values are encoded by the MD5 hash algorithm before they are stored in the directory. These passwords are stored in the RDBM (DB2) prefixed with the tag{MD5}.

SHA Specifies that userPassword attribute values are encoded by the SHA hash algorithm before they are stored in the directory. These passwords are stored in the RDBM (DB2) prefixed with the tag{SHA}.

DES:keylabel Specifies that userPassword attribute values are encoded by the SHA hash algorithm before they are stored in the directory. These passwords are stored in the RDBM (DB2) prefixed with the tag{DES:keylabel}. The *keylabel* must refer to a valid data-encrypting key, also called a data key, generated by the Key Generation Utility Program (KGUP) of ICSF and stored in the Cryptographic Key Data Set (CKDS).

5.4.1 Migrating clear text passwords to hashed or encrypted passwords

The migration of clear text passwords to hashed or encrypted passwords is achieved in one of the following ways:

1. Update the LDAP configuration file (SLAPD.CONF) with `pwEncryption` and a valid value, start the LDAP Server, and then run the DB2PWDEN utility.

   The DB2PWDEN utility reads all the userPassword attribute values and encrypts all clear text userPassword attribute values, and stores them in the RDBM (DB2) backend.

2. Update SLAPD.CONF with `pwEncryption` and a valid value, start the LDAP Server, and then let the passwords be updated as modify operations are processed in order to reset or add userPassword attribute values. This will slowly change clear text userpassword attribute values in the RDBM (DB2) backend to encrypted values.

3. Update SLAPD.CONF with `pwEncryption` and a valid value. Then, run the DB2LDIF utility to unload the RDBM backend and run the utility again to reload the RDBM backend.

The DB2PWDEN utility is provided as part of APAR OW41326 and is located in directory /usr/lpp/ldap/sbin as shown in Figure 234.

```
GRAAFF @ SC57:/usr/lpp/ldap/sbin>ls -all
total 784
drwxr-xr-x   3 STC      OMVSGRP     8192 Dec 22 16:34 .
drwxr-xr-x   9 STC      OMVSGRP     8192 Jun 22  1999 ..
-rwxr-xr-t   2 STC      OMVSGRP       19 Jun 22  1999 GLDSLAPD
drwxr-xr-x   2 STC      OMVSGRP     8192 Dec 22 16:23 IBM
-rwxr-xr-x   2 STC      OMVSGRP   110592 Dec 22 16:34 db2ldif
-rwxr-xr-x   2 STC      OMVSGRP    86016 Dec 22 16:34 db2pwden
-rwxr-xr-x   2 STC      OMVSGRP   172032 Dec 22 16:34 ldif2db
-rwxr-xr-x   2 STC      OMVSGRP      765 Jan  5 13:08 slapd
GRAAFF @ SC57:/usr/lpp/ldap/sbin>
 ===>
```

Figure 234. Listing of /usr/lpp/ldap/sbin

The DB2PWDEN utility allows you to selectively encrypt userPassword attribute values. You can either encrypt the whole RDBM (DB2) backend or do a selective migration of a certain Directory Information Tree (DIT). The parameters supported by DB2PWDEN are:

-? Print help text.

-h host Host address.

-p port Port on LDAP server.

-d level Set LDAP debugging level to 'level'.

-D binddn Bind dn.

-w passwd Bind passwd (for simple authentication).

-b basedn Base dn for search. LDAP_BASEDN in environment is the default.

-Z Use a secure ldap connection for search.

-K keyfile File to use for keys/certificates.

-P key_pw Keyfile password.

-N key_dn Certificate Name in keyfile.

5.4.2 LDAP and OCSF

LDAP uses the Open Cryptographic Services Facility (OCSF) APIs to provide the hashing and encryption functions, as discussed previously. The LDAP Server requires RACF authorizations to use those OCSF functions. The started task user ID of the LDAP Server requires the following RACF authorizations in the RACF FACILITY class:

CDS.CSSM Authorizes the LDAP daemon to call OCSF services

CDS.CSSM.CRYPTO Authorizes the LDAP daemon to call a Cryptographic Service Provider (CSP)

Issue the following RACF commands to authorize the LDAP Server to use the OCSF services required:

```
RDEF FACILITY CDS.CSSM UACC(NONE)
PE CDS.CSSM CLASS(FACILITY) ID(ldap-server-User-ID) ACC(READ)
RDEF FACILITY CDS.CSSM.CRYPTO UACC(NONE)
PE CDS.CSSM.CRYPTO CLASS(FACILITY) ID(ldap-server-User-ID) ACC(READ)
```

Note: The RACF FACILITY is probably raclisted, so you have to issue a SETROPTS RACLIST(FACILITY) REFRESH.

OCSF uses the IBM CCA cryptographic module, as supplied with OCSF as one of the CSPs, to perform the cryptographic operations. The IBM CCA Cryptographic module relies on the Integrated Cryprographic Services Facility (ICSF) and its underlying cryptographic hardware to provide its services. Therefore, the started task user ID of the LDAP Server must be authorized to the following ICSF services in RACF to be able to perform the cryptographic operations:

CSFOWH Gets control during the one-way hash generate callable service.

CSFKGN Gets control during the key generate callable service.

CSFENC Gets control during the encipher callable service.

CSFDEC Gets control during the decode callable service.

Issue the following RACF commands to authorize the LDAP started task for these ICSF callable services:

```
permit CSFOWH class(csfserv) id(ldapsrv) access(r)
permit CSFKGN class(csfserv) id(ldapsrv) access(r)
permit CSFENC class(csfserv) id(ldapsrv) access(r)
permit CSFDEC class(csfserv) id(ldapsrv) access(r)
```

Note: *LDAPSRV* in our example is our started task user ID

If the LDAP Server uses an SDBM backend to access RACF, the OCSF libraries also need to be APF-authorized. The APF-authorized extended attribute must be turned on for the OCSF DLLs. The DLLs (dll and .so files) in the /usr/lpp/ocsf/lib and /usr/lpp/ocsf/addins directories must have their APF-authorized extended attribute turned on by using the `extattr +a` command. Use the following commands:

```
RDEFINE FACILITY BPX.FILEATTR.APF UACC(NONE)

PERMIT BPX.FILEATTR.APF CLASS(FACILITY)ID(userid) ACCESS(UPDATE)

SETROPTS RACLIST(FACILITY)REFRESH
```

where `userid` is the ID from which the `extattr` command will be run. Use the following commands from an OMVS command prompt:

```
cd /usr/lpp/ocsf/lib
extattr +a .dll
cd /usr/lpp/ocsf/addins
extattr +a .so
```

Refer to *OS/390 UNIX System Services Planning*, SC28-1890, for more details. If program control is active, the OCSF DLLs must also be program controlled. See the configuration information in the *Open Cryptographic Services Facility Application Developer's Guide and Reference*, SC24-5875 for more details.

Before we restart our LDAP Server, enabling the password encryption capabilities by using the OCSF support, we first have to update the SLAPD.CONF file to indicate what kind of encryption we want to use. Figure 235 shows our SLAPD.CONF file indicating DES encryption and a label of the DES key we use for encryption and decryption.

Note: The DES key generation is discussed in 5.5.5, "Enabling LDAP support" on page 197.

```
...... other parameters

database          rdbm GLDBRDBM
servername        DB2U
databasename      LDAPPG
dbuserid          GRAAFF
tbspaceentry      PGTBSENT
tbspace32k        PGTBS32K
tbspace4k         PGTBS4K1
tbspacemutex      PGTBS4K2
pwEncryption      DES:LDAP2
...... other parameters
```

Figure 235. SLAPD.CONF file with pwEncryption enabled

After updating the SLAPD.CONF file, we are now able to start the LDAP Server again. To really see if OCSF is functioning correctly, you need to turn on the DEBUG option. In our example we used 65535 as our DEBUG level. The JOBLOG of our LDAP Server that indicates the use of OCSF is shown in Figure 236 on page 192.

```
==> ocsf_setup, *hCSP=0x00000000, algorithm=5
ocsf_init entered
ocsf_init exited, ocsf_rc=0
ocsf_attach entered, *hCSP = 0x00000000, algorithm=5
ocsf_attach: CSSM_ListModules succeeded, num of mod=3
ocsf_attach: CSSM_GetModuleInfo succeeded, num of services=1
ocsf_attach: CSSM_GetModuleInfo succeeded, num of subServices=1
  pCspInfo->CspType=2
  CSSM_CSP_SOFTWARE=1
  CSSM_CSP_HARDWARE=2
ocsf_attach: num of Hardware Capabilities=20
ocsf_attach: DES found
ocsf_attach: Needed algorithm found in hardware
ocsf_attach: Value of NAME   ITEMS(CspIndex) = ibmcca.so
ocsf_attach: Value of DATA 1 ITEMS(CspIndex) = 0x474d0880
ocsf_attach: Value of DATA 2 ITEMS(CspIndex) = 0x0000b44c
ocsf_attach: Value of DATA 3 ITEMS(CspIndex) = 0x000011d1
ocsf_attach: Value of DATA 4 9  ITEMS(CspIndex) = 0x000000cf
ocsf_attach: Value of *hCSP = 0xc9daca9e
ocsf_attach: CSSM_ModuleAttach succeeded
ocsf_attach exited
<== ocsf_setup, *hCSP=0xc9daca9e, rc=0
ocsf_getDESKey entered
ocsf_getDESKey exited
```

Figure 236. LDAPSRV JOBLOG showing the usage of OCSF

When either the *APF* or *program control* extended attributes are not set when required, the JOBLOG of the LDAP Server indicates the errors, as shown in Figure 237. The `cssm32.dll not available` message might be misleading, but it really indicates that either program control or APF is not set for the OCSF modules (DLL) in the /usr/lpp/ocsf/lib directory.

```
==> ocsf_setup, *hCSP=0x00000000, algorithm=5
ocsf_setup, cssm32.dll not available.
<== ocsf_setup, *hCSP=0x00000000, rc=1
```

Figure 237. Error message indicating ocsf setup failed

One of the first exploiters of encrypted passwords stored in an OS/390 LDAP Server is Host On-Demand, which we discuss in the next section.

5.5 Exploitation of the OS/390 LDAP Server by Host On-Demand

Support for an OS/390 LDAP Server to store Host On-Demand (HOD) users, groups and configuration is available with Host On-Demand Version 4 CSD 2.

To configure Host On-Demand to use the OS/390 LDAP Server, complete the following steps:

1. Configure the OS/390 LDAP Server, as described in the *OS/390 Security Server LDAP Server Administration and Usage Guide*, SC24-5861.
2. Check your DB2 environment for enough `ldap32k` tablespace size for use by the OS/390 LDAP Server.

3. Build a Directory Information Tree, which will contain the HOD directory information.
4. Check your TCP/IP environment.
5. Perform the actual migration.

5.5.1 Configure the OS/390 LDAP Server

Our installation already had an OS/390 LDAP Server up and running.

The Host On-Demand utilization of LDAP requires a parent distinguished name to be added to the LDAP installation or the use of an existing one. This is achieved by adding a so-called *suffix* to SLAPD.CONF. We used an existing suffix called `o=IBM,c=US`.

An example of our suffix in our LDAP configuration file is shown in Figure 238.

```
suffix "o=IBM,c=US"
```

Figure 238. Suffix example

The HoD documentation states to add the following LDAP schemas to the LDAP configuration:

- V2.1.IBM.at (attributes)
- V2.1.IBM.oc (objectclasses)
- ods.delta.oc (additional objectclasses)

Attention

At the time of this writing, the schema file additions used in our example were not shipped with the HOD installation; instead, they were sent internally by HOD development. The HOD documentation states it should work with the old (slapd) schemas as well. Our tests showed it did not.

Add the lines shown in Figure 239 to SLAPD.CONF in the database section.

```
index principalPtr eq
index dc eq
index o eq
index name eq
index objectClass eq
index uid eq
```

Figure 239. HOD additions for SLAPD.CONF

Another change to make to SLAPD.CONF is to increase the sizelimit to 5000 (the default is 500). This can be added as a general statement, applying it to all backends; or it can be added after a database statement and it will apply just to that backend. Increasing the sizelimit will return more entries from an LDAP search operation. Figure 240 shows an excerpt of the SLAPD.CONF file we used; bold text indicates the changes we made.

```
port              389
maxthreads        350
maxconnections    100
waitingthreads    10
timelimit         3600
sizelimit         5000
......
......
database          rdbm GLDBRDBM
servername        DB2U
databasename      LDAPPG
dbuserid          GRAAFF
tbspaceentry      PGTBSENT
tbspace32k        PGTBS32K
tbspace4k         PGTBS4K1
tbspacemutex      PGTBS4K2
suffix            "cn=localhost"
suffix            "o=IBM,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
index  principalPtr eq
index  dc eq
index  o eq
index  name eq
index  objectClass eq
index  uid eq
```

Figure 240. slapd.conf except

5.5.2 Checking your LDAP - DB2 environment

The new LDAP schema files (schema.*.at and schema.*.oc) create about 600 tables. It is important to check your current usage of these schema files. See 5.2.5, “Using the schema files” on page 171 for more information.

5.5.3 Build a Directory Information Tree and define the AdminDN

Consult with the administrator of your OS/390 LDAP directory server before proceeding. Your administrator must define the directory structure that Host On-Demand will use. In our example, we added a DIT called `cn=HOD,ou=ITSO, o=IBM,c=US`, under our existing suffix `o=IBM,c=US`. We created a data set called graaff.ldaphod.add that contains the DIT definitions, as shown in Figure 241.

```
dn: cn=HOD,ou=ITSO,o=IBM,c=US
objectclass: person
cn: HOD
sn: HOD
description: HOD Chain for HOD V4.0 LDAP DIT
```

Figure 241. DIT for the HOD usage of LDAP

Figure 242 shows the JCL used to define the DIT to the LDAP Server.

```
//LDIF2DB PROC REGSIZE=2048K,
//*-------------------------------------------------------------------
//* CUSTOMIZABLE SYMBOLIC PARAMETERS
//* Customize the parameters here for desired behavior.
//*-------------------------------------------------------------------
//  PARMS='',
//  GLDHLQ='GLD',
//  OUTCLASS='A'
//LDIF2DB  EXEC PGM=GLDLD2DB,REGION=&REGSIZE,
//         PARM=('/&PARMS')
//*-------------------------------------------------------------------
//* STEPLIB must be customized based on install HLQ.
//*-------------------------------------------------------------------
//STEPLIB   DD DSN=&GLDHLQ..SGLDLNK,DISP=SHR
//         DD DSN=DSN610.SDSNLOAD,DISP=SHR
//DSNAOINI DD DSN=JJONES.ICF.DSNAOINI,DISP=SHR
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(PDGCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//SYSIN    DD DSN=GRAAFF.LDAPHOD.ADD,DISP=SHR
//SYSPRINT DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSERR   DD SYSOUT=&OUTCLASS
//STDOUT   DD SYSOUT=&OUTCLASS
//         PEND
//GO       EXEC LDIF2DB
```

Figure 242. LDIF2DB JCL example

The OS/390 LDAP Server does not require the LDAP administrator DN to be defined within the LDAP Server. A definition in the SLAPD.CONF file is sufficient, as shown in Figure 243.

```
.... other parameters

adminDN           "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US"
adminPW           paddle

.... other parameters
```

Figure 243. SLAPD.CONF parameters indicating the LDAP administrator DN

When using LDAP for HOD, the adminDN needs to be defined within the LDAP Server, otherwise the migration will fail. To define the adminDN, you can use the same procedure as done for the definition of the HOD DIT. Figure 244 on page 196 shows the definitions to first add the `organizationalUnit` ITSO and then the actual `(Admin)DN` of the LDAP Server according to the SLADP.CONF file , `cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US`.

```
dn: ou=ITSO, o=IBM, c=US
ou: ITSO
objectclass: organizationalUnit

dn: cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US
objectclass: person
cn: LDAPSRV Admin
sn: LDAPSRV Admin
description: Administrator DN for ou=ITSO,O=IBM,c=US
userPassword: paddle
```

Figure 244. LDIF format for adding the AdminDN and the OU ITSO

5.5.4 Check your TCP/IP environment

The TCP/IP environment requires a `localhost` definition in the /etc/hosts file. This definition is quite normal in a UNIX or Windows environment, but not so usual in an OS/390 environment. Figure 245 shows our /etc/hosts file that contains the loop back address `127.0.0.1` pointing to `localhost`.

```
9.12.14.247  wtsc57 wtsc57.itso.ibm.com
9.3.1.83.rh-lab
9.3.1.212.rh2420b
9.3.1.160.rh2420c
9.3.1.190.rh2420a
127.0.0.1.localhost
```

Figure 245. /etc/hosts file example

If this definition is not available, the migration to the LDAP Server will fail. If you have the Java console function enabled on your browser, the migration failure will show something similar to Figure 246. The text shows `UnknownHostException` for `localhost`.

```
IBM SecureWay Host On-Demand
Version:  4.0.2-B19991109
Copyright IBM Corporation 1997, 1999. All rights reserved.


Trace: remote.Common.invokeMethod(createUser): Exception from server:
com.ibm.managed.security.AuthException: unknown error in login :
java.net.UnknownHostException: localhost
com.ibm.managed.security.AuthException: unknown error in login :
java.net.UnknownHostException: localhost
       at com.ibm.managed.security.client.LoginContext.login(LoginCon-
text.java:798)
       at com.ibm.managed.security.client.LoginContext.login(LoginCon-
text.java:673)
mon.serverInvokeMethod1(Common.java:440)
```

Figure 246. LOCALHOST UnknownHostException error

5.5.5 Enabling LDAP support

When you are ready to enable the LDAP directory support, log on to Host On-Demand as the administrator and click the **Directory** tab (see Figure 247). It is here that you have fields that identify the directory server and suffix that you want to use, and optionally the ability to migrate your existing Host On-Demand configuration data to the LDAP Server. Regardless of whether or not you migrate, the default administrator ID will be created in the directory server with the arbitrary user ID/password of admin/password.

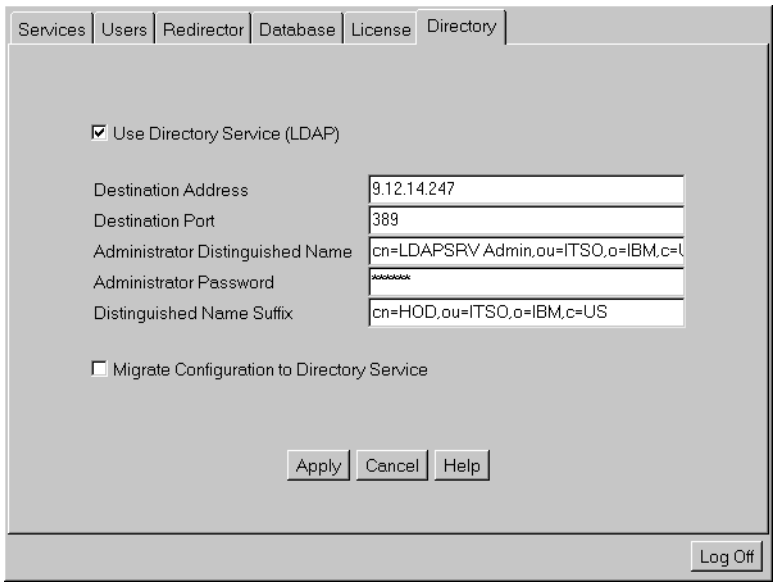


Figure 247. HOD - administrator directory tab

The fields on this panel have the following uses and meanings:

- Use Directory Service (LDAP)

  Placing a check in this box enables the remaining fields on this panel and allows you to specify the directory service that you want to use.

- Destination Address

  Enter the address of the LDAP directory. Use either the host name or the address in dotted decimal format. The default is the address of this server.

- Destination Port

  Enter the TCP/IP port on which the LDAP Server will accept a connection from an LDAP client. The default port is 389.

- Administrator Distinguished Name

  Enter the DN of the directory administrator that allows Host On-Demand to update information. In our example that is `cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US` as shown in Figure 247.

- Administrator password

  Enter the directory administrator's password.

- Distinguished Name Suffix

  Enter the DN of the highest entry in the DIT for which information will be saved. Host On-Demand will store all of its configuration information below

this suffix in the DIT. The DIT was defined previously in 5.5.3, “Build a Directory Information Tree and define the AdminDN” on page 194; it needs to be entered here, `cn=HOD,ou=ITSO,o=IBM,c=US`.

When Apply is clicked, as shown in Figure 248, the Service Manager contacts the LDAP Server, passing the LDAP DN suffix (`cn=HOD,ou=ITSO,o=IBM,c=US`) that will be used by Host On-Demand and the administrator DN (`cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US`) and password (newpass) of the administrator. The password from the Directory tab is used; this is different from the password used to log in to HODAdmin.html.

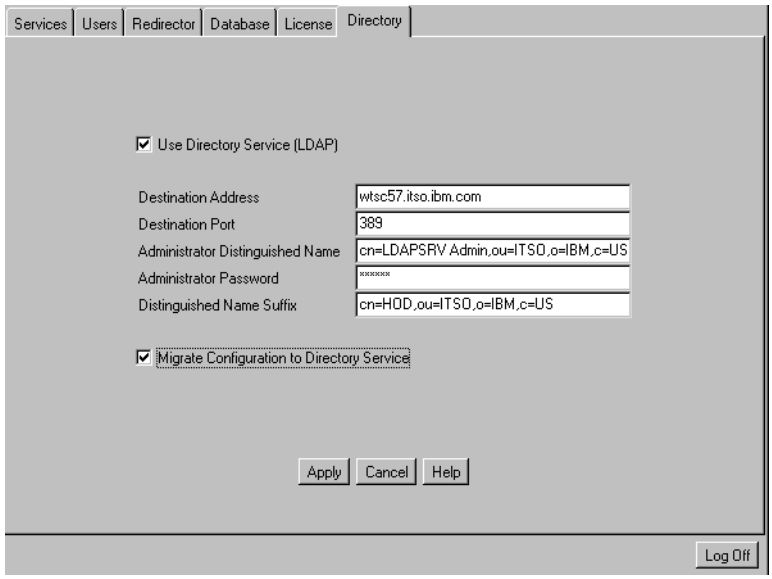


Figure 248. HOD LDAP migration apply process initiation

Next, the LDAP Server authenticates the request as follows:

- Are the DN and password valid as the LDAP administrator?
- Is the suffix defined and initialized?

If the DN and suffix are validated, the Service Manager can proceed.

The Service Manager attempts to create the default Host On-Demand administrator user ID and password (admin/password). If this is the first time this directory has been contacted (or if the admin ID does not exist), this update will succeed. If it is not the first time, and if the admin ID already exists, the request will fail to create the user ID since it is already present (the existing password need not be password); however, the Service Manager ignores the failure and proceeds.

Next, the administration applet is prompted to re-authenticate itself to the LDAP Server and responds by sending admin/newpass (if the administrator had changed its password). In our example, the authentication fails (the directory service is expecting admin/password), which results in the panel shown in Figure 249 on page 199. You should enter the user ID and password of an administrative user that are valid for the new LDAP settings (in our example, admin/password).

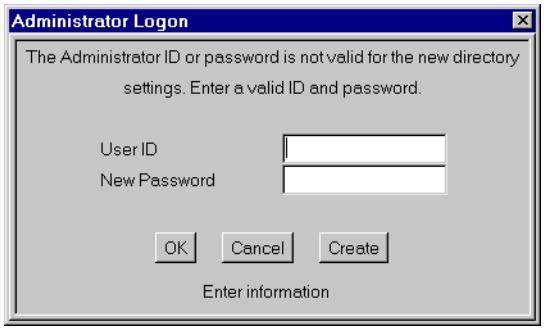


Figure 249. Admin re-signon prompt

Note: Remember that multiple Host On-Demand systems can use the same LDAP Server and suffix to allow for workload balancing, so another system may have reset this password.

After the appropriate user ID/password combination is entered (in our example it is admin/password), the user is authenticated and the Service Manager finishes creating the required information in the LDAP directory. A warning message is displayed, as shown in Figure 250, to warn you that there is no way back to the private data store.

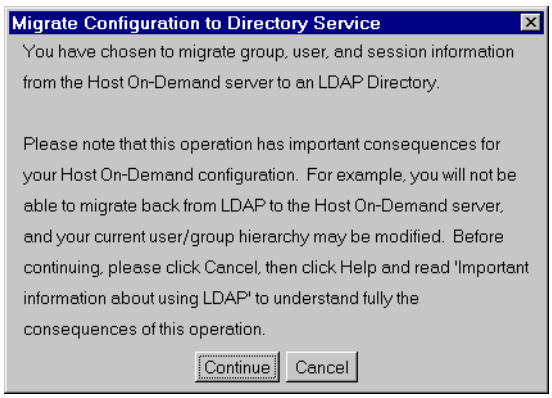


Figure 250. HOD - LDAP migration warning message

Click **Continue**; a migration progress bar is displayed, indicating the progress of the actual migration, as shown in Figure 251.

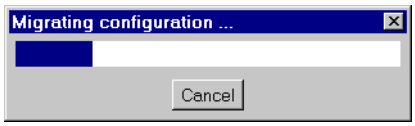


Figure 251. HOD - LDAP migration status bar

If you enable the Java console function on your browser, you can see a more detailed progress report, indicating the users that have been migrated, as shown in Figure 252 on page 200.

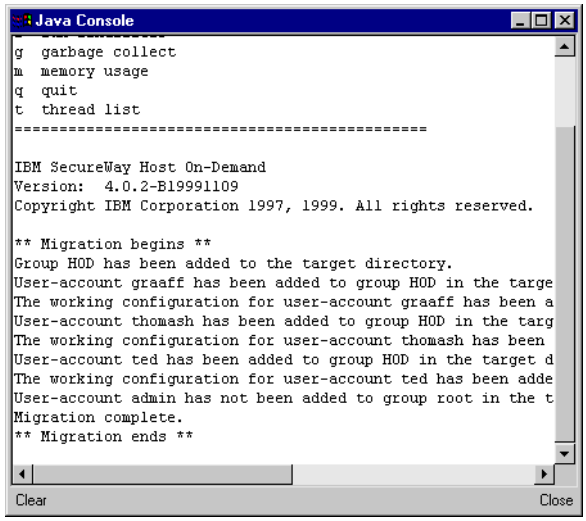


Figure 252. HOD - LDAP migration progress using the Java Console function

An audit trail of all updates is maintained in the directory /usr/lpp/HOD/hostondemand/private/hodldap.log. The contents of our hodldap.log file are shown in Figure 253.

```
** Migration begins **
Group HOD has been added to the target directory.
User-account graaff has been added to group HOD in the target directory.
The working configuration for user-account graaff has been added.
User-account thomash has been added to group HOD in the target directory.
The working configuration for user-account thomash has been added.
User-account ted has been added to group HOD in the target directory.
The working configuration for user-account ted has been added.
User-account admin has not been added to group root in the target directory beca
Migration complete.
** Migration ends **
```

Figure 253. hodldap.log content

After we performed the migration of our HOD users, the passwords of our users were stored in clear text. At the time of this writing, the APAR discussed in 5.4.1, “Migrating clear text passwords to hashed or encrypted passwords” on page 189 has just been released and is not yet installed on our system. When we performed an ldapsearch against the OS/390 LDAP Server, as shown in Figure 254, you can actually see them in clear text, as shown in Figure 255 on page 201.

```
ldapsearch -h 9.12.14.247 -D "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US" -w paddle
-b "ou=ITSO,o=IBM,c=US" "objectclass=*" cn userpassword
```

Figure 254. LDAPSEARCH command to retrieve user passwords

```
cn=admin,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
userpassword=password
cn=admin

principalname=OnDemandDefault,cn=admin,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US

cid=HOD,cn=user groups,cn=HOD,ou=ITSO,o=IBM,c=US

cn=graaff,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
userpassword=racf99p
cn=graaff

principalname=OnDemandDefault,cn=graaff,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US

cn=thomash,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
userpassword=mt1fort
cn=thomash

principalname=OnDemandDefault,cn=thomash,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US

cn=ted,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
userpassword=telnet
cn=ted

principalname=OnDemandDefault,cn=ted,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
```

Figure 255. LDAPSEARCH output showing the clear text passwords

Before we can migrate the `userPassword` values to encrypted or hashed format, we have to indicate in our SLAPD.CONF file that we want to use encryption or hashing using the `pwEncryption` parameter, as shown in Figure 256.

```
.....
database          rdbm GLDBRDBM
servername        DB2U
databasename      LDAPPG
dbuserid          GRAAFF
tbspaceentry      PGTBSENT
tbspace32k        PGTBS32K
tbspace4k         PGTBS4K1
tbspacemutex      PGTBS4K2
pwEncryption      DES:LDAP2
.....
```

Figure 256. SLAPD.CONF example using pwEncryption

Our example shows we are using DES encryption. This requires that a label is specified for the DES key (ldap2) to be used. DES keys are generated using the Key Generator Utility Program (KGUP) and are stored in the Cryptographic Key Data Set (CKDS) of ICSF.

Figure 257 on page 202 shows the JCL and parameters we used to generate the DES key to encrypt our userpassword values in the LDAP directory.

```
//CSFKGUP JOB (999,POK),'Paul de Graaff',CLASS=A,REGION=4M,
//     MSGCLASS=T,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//KGUPPROC EXEC PGM=CSFKGUP,PARM=('SSM')
//CSFCKDS  DD   DSN=ICSF.V2R1M0.SCSFCKDS,DISP=SHR
//CSFIN    DD   *
  ADD LABEL(LDAP2) TYPE(DATA) LENGTH(8) DES
/*
//CSFDIAG  DD   SYSOUT=*
//CSFKEYS  DD   DSN=ICSF.SCSFKEYS,DISP=SHR
//CSFSTMNT DD   DSN=ICSF.SCSFSTMT,DISP=SHR
```

Figure 257. KGUP sample JCL and parameters

The output from the KGUP utility is shown in Figure 258.

```
KEY GENERATION DIAGNOSTIC REPORT
>>>CSFG0402 INSTALLATION EXIT NOT LOADED.
  ADD LABEL(LDAP2) TYPE(DATA) LENGTH(8) DES
>>>CSFG0321 STATEMENT SUCCESSFULLY PROCESSED.
>>>CSFG0002 CRYPTOGRAPHIC KEY GENERATION - END OF JOB.  RETURN CODE = 0.
```

Figure 258. KGUP results

After the DES key has been generated and the SLAPD.CONF file has been updated with the DES key information, we can re-start the LDAP Server. To see if your start-up was successful, you will have to enable debugging, as discussed previously in 5.4.2, “LDAP and OCSF” on page 190.

For us, the next step was to use the DB2PWDEN utility to migrate the passwords to an encrypted form. In our example, the migration of userPassword values are those of just the DIT we used for Host On-Demand: `ou=ITSO,o=IBM,c=US`. Figure 259 shows our migration of the `userPassword` values from clear text to using a DES key to encrypt the `userPassword` value.

```
GRAAFF @ SC57:/u/graaff>db2pwden -D "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US" -w paddle
-b "ou=ITSO,o=IBM,c=US"
GLD2086I Encrypt all passwords that are presently in CLEAR format (yes/no)? yes
GRAAFF @ SC57:/u/graaff>
```

Figure 259. DB2PWDEN execution example

DB2PWDEN does not return a message that indicates whether or not the migration was successful, so we have to execute an LDAPSEARCH command, as shown in Figure 254, to see if the `userPassword` attribute values have been encrypted. The result of the LDAPSEARCH command, indicating that the passwords are not encrypted, is shown in Figure 260 on page 203.

```
GRAAFF @ SC57:/u/graaff>ldapsearch -D "cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US" -w padd
-b "ou=ITSO,o=IBM,c=US" "cn=ted" cn userpassword
cn=ted,cn=users,cn=HOD,ou=ITSO,o=IBM,c=US
cn=ted
userpassword=telnet
GRAAFF @ SC57:/u/graaff>
```

Figure 260. LDAPSEARCH command to view userpassword attribute value

This is *not* an error, rather the LDAPSEARCH command initiates a decryption of the `userPassword` value.

Note: Remember we used the administrator for authentication here, to retrieve the `userPassword` value.

Attention

We also performed tests using MD5 and SHA1 as hashing mechanisms for the userpassword values, but were unsuccessful in testing this with Host On-Demand. At the time of this writing we were unable to find out whether this was related to the way the password is stored in the OS/390 LDAP Server, or whether Host On-Demand expects the password to be returned in clear text. Our tests indicated the first rather than the second reason.

For more information on Host On-Demand and LDAP usage in general, see *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01.

5.6 Directory management tools

The directory management tools available with the OS/390 are command line driven, like the LDAPADD, LDAPMODIFY and so on. There are, however, also tools available that provide a Graphical User Interface (GUI) to manage an LDAP directory. In this section we will show examples of two of the available tools, for demonstration purposes only. These directory management tools are :

- IBM's SecureWay Directory Management Tool

  This tool is part of the SDK for the SecureWay Directory Server and can be downloaded from: `http://www.ibm.com/software/network/directory/downloads`

- The LDAP Browser/Editor by Jarek Gawor

  This tool is available for download from:
  `http://www-unix.mcs.anl.gov/~gawor/ldap`

  It is also commercially available through Argonne National Laboratory.

5.6.1 SecureWay Directory Management Tool

IBM's SecureWay Directory Management Tool (DMT) provides a Java-based graphical user interface that enables you to manage information stored in LDAP directory servers. You can use this tool to:

- Connect to one or more LDAP directory servers via SSL or non-SSL connections
- Display server properties and rebind to the server
- List, add, edit and delete schema attributes and objectclasses
- List, add, edit and delete directory entries
- List, add, edit and delete Access Control Lists (ACLs)
- Search the directory tree

The examples we show here are from an installation of the SDK in a WIndows NT environment.

5.6.1.1 Starting the SecureWay DMT

During the installation of the SecureWay SDK, a menu item was added to the Start>Programs>IBM SecureWay directory called Directory Management Tool. This can be used to start the DMT.

By default when starting the DMT, it will look for a directory server on the system on which it is started. The configuration file for the DMT contains a default entry, shown in Figure 261, for the localhost.

```
#browser=
server1.url=ldap://localhost:389
#server1.security.bindDN=
#server1.security.password=
#server1.security.ssl.keyclass=
#server1.security.ssl.keyclass.password=
```

Figure 261. DMT.CONF defaults

To connect to the OS/390 LDAP Server, you have two options:

- Change the DMT.CONF file to reflect your OS/390 LDAP Server.
- Point the DMT to your OS/390 LDAP Server after startup of the DMT.

To change DMT.CONF, open the file with your favorite text editor, and add your LDAP Server information to it, as shown in Figure 262.

```
#browser=
server1.url=ldap://9.12.14.247:389
#server1.url=ldap://localhost:389
server1.security.bindDN=cn=LDAPSRV Admin,ou=ITSO,o=IBM,c=US
server1.security.password=paddle
#server1.security.ssl.keyclass=
#server1.security.ssl.keyclass.password=
```

Figure 262. DMT.CONF file with OS/390 LDAP Server information

Note: The information supplied is for the administrator for the DB2 backend.

When you start the DMT, you might receive the error message shown Figure 263. This message can be ignored; click **OK** to continue.

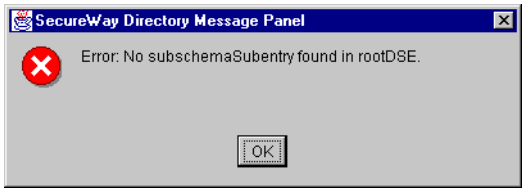


Figure 263. DMT error message

Next you might receive a message about entry sysplex=local, if you have the RACF backend configured, as shown in Figure 264.

Figure 264. DMT warning message

Note: Remember that we used the administrator for the DB2 backend.

Next you will see the main DMT screen, which enables you to manage your OS/390 LDAP Server, as shown in Figure 265.

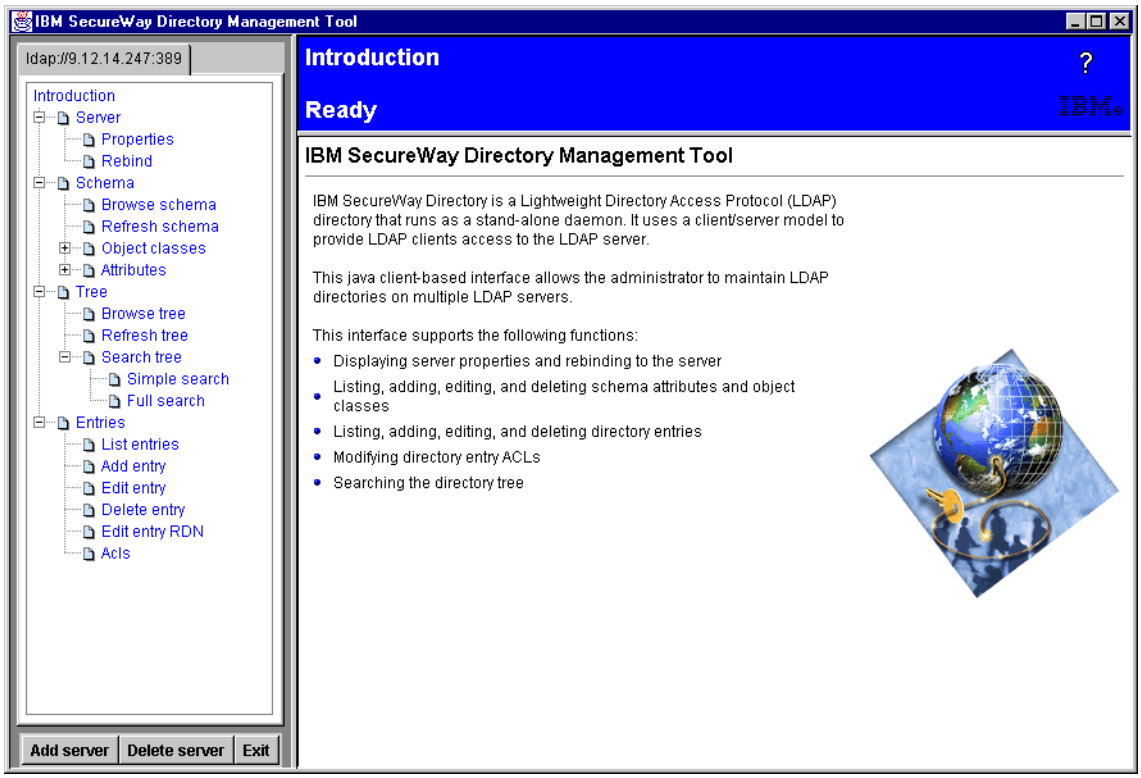


Figure 265. DMT main menu

Next you have the ability to take a look at your OS/390 LDAP Server. For example, you can show the properties of the OS/390 LDAP Server by selecting the option **Properties** under the Server tab, as shown in Figure 266.

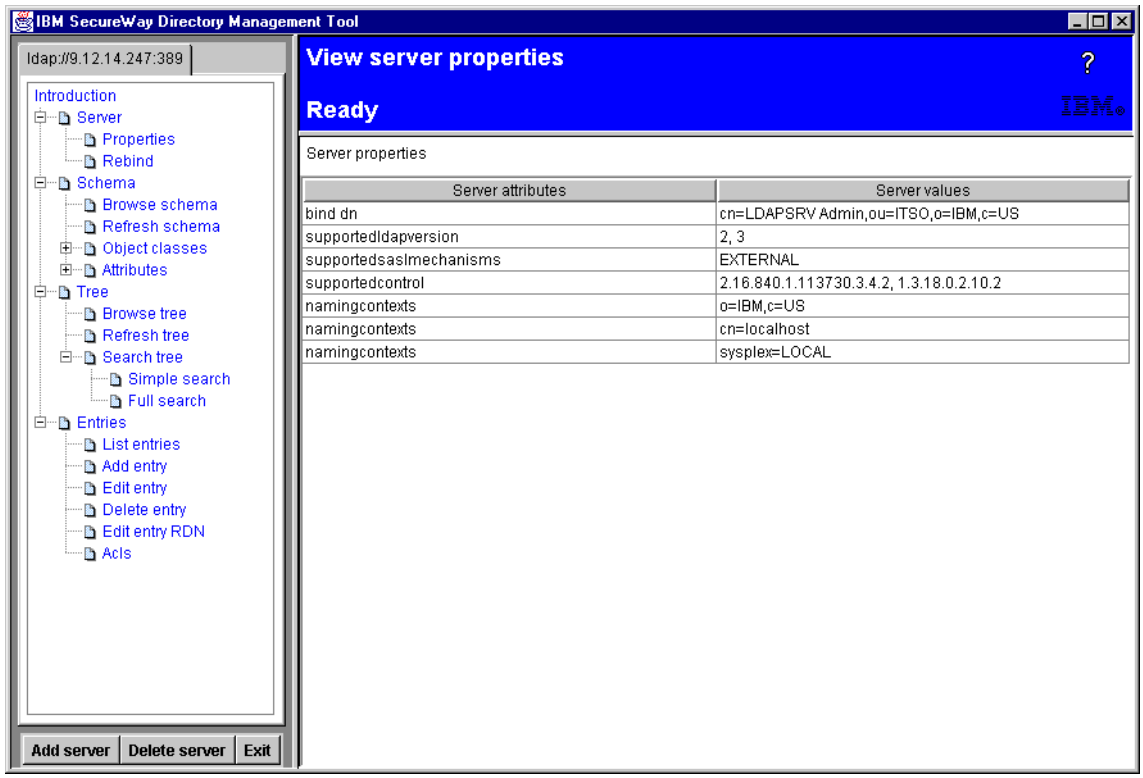


Figure 266. DMT panel showing the OS/390 LDAP Server properties

Another interesting option is to actually show the Directory Information Tree, using the Tree option, and selecting Browse tree. The result in shown in Figure 267 on page 207.

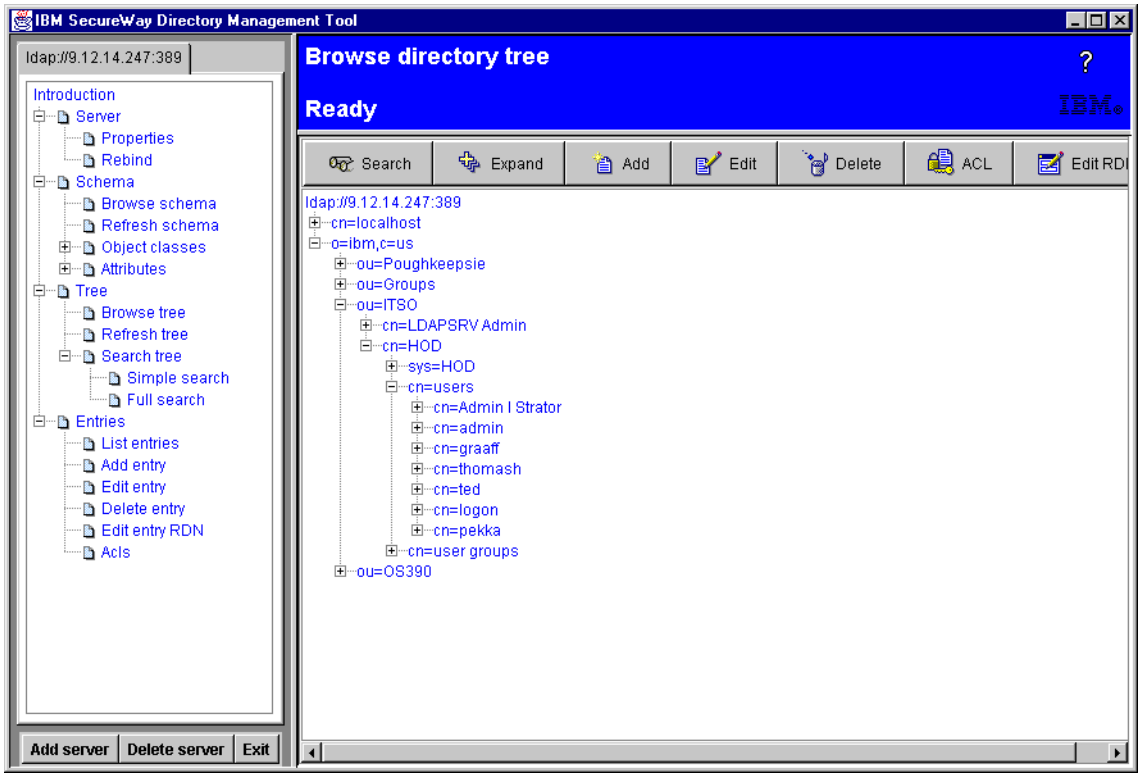


Figure 267. DMT panel showing the Directory Tree

We opened the tree that we defined for Host On-Demand. You can see the user that we migrated and added since.

If you do not like specifying the LDAP Server and administrators distinguished name and password in the DMT.CONF file, you can just leave the default entry in there and specify the required information for your LDAP Server after the DMT is started. When you start the DMT with the default entry, you receive an extra error message, shown in Figure 268.

Figure 268. DMT error message

After you have ignored the other error and warning message, you get the main menu as shown previously in Figure 265 on page 205. You are now able to add your OS/390 LDAP Server by selecting the option **Add server**, in the lower left corner. You then get the window shown in Figure 269 on page 208, to add the parameters relevant to your OS/390 LDAP Server.

DMT

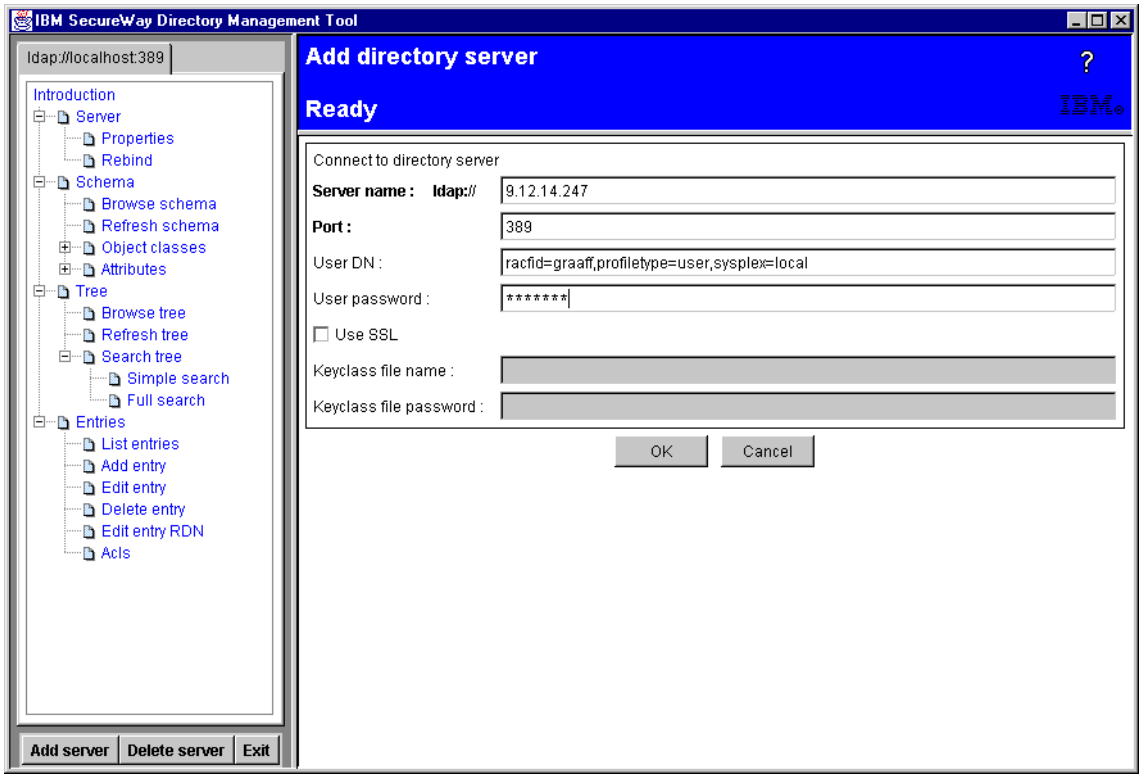


Figure 269. DMT window to add LDAP Server definitions

In this example we specify a RACF user ID (graaff) as the administrator and the password of the RACF user ID, to be able to see the information in the RACF backend using the LDAP interface.

You receive again some of the error and warning messages received earlier in the other examples. Note that you do not receive the one about the `sysplex=local` entry because we are now connecting to the RACF backend.

After clicking **OK**, we see the same main panel that we saw previously. We are now able to browse the tree again, and actually see the RACF information, as shown in Figure 270 on page 209. The DIT `sysplex=local` has two leaves, called profiletype=user and profiletype=group. The profiletype=user leaf, when expanded, shows all the users defined in the RACF database, as shown in Figure 271 on page 209. The profiletype=group leaf shows all the RACF groups defined, as shown in Figure 272 on page 210.

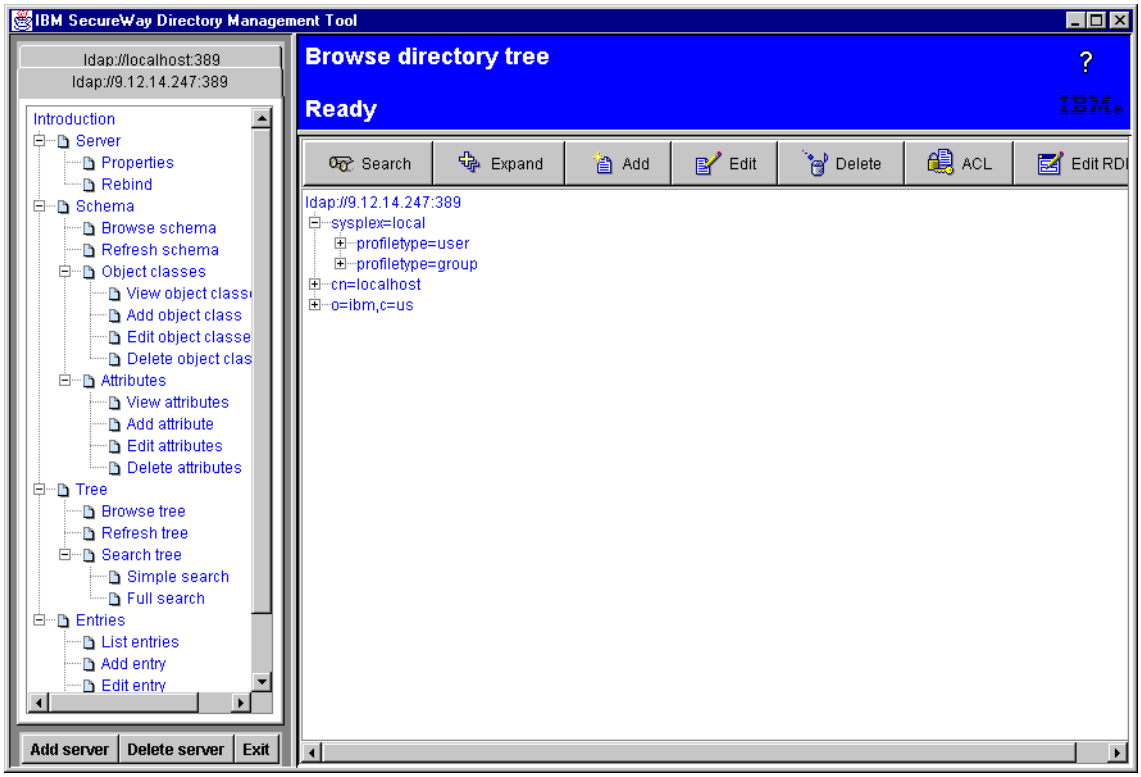


Figure 270. DMT windows showing the RACF leaves

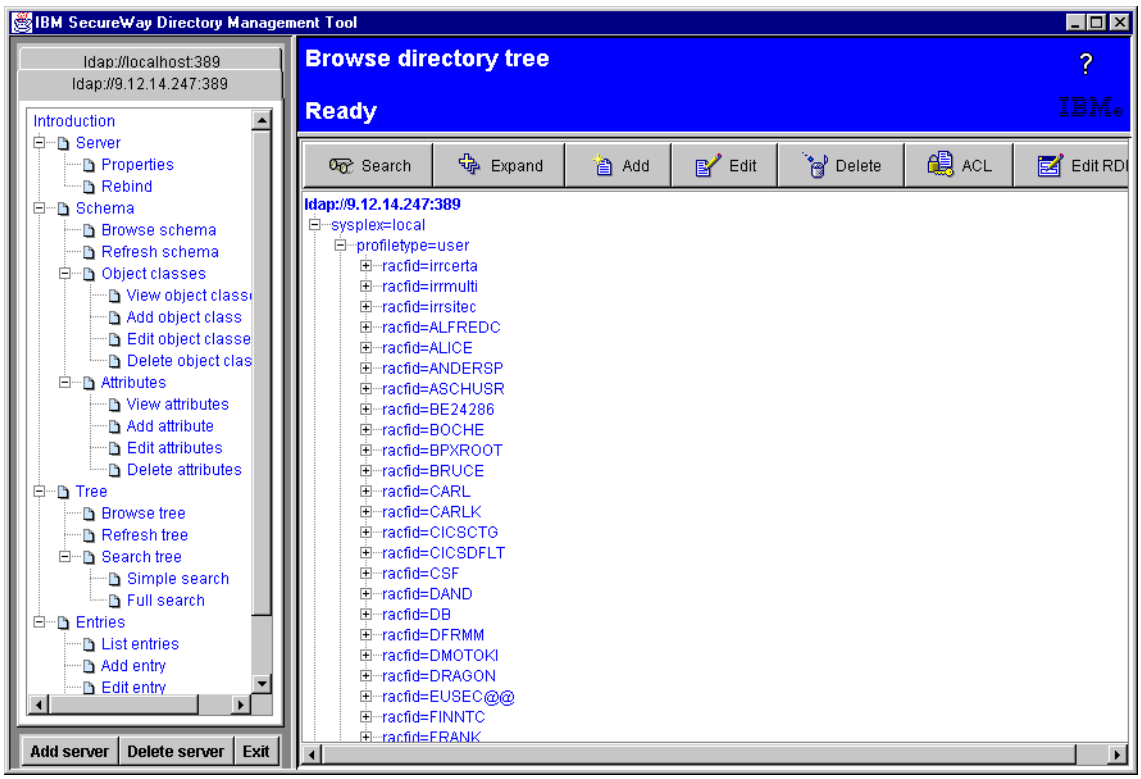


Figure 271. DMT window showing all defined RACF users

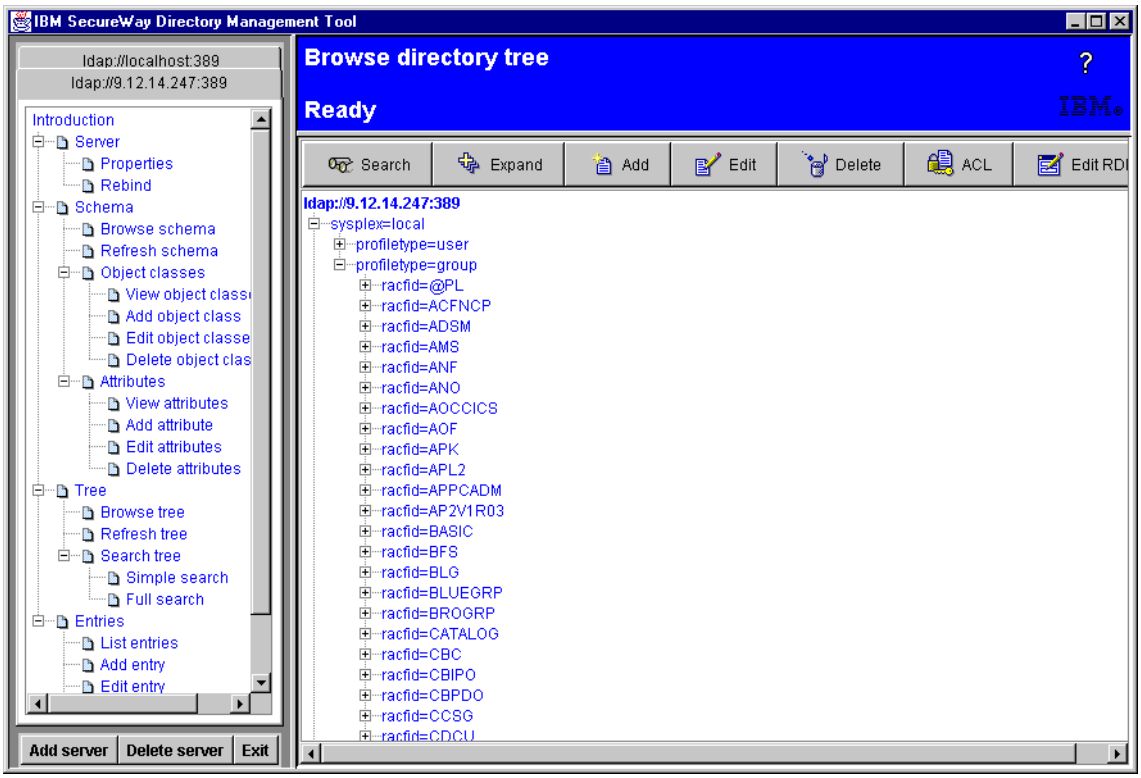


Figure 272. DMT windows showing all RACF groups defined

There are many more features of the DMT, but that is beyond the scope of this redbook.

5.6.2 LDAP Browser/Editor

The LDAP Browser/Editor developed by Jarek Gawor not only offers browsing capabilities for an LDAP Server, but also editing capabilities, which we will explore in this section.

As stated earlier this tool can be downloaded from
`http://www-unix.mcs.anl.gov/~gawor/ldap`

We selected the browser.zip file for download. Follow the instructions on the related webside to install the LDAP Browser/Editor. We installed the LDAP Browser/Editor on a Windows NT machine with JDK 1.2 installed. We updated the runnit2.bat file to include the directory where the JDK was installed. The content of our runnit2.bat is shown in Figure 273.

```
@echo off
set JAVA_HOME=d:\jdk1.2.2
set JNDI_LIB=.\lib\ldap.jar;.\lib\jndi.jar;.\lib\providerutil.jar;.\lib\ldapbp.jar
set NETSCAPE_LIB=.\lib\ldapjdk.jar;.\lib\ldapfilt.jar
set COMMON=%JNDI_LIB%;%NETSCAPE_LIB%;.\plugins;.
set EXEC=Browser.jar Browser
set CMD=%JAVA_HOME%\bin\java -classpath %COMMON%;%EXEC%
echo %CMD%
%CMD%
```

Figure 273. RUNNIT2.BAT example

The first time you start the LDAP Browser/Editor it will try to connect to the University of Michigan, as shown in Figure 274.

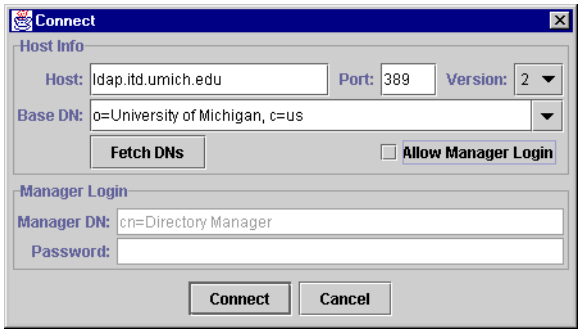


Figure 274. LDAP Browser/Editor connection window

Off course that is not where we want to connect to. The default configuration file is called browser.cfg and contains the configuration settings, as shown in Figure 275.

```
#LDAP Browser v2.7 config file
#Thu Jul 15 09:49:13 CDT 1999
managerdn=cn=Directory Manager
version=2
managerlogin=no
password=
host=ldap.itd.umich.edu
basedn=o=University of Michigan, c=us
port=389
autoconnect=no
```

Figure 275. browser.cfg default settings

To change the default settings to your server, you can change the information on the window presented (see Figure 274), as shown in Figure 276.

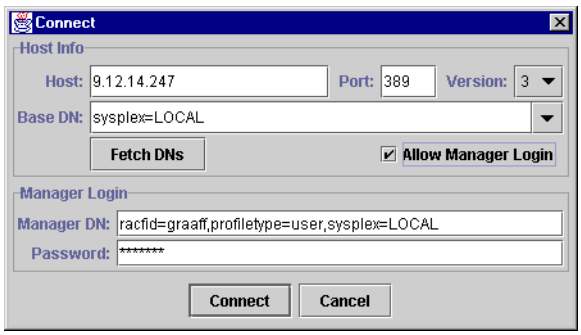


Figure 276. LDAP Browser/Editor window with our configuration information

After clicking **Connect**, a connection is attempted with the OS/390 LDAP Server.

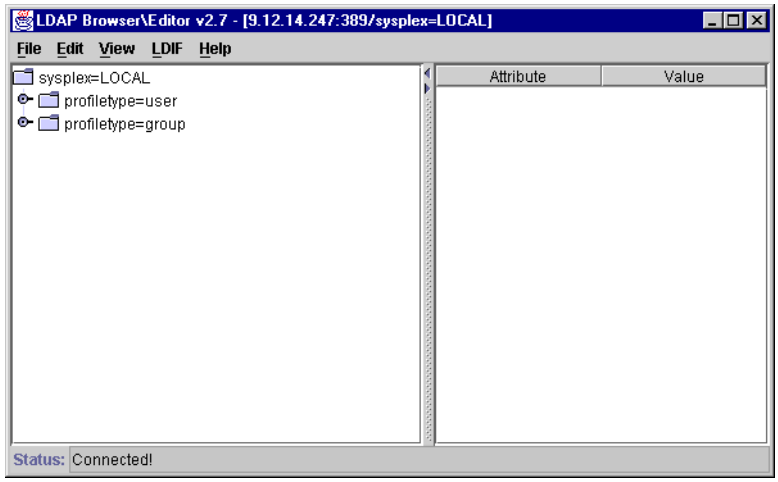


Figure 277. LDAP Browser/Editor main window

Figure 277 shows the main window and display, in this case the `sysplex=local` DIT, and the two leaves profiletype=user and profiletype=group.

You have the ability now to save this configuration using the File menu option and clicking **Save Config**, as shown in Figure 278.

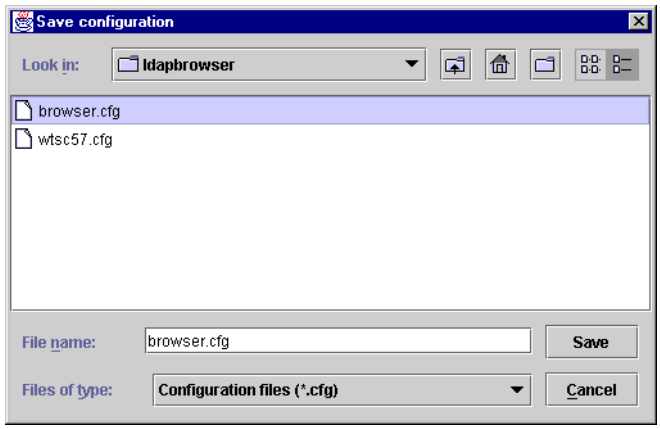


Figure 278. Save configuration window

After pressing **Save**, the browser.cfg file contains our configuration information, as shown in Figure 279.

```
#LDAP Browser v2.7 config file
#Thu Feb 10 17:06:31 EST 2000
basedn=sysplex\=LOCAL
password=racf99p
port=389
managerlogin=yes
version=3
host=9.12.14.247
autoconnect=no
managerdn=racfid\=graaff,profiletype\=user,sysplex\=LOCAL
```

Figure 279. browser.cfg with new configuration information

After we make the connection to our OS/390 LDAP Server, we can take a look again at our RACF information.

Note: Remember, we connected using a RACF userid (graaff) that has RACF special authorization, to be able to see RACF information.

Again we see the same information that we saw when we used the other tool, but now we can also see some attribute information and values, as shown in Figure 280.

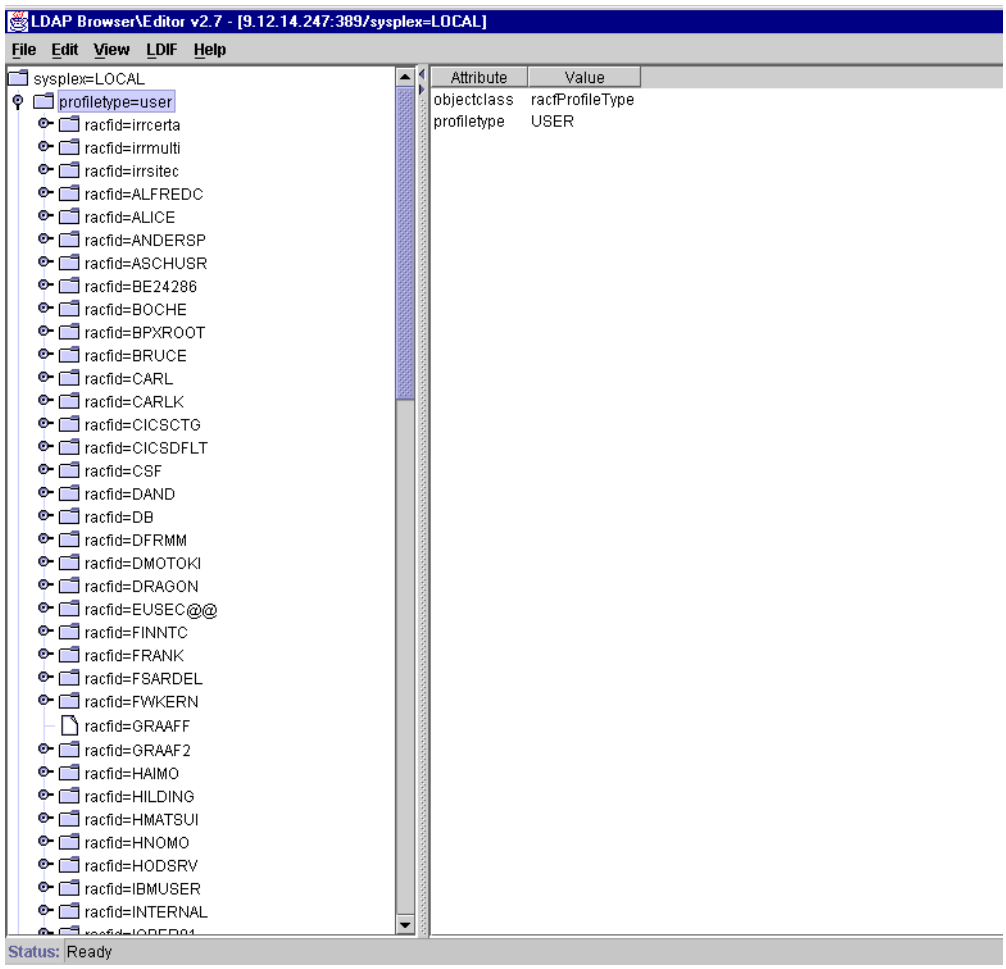


Figure 280. LDAP Browser/Editor management window

We can then select one of the users, and display attribute information and values for that user, as shown in Figure 281 on page 214.

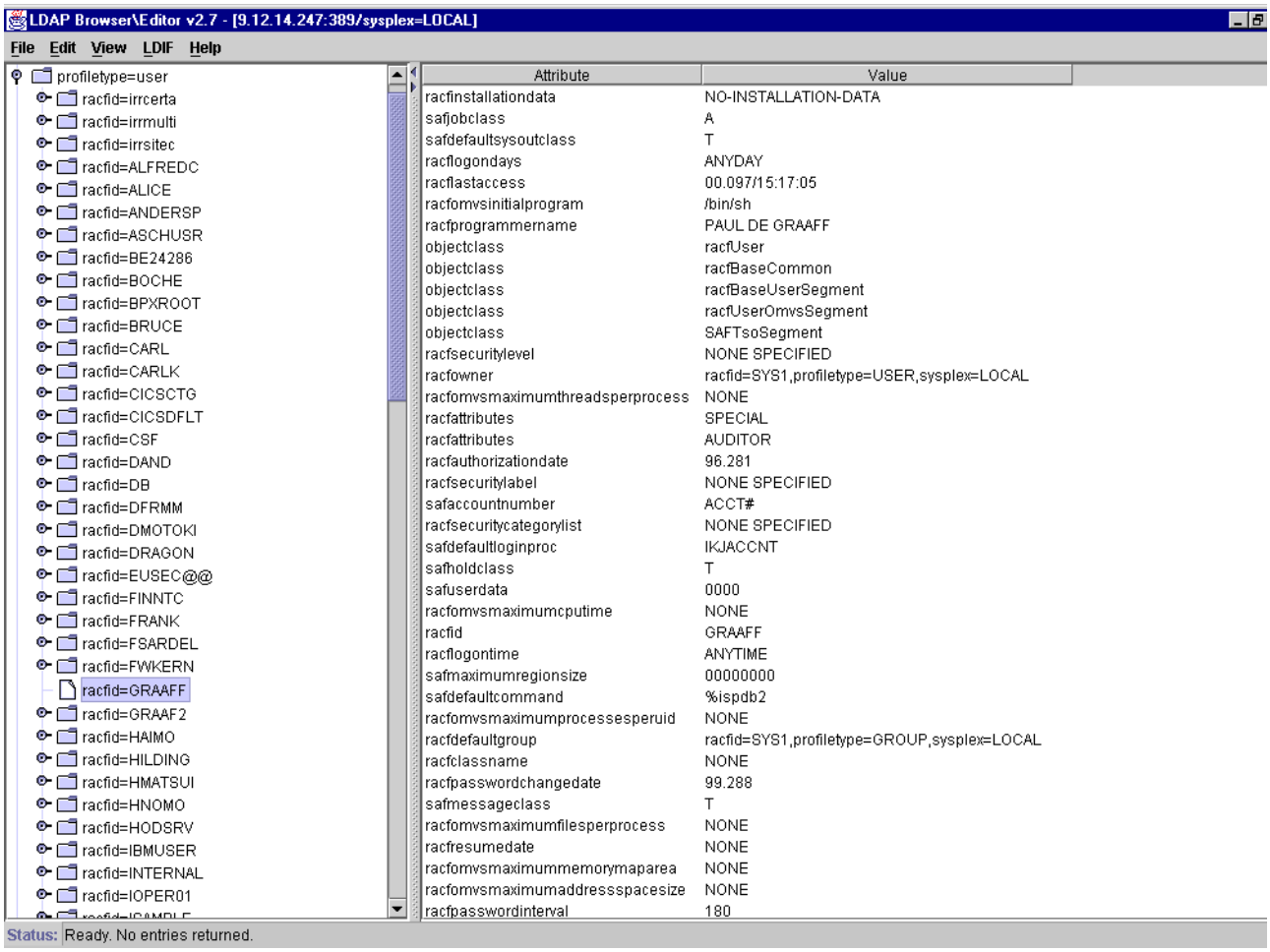


Figure 281. LDAP Browse/Editor window showing information about RACF user graaff

For a long time, users have wanted a GUI interface to RACF. Now you have it and you can manage RACF users and groups through the LDAP interface.

Next we give user GRAAFF the RACF OPERATIONS attribute using the LDAP Browser/Editor.

To add or modify attribute values:

1. Select the attribute.
2. Right-click the mouse and select **Manager**.
3. Select **Edit Attribute**, as shown in Figure 282 on page 215.

In our example we select the racfattributes field with a value of SPECIAL and we will add OPERATIONS to it.

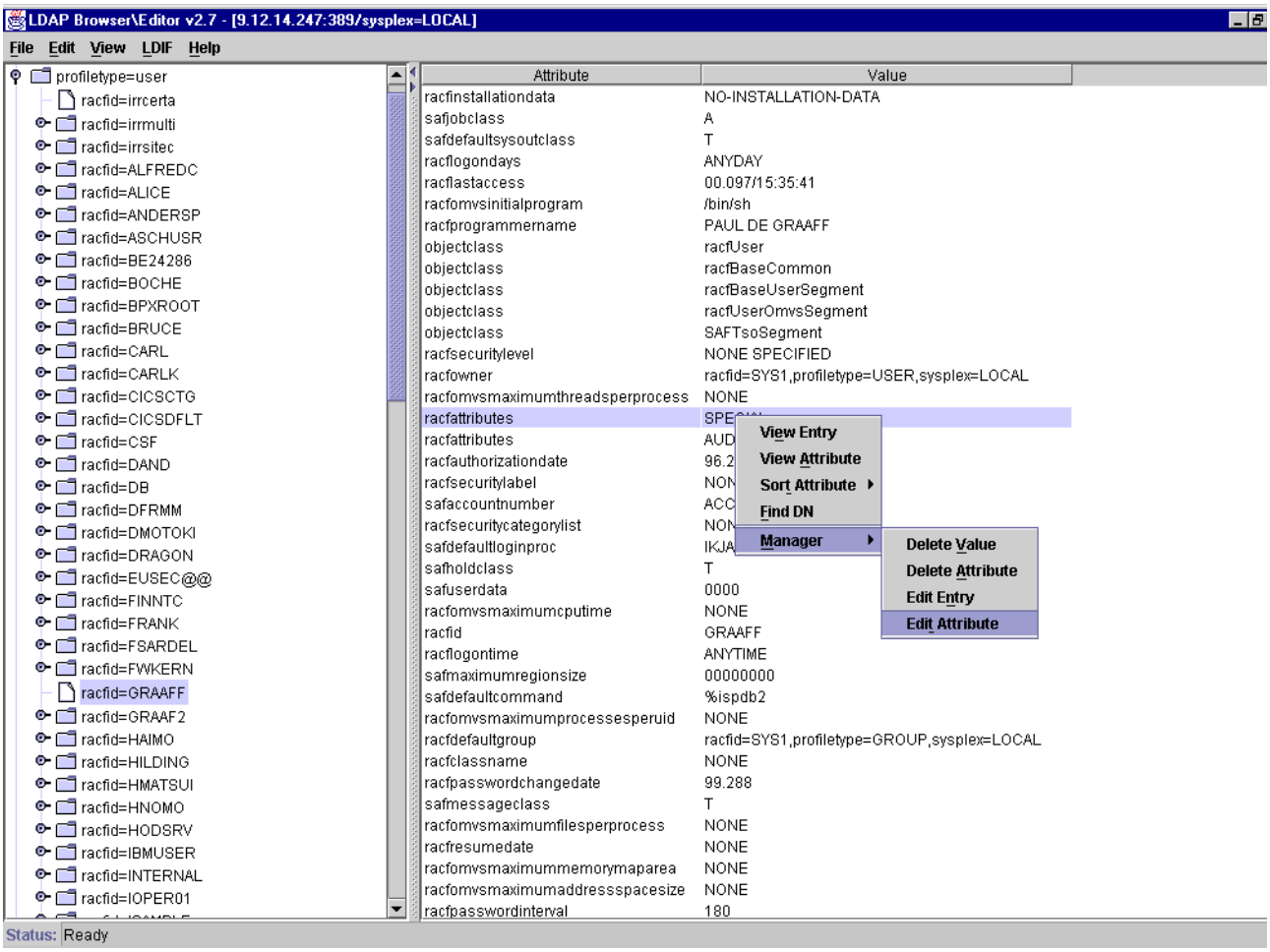


Figure 282. LDAP Browser/Editor window to manipulate attribute values

When we select **Edit Attribute**, the window shown in Figure 283 is displayed.

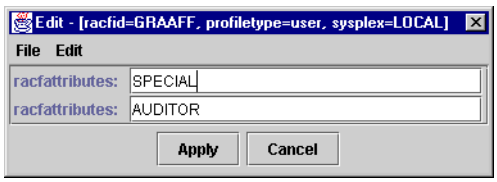


Figure 283. LDAP Browser/Editor Edit Attribute window

We can now add the OPERATIONS attribute to user GRAAFF, by typing it next to the SPECIAL value, as shown in Figure 284.

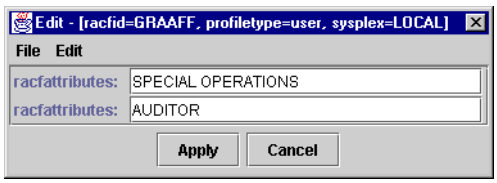


Figure 284. LDAP Browser/Editor window to add attribute values

When we click **Apply**, the RACF operations attribute is added to user GRAAFF, as shown in Figure 285. The LDAP Browser/Editor performs the update, retrieves the new values, and updates the window showing the updated values.

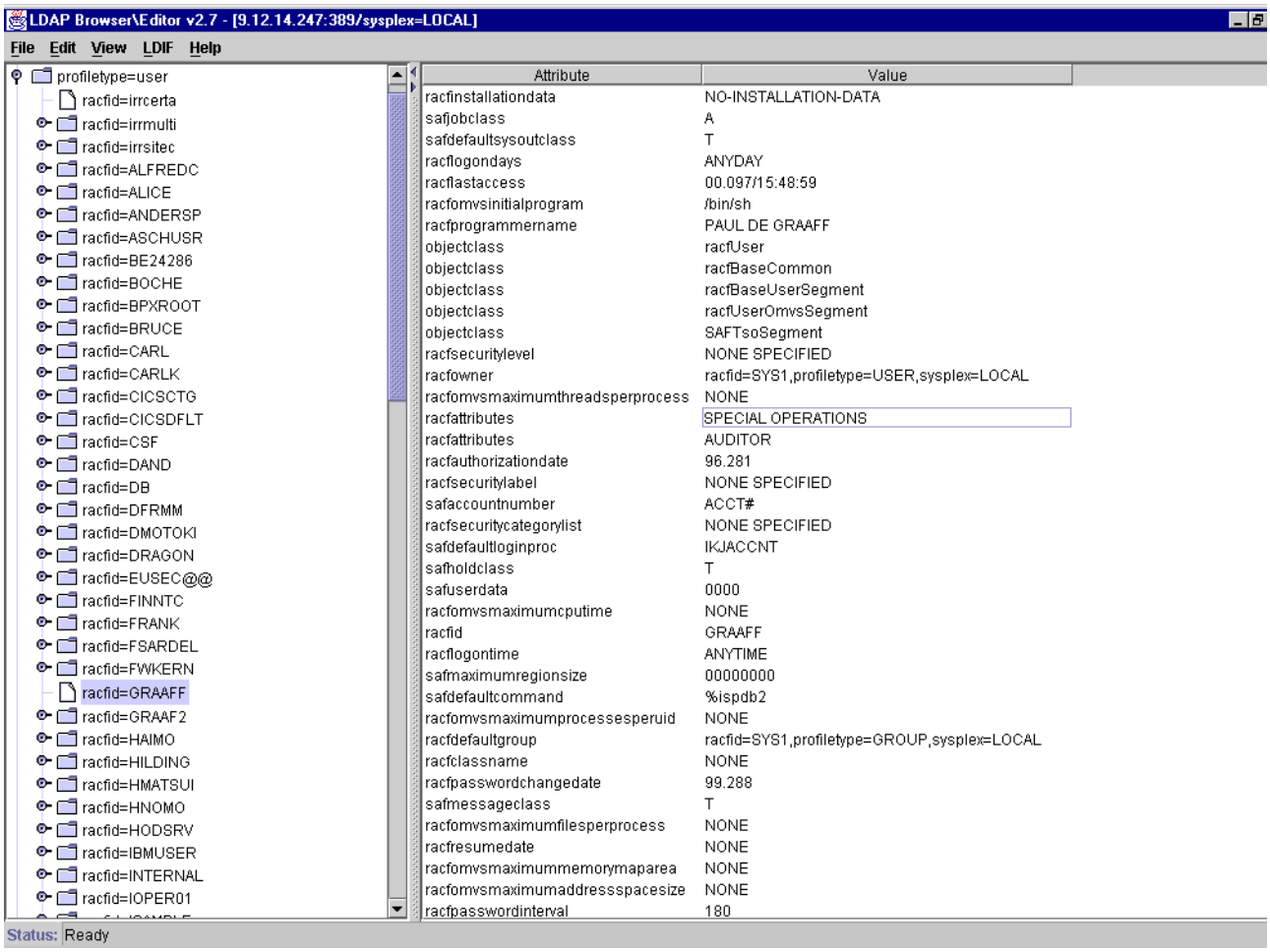


Figure 285. LDAP Browser/Editor window after RACF update

The LDAP Browser/Editors offers a lot more functionality, but that goes beyond the scope of our redbook.

Chapter 6. RACFICE reporting made easy

This chapter describes the RACFICE samples shipped in the OS/390 Security Server R2.8 SYS1.SAMPLIB. The intent is to provide a background on what the sample reports are and how to use them.

6.1 The background of RACFICE

Since RACF 1.9.2 the RACF Report Writer (RACFRW) is no longer the IBM-recommended utility for processing RACF audit fields. The utility of choice for processing SMF event codes since RACF 1.9.2 has been the RACF SMF Data Unload Utility (IRRADU00). See *Security Server (RACF) Auditor's Guide,* SC28-1916, and *Security Server (RACF) Macros and Interfaces,* SC28-1914 for additional information.

IRRADU00 enables installations to create a sequential file from the security-relevant audit data. The output files were intended to be used as input data to report writer languages, sort/merge utilities, and relational data managers such as DB2 to process complex inquiries. For instance, member IRRADUTB in SYS1.SAMPLIB contains examples that create a separate DB2 table for each record type. A previous ITSO publication (*OS/390 Security Server Audit Tool and Report Application*, SG24-4820) documents how to create ISPF interfaces to QMF queries against DB2 tables containing the IRRADU00 output.

Since RACF 1.9.0 the RACF Database Unload Utility (IRRDBU00) has been available for use. See *Security Server (RACF) Macros and Interfaces*, SC28-1914, and *Security Server (RACF) Systems Administrators Guide*, SC28-1915 for more information about this utility, which creates a sequential file of the RACF database. The ability to manipulate this sequential file for reporting purposes gives rise to a great degree of flexibility for each installation. The degree of flexibility in report writing using IRRDBU00 and IRRADU00 is far greater than what could be done using a standard report writer such as RACFRW. However, the flexibility has generated a degree of complexity resulting in requests from some installations to provide examples of the types of reports which can be generated. Thus, since 1994 the RACF development team put samples of what can be done with the output from IRRADU00 and IRRDBU00 on the internet. See the download section of `www.ibm.com/S390/racf` to obtain these samples. The samples were written using IBM's DFSORT utility called ICETOOL, and the sample reports became known as RACFICE. For more information about the ICETOOL utility, see *DFSORT Application Programming Guide,* SC33-4035, and the *ICETOOL Mini-User Guide* which is available on the DFSORT home page at `www.ibm.ocm/storage/dfsort/`

Beginning with OS/390 Security Server V2.8 the sample ICETOOL control statements are now shipped in SYS1.SAMPLIB. This provides a greater level of support from IBM. Since these samples are now officially part of the OS/390 Security Server, IBM's full level 1 and level 2 support are available to resolve issues with these samples.

6.2 Background of ICETOOL

ICETOOL was introduced in Release 11 of DFSORT and enhanced in every subsequent release of the product. ICETOOL works together with the basic DFSORT functions to provide a flexible report writer. ICETOOL uses DFSORT's sorting, copying, merging and record selection functions. Additionally, ICETOOL adds new features for reporting, formatting, statistics, and dealing with duplicates. The end result is a quick method of generating reports.

DFSORT Release 14 allows you to use symbol mappings for RACF records available from IBM as described in 6.5.6, "Using symbols for DFSORT/ICETOOL" on page 234.

The RACFICE tool uses four of the twelve available ICETOOL operators. The four used by RACFICE are:

1. SORT
2. COPY
3. DISPLAY
4. OCCURS

The other ICETOOL operators are:

1. COUNT
2. DEFAULTS
3. MODE
4. RANGE
5. SELECT
6. STATS
7. UNIQUE
8. VERIFY

6.2.1 SORT and COPY operators

RACFICE uses the COPY operator to select specific records. Secondarily, the SORT operator is used to select specific records and order (sort) the records. The SORT and COPY operators specify the DDNAMEs of the input and output data sets and the DDNAME containing the ordering and selection criteria.

The RACFICE reports use standard DFSORT record selection syntax. Figure 286 shows the standard COPY or SORT control statement.

```
COPY    FROM(inddn) TO(outddn) USING(cccc+'CNTL')
                -or-
SORT    FROM(inddn) TO(outddn) USING(cccc+'CNTL')
```

Figure 286. ICETOOL Sort/Copy

The lowercase variables have the following meanings:

INDDN The input DDNAME as found in the JCL for this step. Typically this is either ADUDATA or DBUDATA, indicating what kind of input to expect.

OUTDDN The output DDNAME as found in the JCL for this step. This DDNAME is typically a TEMP file as used in the RACFICE tool.

The TEMP file is usually passed as input to the DISPLAY or OCCURS reporting functions.

CCCC ccccCNTL is the DDNAME for the DFSORT control statements found in the JCL for this step (the suffix of CNTL is assumed). Therefore, since the RACFICE uses a USING(RACF), the corresponding DDNAME in the JCL is RACFCNTL

The following DFSORT cards are typically found in the members with a CNTL suffix. The example RACFICE PROC in Figure 290 on page 222 illustrates how the DDNAMEs relate to the data set names. The following example describes what is found in the CNTL suffixed members.

```
SORT    FIELDS=(start,length,type,sequence)
INCLUDE COND=(start,length,type,eval,value,and|or,
              start,length,type......)
```

Figure 287. ICETOOL control card description (CNTL suffix members)

Figure 287 shows a sample description of the DFSORT operands. The syntax of the FIELDS operand of the SORT statement and the COND operand of the INCLUDE statement describe the fields to be used in a similar manner. The SORT operator will be ordering/sorting the records based on the field descriptions in the FIELDS operand. The INCLUDE statement will be selecting records based on the criteria descriptions in the COND operand. A brief description of the subset of DFSORT's syntax used for RACFICE follows. For complete details see *DFSORT Application Programming Guide*, SC33-4035, for your version of DFSORT.

START Starting position of the string
LENGTH Length of the string
TYPE Description of the data
- "CH" Character data
- "SS" Substring data

SEQUENCE Order the data is to be in
- "A" Ascending order
- "D" Descending order

EVAL Type of comparison
- "EQ" is equal
- "NE" is not equal
- "LT" is less than
- "LE" is less than or equal to
- "GT" is greater than
- "GE" is greater than or equal to

Using Figure 299 on page 232 as an example, you can see that we are looking for the string SLACKER or TEEDEE or PEKKAH, starting in column 63 for a length of 8. If you cross-reference that to the *OS/390 Security Server (RACF) Macros and Interfaces,* SC28-1914, you can see that we are looking for the user IDs of SLACKER, TEEDEE and PEKKAH. Since there are no other selection criteria, this simply selects any audit event which occurred for any of those 3 user IDs.

Note

The ICETOOL includes the record descriptor word (RDW) when calculating the column. The RDW has a length of 4. The record layouts described in the *OS/390 Security Server (RACF) Macros and Interfaces* manual do not include the RDW. Therefore, add 4 to the location found in the manual to get the accurate column displacement used with ICETOOL.

6.2.2 DISPLAY

The DISPLAY statement describes the “Look” of the report. The DISPLAY statement immediately follows the SORT/COPY statement discussed in the previous section. Additionally, the input to the DISPLAY is the output from the SORT/COPY.

The DISPLAY statement defines how you want the top of each page to look: you can format such things as the date and time the report was run, the title of the report, and the data which is to appear in the report. The order in which you put the commands will influence the look of the report. “Typical” RACFICE commands and formats used are shown in Figure 288.

```
DISPLAY FROM(indd) LIST(listdd) -
        PAGE -
        TITLE('string')-
        DATE(abcd) -
        TIME(abc)  -
        BLANK -
        ON(start,length,type)  HEADER('string') -
        ON(start,length,type)  HEADER('string') .....
```

Figure 288. DISPLAY control card description

The lowercase variables have the following meanings:

INDD The input DDNAME as found in the JCL for this step, this is typically a TEMP file passed from the SORT/COPY portion of the ICETOOL.

LISTDD The output DDNAME for the Report, as found in the JCL for this step. As shipped, this points to SYSOUT.

STRING Any string of characters (1 to 50 long).

ABCD Format of the date — any combination of M (month) D (day) Y (year) and 4 (4 digit year). Specified only once.

ABC Format of the time:
12: indicates 12 hour clock with a.m./p.m.
24: indicates use of 24 hour clock.

BLANK Insert a blank line.

Note

The start, length, and type are the same as in 6.2.1, “SORT and COPY operators” on page 218.

Refer to Figure 291 on page 223 for an example of the DISPLAY statement used on the VIOL ICETOOL report. Figure 487 on page 409 shows the corresponding report generated as a result of the VIOL DISPLAY statement.

6.2.3 OCCURS

The OCCURS statement has the same syntax as the DISPLAY statement, but it adds an additional variable, the HIGHER command. The HIGHER command adds to the flexibility of the ICETOOL by allowing selection into the report of only those records which are above a certain threshold.

Refer to Figure 296 and Figure 297 on page 231 for an example of the use of the HIGHER command. In Figure 289 just the HIGHER statement with VALCNT has been cut out.

```
ON(63,8,CH) HEADER('User ID') -
ON(VALCNT)  HEADER('Number of Incorrect Passwords') -
HIGHER(3)
```

Figure 289. Sample HIGHER statement

Careful examination of the LOGFCNTL control cards in Figure 296 and Figure 297 shows that you have already selected for JOBINIT events with an event qualifier of INVPSWD event types. This means you are already selecting for JOB initiation events that failed due to an invalid password. Additionally, you now have added the criteria of HIGHER(3) against the same ID. This means you want to report on only those IDs which had more than 3 failed JOB initiation events due to an invalid password. (TSO signons are considered a JOB initiation event.) The resulting report is in Figure 479 on page 407.

6.3 RACFICE description

This section describes the components which make up the RACFICE samples.

The RACFICE samples are specific to DFSORT's ICETOOL. The control cards and techniques can easily be modified to function with other sort/merge utilities. For the current discussion only the DFSORT ICETOOL control cards will be considered. Though these control cards are specific to DFSORT's ICETOOL, they can easily be modified to function with other sort/merge utilities. Since IBM's intention has always been to provide flexible reporting, the ICETOOL samples should provide further ideas on how best to exploit IRRADU00 and IRRDBU00 at your installation.

The RACFICE PDS is a self-contained "kit" of JCL, PROCs and control cards needed to produce the RACFICE sample reports. The assumption is that the IRRDBU00 and the IRRADU00 utilities have already been run and that you have saved the output from those utilities.

Attention

Be careful running IRRADU00 and IRRDBU00. The IRRADU00 is based upon the SMF dump utility, IFASMFDP. Running this against your SMF files should be coordinated with your installation's current procedures. Likewise, running the IRRDBU00 utility may create performance problems if done without coordination with other of your installation's procedures. It is likely you will run both utilities, IRRADU00 and IRRDBU00 against offline copies of the SMF file and RACF database respectively.

The RACFICE PDS contains:

- RACFICE PROC
- The JCL that invokes the PROC and controls with which reports are run.
- Control cards for DFSORT's ICETOOL.
- Three independent RACFICE procedures: $CFQG, $CHLQ, and $ULAST90.

6.3.1 RACFICE PROC

The RACFICE PROC name is RACFICE. You should not need to modify this member, unless you need to change the size allocation on the TEMP DDNAMEs or some similar trivial change which may be unique to your installation. The PROC is shown in Figure 290.

```
//*-------------------------------------------------------------------
//RACFICE  PROC  REPORT=                      Name of report
//*************************************************************
//* These samples are provided for tutorial purposes only. **
//* This code has not been submitted to formal IBM testing.**
//* This source is distributed on an "as-is" basis,        **
//* without any warranties either expressed or implied.    **
//*                                                        **
//* (c) Copyright 1994, 1999 IBM Corporation               **
//*************************************************************
//*************************************************************
//*         See the "DFSORT Application Programming:       **
//*         Guide" (SC33-4035) for more information on     **
//*         DFSORT and ICETOOL.                            **
//*************************************************************
//*                                                        **
//* DDNAME   USE                                           **
//* -------  ---------------------------------------       **
//* ADUDATA  IRRADU00 output that is input to the report   **
//* DBUDATA  IRRDBU00 output that is input to the report   **
//* DFSMSG   DFSORT diagnostic and informational messages  **
//* PRINT    Report output                                 **
//* TEMP0001 Work data set                                 **
//* TOOLMSG  ICETOOL diagnostic and informational messages **
//*************************************************************
//RACFICE   EXEC PGM=ICETOOL
//TOOLMSG   DD DUMMY
//PRINT     DD SYSOUT=*
//DFSMSG    DD DUMMY
//ADUDATA   DD DISP=SHR,DSN=&ADUDATA
//DBUDATA   DD DISP=SHR,DSN=&DBUDATA
//TEMP0001  DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(20,5,0))
//TOOLIN    DD DISP=SHR,DSN=&ICECNTL(&REPORT)
//RACFCNTL  DD DISP=SHR,DSN=&ICECNTL(&REPORT.CNTL)
```

Figure 290. Example RACFICE PROC

6.3.2 RACFICE JCL

The JCL that invokes the RACFICE PROC is contained in member $$CNTL$$. You will need to modify some of the JCL statements in this member prior to running this at your installation. Refer to Appendix B, "$$CNTL$$ member" on page 397 for the complete listing of the $$CNTL$$ member.

6.3.3 RACFICE control cards

The control cards for the RACFICE tool are in pairs. At this point you have already observed the symbolic definitions in the RACFICE PROC. The symbolic &REPORT is used as the member name in the &ICECNTL PDS for the TOOLIN DDNAME, and the &REPORT symbolic with the characters CNTL is used as the member name for the RACFCNTL DDNAME. Therefore, for the VIOL report there are two members used from the &ICECNTL PDS your installation has defined. In the case of VIOL there is a VIOL and a VIOLCNTL member in the &ICECNTL PDS. For a complete list of all the “paired” members, refer to 6.4, “Sample reports shipped in SYS1.SAMPLIB” on page 224. There you will find a complete list of the report names. In conclusion, note that for every report name there are 2 members.

6.3.3.1 Example of control cCard use of the VIOL report

To further explain the control cards, lets take a look in more detail at the VIOL RACFICE report. There are 2 members in the RACFICE PDS used to produce the VIOL report: a VIOL member and a VIOLCNTL member.

Following is the VIOL member:

```
************************************************************************
* Name: VIOL                                                           *
*                                                                      *
* Find all of the resource accesses which represent a violation.       *
************************************************************************
 SORT    FROM(ADUDATA) TO(TEMP0001) USING(RACF)
 DISPLAY FROM(TEMP0001) LIST(PRINT) -
         PAGE -
         TITLE('VIOL: Access Violations')-
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(32,10,CH)  HEADER('Date') -
         ON(23,8,CH)   HEADER('Time') -
         ON(14,8,CH)   HEADER('Result') -
         ON(63,8,CH)   HEADER('User ID') -
         ON(286,30,CH) HEADER('Resource Name') -
         ON(578,8,CH)  HEADER('Class') -
         ON(564,6,CH)  HEADER('Volume') -
         ON(605,30,CH) HEADER('Profile')
```

Figure 291. Example ICETOOL control card VIOL

```
SORT    FIELDS=(32,10,CH,A,23,8,CH,A,63,8,CH,A)
INCLUDE COND=(5,8,CH,EQ,C'ACCESS',AND,
              48,3,CH,EQ,C'YES')
OPTION  VLSHRT
```

Figure 292. Example ICETOOL control card VIOLCNTL

The VIOL member is the member that defines the report headings, gives information about which DDNAME is input (ADUDATA), which DDNAME is the temporary output DDNAME (TEMP001) and points to the ’CNTL’ DDNAME (which is RACFCNTL based upon the USING statement). Notice that the ’CNTL’ suffix is assumed by the ICETOOL. This information is provided on the ’SORT’

statement. Refer to Figure 290 on page 222 for the data set names that each of the SORT statements use. The JCL takes the DDNAMEs from the SORT statement, thus defining the data set names which will be used for the VIOL RACFICE report.

6.3.4 Stand-alone RACFICE reports

There are 3 stand-alone RACFICE reports shipped in SYS1.SAMPLIB. These procedures do not use the RACFICE PROC. The procedures are $CFQG, $CHLQ and $ULAST90. They all use IRRDBU data as input. The reason they can not use the RACFICE PROC is due to the uniqueness of their processes. Refer to 6.4.3, “Stand-alone sample reports” on page 227 for more details.

6.4 Sample reports shipped in SYS1.SAMPLIB

This section describes the reports that are shipped in SYS1.SAMPLIB. It also describes why an installation may be interested in using the report.

The reports are broken into two major groups. The first group is those reports which are based on the output from IRRDBU00, the second group is those reports which are based on the output from IRRADU00. There is a third, minor group of reports — the stand-alone reports. Unlike the first two groups, these three reports do not use the RACFICE PROC.

The 4 character report names correspond to the member names within the RACFICE PDS. There are 2 members for each report. The first member is simply the 4 character report name, the second member is the 4 character report name with the ’CNTL’ suffix added to the end of the name. So the first report, ALDS has member names ALDS and ALDSCNTL, which contain the control cards for DFSORT’s ICETOOL.

6.4.1 RACFICE samples from IRRDBU00 output

This section lists the reports available based upon the output of the IRRDBU00 utility and the purpose of each report. This utility is generated off the RACF database.

ALDS Users with Alter authority to discrete data set profiles.
Lists possible sources of problems with discrete profiles. When a discrete data set is deleted the RACF profile is also deleted. If your installation has the SETR ADDCREATOR authority in effect, this may indicate that administrators are not removing themselves from the access lists after they create the data set profiles.

ASOC Users with explicit RACF Remote Sharing Facility (RRSF) associations defined.
Identifies users who can direct commands.

BGGR Discrete general resource profiles with generic characters.
Finds error profiles, for example, profiles which aren’t protecting what you may think they are protecting.

CCON Count of users’ connections, flagging those users with more than n connections.
Finds excessive group connections which may be negatively impacting performance.

CGEN Count of general resource profiles.
General information about your RACF database.

CPRO Count of all profiles.
General information about your RACF database.

CONN Users with group privileges above use.
Identify users with above normal privileges.

IDSC Data set conditional access list entries with an ID(*) entry of other than NONE.
Identifies data set profiles that allow any RACF-authenticated user to access data.

IDSS Data set standard access list entries with an ID(*) entry of other than NONE.
Identifies data set profiles that allow any RACF-authenticated user to access data.

IGRC General resource conditional access list entries with an ID(*) entry of other than NONE.
Identifies general resource profiles that allow any RACF-authenticated user to access a resource.

IGRS General resource standard access list entries with an ID(*) entry of other than NONE.
Identifies general resource profiles that allow any RACF-authenticated user to access a resource.

OMVS List of IDs which have a UNIX System Services (OMVS) segment defined.
Identifies users who can use OS/390's UNIX Systems Services with a non-default UID.

SUPU UNIX System Services super users (UID of zero).
Identifies users who have extraordinary privileges within the OS/390 UNIX Systems Services environment.

UADS Data set profiles with UACCs other than NONE.
Identifies any data set profiles that allow any user — regardless of whether the user has been RACF-authenticated or not — access to data.

UAGR General resource profiles with UACCs other than NONE.
Identifies general resource profiles that allow any user — regardless of whether the user has been RACF-authenticated or not — access to a resource.

UGLB Users with extraordinary global authorities.
Identifies users with extraordinary global RACF authority, for example Special, Operations or Audit attribute.

UGRP Users with extraordinary RACF group authorities.
Identifies users with extraordinary Group RACF authority, for example group-Special, group-Operations or group-Audit privileges.

UIDS UNIX System Services UIDs which are used more than once.
Identify UNIX Systems Services users who are sharing authority characteristics.

URVK Users which are currently revoked.
Identifies users who are currently revoked.

WNDS Dataset profiles in warning mode.
Since access is being allowed to these dataset profiles, it is a good idea to report which data sets are affected.

WNGR General resource profiles in warning mode.
As with the data sets in warning mode, it is a good idea to track which resources are available for use by all IDs in your installation.

6.4.2 RACFICE samples from IRRADU00 output

This section describes the sample reports available based upon the output of the IRRADU00 utility, as well as the purpose of each report. This utility is based upon RACF event records generated by SMF.

ACD$ Users who are using automatic command direction.
Identifies users who are using the RACF Remote Sharing Facility (RRSF).

CADU Total count of events by category.
A good overview of all the types of audit events for the current reporting period. This is a way to find high level event anomalies.

CCMD A count of all RACF commands issued by USER.
A good high-level look at which of your administrators is doing what sort of commands.

ECD$ Users who are directing commands explicitly using the RACF Remote Sharing Facility (RRSF)

LOGB Users who log on with LOGON BY.
Identifies users who are logging on as another user.
Note: This is a VM only function.

LOGF All users with excessive incorrect passwords.
Identifies users who exceed a “bad password” threshold. This threshold is independent of the SETROPTS PASSWORD(REVOKE(nn)) value. The sample is shipped with a value of 3.
Note: This value continues to accrue; for example, it is not reset to zero by a valid signon.

OPER Accesses allowed because the user has the OPERATIONS attribute.
Audit users with the OPERATIONS attribute.

PWD$ Users who are using password synchronization, by using the RACF Remote Sharing Facility (RRSF)

RACL Show RACLINK audit records.
Identifies users who are using the RACF Remote Sharing Facility (RRSF).

RINC Show all classes at IPL.
Gives a great view of whether classes are active, RACLISTED, generics and GENCMD activated. Based upon the CLASNAME event recorded at IPL.

SELU Identify all audit records for a specific user.
Audit a specific user. The sample uses IBMUSER.

SPEC Accesses allowed because the user has SPECIAL attribute.
Audit users with the SPECIAL attribute.

TRMF Excessive incorrect passwords from a terminal.
Audit of possible intrusion by guessing passwords but switching user IDs to prevent the revocation of an ID. The sample is shipped with a value of 3.

VIOL Access Violations.
Identifies failed events.

WARN Accesses allowed because the resource is in WARNING mode.
Identifies events which were allowed but which you may want to prevent in the future or explicitly allow by changing a profile’s access list.

6.4.3 Stand-alone sample reports

This section describes the stand-alone reports in the RACFICE samples. These reports are stand-alone since they do not use the RACFICE PROC. All three reports use IRRDBU00 data as input.

$CFQG HLQs with excessive fully qualified generic qualifiers data set profiles.
As shipped, this report will list any HLQs which have over 100 fully qualified generic profiles associated with them. An installation needs to use their judgement on what is a reasonable quantity of these definitions. If there are many of them under one high-level qualifier, it may adversely affect the entire system performance.

$CHLQ HLQs with excessive generic Profiles.
As shipped, this report will list HLQs which have over 200 generic profiles defined to them. This also could indicate a potential performance issue.

$ULAST90 This reports on any IDs added in the last 90 days.
As shipped, this report will list any IDs added in the last 90 days. This is a good mechanism for tracking administrator activity. This sample contains REXX code, which is why it does not use the RACFICE PROC.

6.5 Running RACFICE

This section describes the steps needed to run the RACFICE reporting tool at your installation. The assumption is that your installation is running IBM's sort/merge utility DFSORT. If you are using another sort/merge utility you will need to make some changes to the control cards prior to running RACFICE.

The steps needed to run the RACFICE reports are:

1. Unpack the SYS1.SAMPLIB samples. This step only needs to be done once.
2. Modify the $$CNTL$$ and any of the Stand Alone members ($CFQG, $CHLQ and $ULAST90) of RACFICE. This step only needs to be done once.
3. Run IRRADU00.
   Note: Be sure to work with the appropriate staff at your installation to get details on where the best place to access your SMF records may be. You should probably *not* run against the live SMF MAN files.
4. Run IRRDBU00.
   Note: Be sure to work with the appropriate staff at your installation to get details on how best to access your RACF database. Just as with the SMF files, you probably want to go against an offline RACF database.
5. Run the $$CNTL$$ JCL you have produced, and any of the stand-alone reports.

6.5.1 Unpack SYS1.SAMPLIB(IRRICE)

Unpacking IRRICE is very straightforward. The JCL needed to create the PDS is in SYS1.SAMPLIB(RACJCL). The steps are as follows:

1. Preallocate a PDS.
2. Modify the ICEUPDTE JCL.
3. Run the modified ICEUPDTE JCL.

6.5.1.1 Pre-allocate a PDS

Create a PDS with the following characteristics: LRECL=80, RECFM=FB using five directory blocks and five 3390 cylinders. You may want to allocate it a little bigger if you plan on modifying this PDS.

6.5.1.2 Modify the ICEUPDTE JCL

You can find the ICEUPDTE JCL in SYS1.SAMPLIB(RACJCL).

```
//*****************************************************************//
//*  ICEUPDTE:                                                      *//
//*      Unload the RACFICE reports from 'SYS1.SAMPLIB(IRRICE)'     *//
//*      and place them in the data set allocated to DDNAME         *//
//*      SYSUT2.                                                    *//
//*                                                                 *//
//*      This job must be modified to match your installation's     *//
//*      needs before it is executed.                               *//
//*                                                              @L1A*//
//*****************************************************************//
//ICEUPDTE JOB  ,'Unload RACFICE Rpts',
//         MSGLEVEL=(1,1),TYPRUN=HOLD
//UNLOAD   EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   DSN=USER01.RACFICE.CNTL,DISP=OLD
//SYSIN    DD   DSN=SYS1.PARMLIB(IRRICE),DISP=SHR
```

Figure 293. ICEUPDTE as shipped in SYS1.SAMPLIB

The ICEUPDTE JCL as shipped in SYS1.SAMPLIB is shown in Figure 293.

Copy this JCL into your own PDS prior to making any modifications. You will probably need to change the JOB card. You will certainly need to change the SYSUT2 data set name to point to the PDS you have just created.

Note

The early versions of RACFICE shipped with the SYSIN DDN pointing to SYS1.PARMLIB(IRRICE). Verify that the SYSIN DDN is SYS1.SAMPLIB(IRRICE). SYSIN should be:

```
DSN=SYS1.SAMPLIB(IRRICE),DISP=SHR
```

6.5.1.3 Run ICEUPDTE

You now should be able to submit the ICEUPDTE JCL.

6.5.2 Modify the $$CNTL$$ member of RACFICE

There should be very little customizing needed in the $$CNTL$$ member of the RACFICE PDS. Appendix B, “$$CNTL$$ member” on page 397 gives a listing of the $$CNTL$$ member. The JCLLIB statement and the 3 SET statements will need to be modified for your installation, using the following steps:

1. Create a valid JOB Card and modify the /*JOBPARM statement to something appropriate for your installation.
2. Modify the JCLLIB ORDER= Card to point to the PDS where your RACFICE PROC is located.
3. Modify the SET ADUDATA statement to point to the output from your IRRADU00 utility.

4. Modify the SET DBUDATA statement to point to the output from your IRRDBU00 utility.
5. Modify the SET ICECNTL statement to point to the PDS which contains all the ICETOOL control cards.
6. Optionally, determine if you want to comment out any of the steps or run all the steps at once.

6.5.3 Run IRRADU00

The IRRADU00 JCL can be found in SYS1.SAMPLIB(RACJCL). Figure 294 also contains a sample of this JCL. Of primary importance is to get the proper timing of this 'SMF' collection. For instance, if an installation is interested in daily, weekly or monthly data, then controlling the input will be of most importance. Also make certain that the output data set name matches the changes made to $$CNTL$$. Specifically, the SET ADUDATA statement should point to the output generated by IRRADU00. The three stand-alone utilities should point to this file as well.

```
//TSTBUDS0 JOB (999,POK),'ITSO RULES',NOTIFY=TSTBUDS,
//             CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1)
//*********************************************************************
//*                                                                   *
//*  DB UNLOAD FOR THE SMF SECURITY INFORMATION                       *
//*                                                                   *
//*********************************************************************
//STEP2    EXEC PGM=IFASMFDP
//SYSPRINT DD  SYSOUT=*
//ADUPRINT DD  SYSOUT=*
//*
//*   MY IRRADU00 FILE FOLLOWS - MUST BE INPUT TO RACFICE
//*     MATCHES SET ADUDATA STATEMENT IN $$CNTL$$ AND ALL
//*     STAND-ALONE RACFICE IRRADU00 REPORTS
//*
//OUTDD    DD  DSN=TSTBUDS.SMF.FLATFILE,DISP=(NEW,CATLG),
//    UNIT=SYSDA,SPACE=(CYL,(15,5),RLSE),
//    DCB=(RECFM=VB,LRECL=5096,BLKSIZE=0)
//*
//*    MY OFFLINE SMF FILE FOLLOWS
//*
//DUMPIN   DD  DSN=SYS1.YOUR.OFFLOAD.COPY.MAN.FILE,DISP=SHR
//DUMPOUT  DD  DUMMY
//SYSIN    DD  *
    INDD(DUMPIN,OPTIONS(DUMP))
    USER2(IRRADU00)  USER3(IRRADU86)
/*
//
```

Figure 294. Sample IRRADU00 JCL

6.5.4 Run IRRDBU00

The IRRDBU00 JCL can be found in SYS1.SAMPLIB(RACFJCL). Figure 295 on page 230 also has a sample of this JCL. As previously stated, it is advisable to run this utility against an offline copy of your RACF data base. As with the IRRADU00 utility, generating timely data will be the key to the accuracy of your reporting.

```
//TSTBUDS0 JOB (999,POK),'SMPO FOREVER',NOTIFY=TSTBUDS,
//             CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1)
//*********************************************************************
//*                                                                   *
//*  RACF CROSS REFERENCE UTILITY PROGRAM                             *
//*                                                                   *
//*********************************************************************
//STEP2    EXEC PGM=IRRDBU00,PARM=NOLOCKINPUT
//SYSPRINT DD  SYSOUT=*
//*
//*          YOUR OFFLINE RACF DATABASE FOLLOWS
//*       PROBABLY PRODUCED BY THE IRRUT200 UTILITY
//*
//INDD1    DD  DSN=MY.FAV.OFFLINE.RACF.DB,DISP=SHR
//OUTDD    DD  DSN=TSTBUDS.RACFDB1.FLATFILE,DISP=(NEW,CATLG),
//             MGMTCLAS=YOURMC,
//             SPACE=(CYL,(3,5)),
//             DCB=(RECFM=VB,LRECL=4096)
/*
//
```

Figure 295. Sample IRRDBU00 JCL

6.5.5 Modify RACFICE control cards

There are six control cards which should be uniquely considered for modification. Your installation may want to modify more than these six reports, but these all contain counts or thresholds or selection criteria which may not be appropriate for your installation. The CCON, LOGF, SELUCNTL, TRMF, $CFQG, $CHLQ and $ULAST90 should be looked at to confirm if parameters are appropriate for your installation. The modifications to these control cards may be driven by a number of factors. For instance, if the reports are based upon data collected every 24 hours the values would be different than a report which spanned 30 days. An example would be the LOGF report. This reports on all users with excessive incorrect passwords. A user entering 3 incorrect passwords over 30 days is probably not considered excessive at most installations. However, a user entering more than 3 incorrect passwords over 24 hours *may* be considered excessive. Good judgement is needed to establish what is normal versus what is excessive at each installation.

The LOGF and TRMF both have 'HIGHER(3)' statements as the last card in the member. The '3' is a count. For the LOGF it means report on any users who have entered more than '3' incorrect passwords. For the TRMF it means report any terminals where more than '3' incorrect passwords have been entered. In both cases change the 'HIGHER(3)' to a number which is appropriate at your installation. A sample of the LOGF control cards as shipped from IBM is shown in Figure 296 on page 231. The HIGHER(3) is the statement to change for use at your installation.

```
*********************************************************************
* Name: LOGF                                                        *
*                                                                   *
* Find all of the user IDs which have had an excessive number of    *
* incorrect passwords.                                              *
*                                                                   *
* The ICETOOL "HIGHER(x)" keyword is used to set the failure        *
* threshold.                                                        *
*********************************************************************
 COPY    FROM(ADUDATA) TO(TEMP0001) USING(RACF)
 OCCURS  FROM(TEMP0001) LIST(PRINT) -
         PAGE -
         TITLE('LOGF: User IDs With Excessive Incorrect Passwords')-
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(63,8,CH) HEADER('User ID') -
         ON(VALCNT)  HEADER('Number of Incorrect Passwords') -
         HIGHER(3)
```

Figure 296. LOGF as shipped from IBM

```
INCLUDE COND=(5,8,CH,EQ,C'JOBINIT',AND,
              14,8,CH,EQ,C'INVPSWD')
OPTION   VLSHRT
```

Figure 297. LOGFCNTL as shipped from IBM

The sample output for LOGF is in Figure 479 on page 407.

The $CFQG contains a 'HIGHER(100)' statement as the last control card. The $CHLQ stand-alone report contains a 'HIGHER(200)' statement as the last control card. As with LOGF and TRMF, it may be appropriate to change these to numbers which are more reflective of the needs of your installation. In the $CFQG the 'HIGHER(100)' is the threshold for the number of fully qualified generic profiles which must exist for a single HLQ for it to show up on the report. The 'HIGHER(200)' in the $CHLQ report is the threshold of generic profiles for a single high-level qualifier (HLQ) which will be reported on in the $CHLQ report. As noted earlier a high number of generic profiles for a single HLQ *may* adversely affect performance. Find an appropriate number for your installation to audit. The report generated in Figure 490 on page 410 used as a threshold 'HIGHER(2)'. In a normal installation this would be considered a ridiculously low number; however, in the lab environment this was a good way to produce a report for demonstration purposes.

There is an excellent discussion of setting up the SELUCNTL and SELU in *OS/390 Security Server (RACF) Security Administrator's Guide for V2.8,* SC28-1915. The discussion is in the chapter titled "Working with the RACF Database". A brief discussion of what can be done is also included here.

The SELUCNTL as shipped is shown in Figure 298 on page 232.

```
SORT    FIELDS=(63,8,CH,A,32,10,CH,A,23,8,CH,A)
INCLUDE COND=(63,8,CH,EQ,C'IBMUSER',AND,
              5,8,CH,NE,C'RACFINIT',AND,
              5,8,CH,NE,C'CLASNAME',AND,
              5,8,CH,NE,C'DSAF')
OPTION  VLSHRT
```

Figure 298. SELUCNTL as shipped in SYS1.SAMPLIB

Your installation is probably more concerned about users other than IBMUSER, so the first consideration should be to change the IBMUSER to a user your installation needs to audit. It may also be advisable to track more than one user at a time, thus producing a SELUCNTL which is similar to Figure 299.

```
INCLUDE COND=(63,8,CH,EQ,C'SLACKER',OR,
              63,8,CH,EQ,C'TEEDEE',OR,
              63,8,CH,EQ,C'PEKKAH')
OPTION   VLSHRT
```

Figure 299. SELUCNTL modified for your Installation

Optionally, it may be prudent to change the title of the report. The title statement is located in the SELU member. The SELU member as shipped is shown in Figure 300.

```
************************************************************************
* Name: SELU                                                           *
*                                                                      *
* Find all of the records which are applicable to a specific           *
* user ID.                                                             *
*                                                                      *
* The DFSORT "INCLUDE" statement is used to select the user ID.        *
* The DFSORT control statements are pointed to by the ICETOOL          *
* "USING" keyword.                                                     *
************************************************************************
 COPY    FROM(ADUDATA) TO(TEMP0001) USING(RACF)
 DISPLAY FROM(TEMP0001) LIST(PRINT) -
         PAGE -
         TITLE('SELU: Events Associated with a Specific User')-
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(63,8,CH)  HEADER('User ID') -
         ON(5,8,CH)   HEADER('Event') -
         ON(14,8,CH)  HEADER('Qualifier') -
         ON(23,8,CH)  HEADER('Time') -
         ON(32,10,CH) HEADER('Date') -
         ON(43,4,CH)  HEADER('System') -
         ON(175,8,CH) HEADER('Terminal') -
         ON(184,8,CH) HEADER('Jobname')
```

Figure 300. SELU as shipped in SYS1.SAMPLIB

To modify the title so it accurately reflects the criteria in the SELUCNTL statement, produce the SELU member as shown in Figure 301 on page 233.

```
*********************************************************************
* Name: SELU                                                        *
*                                                                   *
* Find all of the records which are applicable to a specific        *
* user ID.                                                          *
*                                                                   *
* The DFSORT "INCLUDE" statement is used to select the user ID.     *
* The DFSORT control statements are pointed to by the ICETOOL       *
* "USING" keyword.                                                  *
*********************************************************************
 COPY    FROM(ADUDATA) TO(TEMP0001) USING(RACF)
 DISPLAY FROM(TEMP0001) LIST(PRINT) -
         PAGE -
         TITLE('Events Associated with SLACKER, TEEDEE and PEKKAH')-
         DATE(YMD/) -
         TIME(12:)  -
         BLANK -
         ON(63,8,CH)  HEADER('User ID') -
         ON(5,8,CH)   HEADER('Event') -
         ON(14,8,CH)  HEADER('Qualifier') -
         ON(23,8,CH)  HEADER('Time') -
         ON(32,10,CH) HEADER('Date') -
         ON(43,4,CH)  HEADER('System') -
         ON(175,8,CH) HEADER('Terminal') -
         ON(184,8,CH) HEADER('Jobname')
```

Figure 301. Modified SELU title statement

The report produced by these modifications is shown in Figure 484 on page 408. For a complete list of the event codes and qualifiers which are shown in the SELU report, refer to the *OS/390 V2R8 Security Sever (RACF) Macros and Interfaces,* SC28-1914.

The $CHLQ stand-alone report contains a 'HIGHER(200)' statement as the last control card. As with LOGF and TRMF, it may be appropriate to change this to a number which is more reflective of the needs of your installation. The 'HIGHER(200)' is the threshold of generic profiles for a single high-level qualifier (HLQ) which will be reported on in the $CHLQ report. As noted earlier, a high number of generic profiles for a single HLQ *may* adversely affect performance. Find an appropriate number for your installation to audit. The report generated in Figure 490 on page 410 used as a threshold 'HIGHER(2)'. In a normal installation this would be considered a ridiculously low number; however, in the lab environment this was a good way to produce a report for demonstration purposes.

The final report which contains a *count*-type field is the $ULAST90. This sample employs the use of a REXX exec. Therefore, the number to change is not an ICETOOL parameter but a REXX variable. The variable name which can be changed is called user_age. As shipped, user_age is set to 90. This means that any USERs added in the last 90 days will show up on this report. The sample

output from this report is shown in Figure 491 on page 411. Modify the user_age variable to an appropriate value for your installation.

Table 8. RACFICE reports with count fields

Report name	Count HIGHER(n)	Description of count in context of report
$CFQG	100	Lists HLQs with more than 100 fully qualified generics
$CHLQ	200	Lists HLQs with more than 200 generic profiles
CCON	100	Lists USERs with more than 100 group connections
LOGF	3	Lists USERs with more than 3 invalid password attempts
TRMF	3	Lists Terminals which had more than 3 invalid password attempts - regardless of the USER associated with the signon attempt
$ULAST90	90[a]	Lists USERs added to the RACF database in the last 90 days.

a. This value is controlled by a REXX variable called user_age.

6.5.6 Using symbols for DFSORT/ICETOOL

Starting in R14 of DFSORT you can use symbols to replace explicit field definitions within the ICETOOL. On the RACF web page there are symbol mappings for all of the current records and events generated by IRRDBU00 and IRRADU00. Currently these symbols are not shipped with RACFICE, but if you are interested in tailoring your reports you can download RACFICE from the web and use the symbol definitions.

If you are going to use symbols, here are the steps that work to allow you to use them. For the sake of example, the OPER report has been modified.

To use DFSORT symbols there needs to be a symbol DDN in the JCL. This DDN is called 'SYMNAMES'. One method of using the predefined symbols would be:

1. Alter the RACFICE PROC.
2. Alter the $$CNTL$$ JCL.
3. Add the symbol definition members to your RACFICE PDS.
4. Create the ICETOOL control cards.

6.5.6.1 Alter the RACFICE PROC

The DDN for SYMNAMES must be added to the RACFICE PROC. Since you may not want to use symbols for all the existing reports, create a new RACFICE PROC. The reason you may not want to use symbols is that the 'ON' portion of the DISPLAY operand requires that lengths of fields fit on the printed page. The predefined symbols match the length of the fields in the IRRADU00 and IRRDBU00 records. However, certain resource fields and certain Open Edition segment fields are quite long. Therefore, if you were to use those symbols in the 'ON' statements of the DISPLAY or OCCURS operand, ICETOOL will not process that report.

```
//*-------------------------------------------------------------
//RACFICE  PROC  REPORT=                      Name of report
//RACFICE   EXEC PGM=ICETOOL
//TOOLMSG   DD SYSOUT=*
//PRINT     DD SYSOUT=*
//DFSMSG    DD SYSOUT=*
//ADUDATA   DD DISP=SHR,DSN=&ADUDATA
//DBUDATA   DD DISP=SHR,DSN=&DBUDATA
//TEMP0001  DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(20,5,0))
//TOOLIN    DD DISP=SHR,DSN=&ICECNTL(&REPORT)
//RACFCNTL  DD DISP=SHR,DSN=&ICECNTL(&REPORT.CNTL)
//SYMNAMES  DD DISP=SHR,DSN=&SYMBOLS(&INPUT.SYMBL)
```

Figure 302. RACFICE PROC modified for symbol usage.

Figure 302 is an example of a modified RACFICE PROC. The only addition has been the DDN of SYMNAMES. Notice the new JCL symbolics that have been added. They are &SYMBOLS and &INPUT. The &SYMBOLS symbolic should point to a PDS containing the symbol definition downloaded from the RACF Web site. The names of the symbol members are ADUSYMBL and DBUSYMBL. In this example the &SYMBOLS PDS is the RACFICE PDS, which already contains the PROCs, ICETOOL control cards and JCL to run RACFICE. However, your installation may chose to split each of these into separate PDSs.

6.5.6.2 Alter the $$CNTL$$ member

To handle the new JCL symbolics generated by the changes to the RACFSYM PROC, changes are needed in the $$CNTL$$ member. As with the RACFICE PROC a new $$CNTL$$ member could be created. The member created i n this sample is called $$CNTLOP.

```
//TSTBUDS0 JOB (999,POK),'ITSO RULES',NOTIFY=TSTBUDS,
//             CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1)
//*
//         JCLLIB ORDER=TSTBUDS.RACFICE.R8.CNTL
//         SET  ADUDATA=TSTBUDS.SMF.FLATFILE
//         SET  DBUDATA=TSTBUDS.RACFDB1.FLATFILE
//         SET  ICECNTL=TSTBUDS.RACFICE.R8.CNTL
//         SET  SYMBOLS=TSTBUDS.RACFICE.R8.CNTL
//*
//OPRT     EXEC RACFSYM,REPORT=OPRT,INPUT=ADU
```

Figure 303. $$CNTL$$ modified for use of DFSORT symbols

There are two additions from the base $$CNTL$$ JCL as shown in Figure 303: the SET SYMBOLS and the INPUT=ADU. The SET statement can be left alone regardless of what type of report is being produced. However, the INPUT= statement needs to be changed depending on whether an IRRADU00 or an IRRDBU00 report is being run. As it is constructed the input statement should be INPUT=ADU for IRRADU00 reports and INPUT=DBU for IRRDBU00 reports.

6.5.6.3 Add the symbol definitions

The symbol definition PDS member names from the web page are ADUSYMBL and DBUSYMBL. If you leave the name of the members the same, and simply add them to the PDS pointed to in the SET SYMBOLS JCL statement, you will have the symbols up and running.

```
*-----------------------------------------------*
*                    ACCESS                     *
*-----------------------------------------------*
ACC_EVENT_TYPE,5,8,CH
ACC_EVENT_QUAL,14,8,CH
ACC_TIME_WRITTEN,23,8,CH
ACC_DATE_WRITTEN,32,10,CH
ACC_SYSTEM_SMFID,43,4,CH
ACC_VIOLATION,48,1,CH
ACC_USER_NDFND,53,1,CH
ACC_USER_WARNING,58,1,CH
ACC_EVT_USER_ID,63,8,CH
ACC_EVT_GRP_ID,72,8,CH
ACC_AUTH_NORMAL,81,1,CH
ACC_AUTH_SPECIAL,86,1,CH
ACC_AUTH_OPER,91,1,CH
ACC_AUTH_AUDIT,96,1,CH
```

Figure 304. Example of ICETOOL symbol definitions

Figure 304 contains an example of ICETOOL symbol definitions as they are defined in the ADUSYMBL member from the RACF Web site. The symbols are mapped to specific locations and data types. The use of ICETOOL symbols eliminates the tedious and error-prone specification of column numbers.

6.5.6.4 Create the ICETOOL control cards

For this example the OPERCNTL control cards have been converted to using the ICETOOL symbols.

```
SORT    FIELDS=(63,8,CH,A)
INCLUDE COND=(5,8,CH,EQ,C'ACCESS',AND,
              91,3,CH,EQ,C'YES')
OPTION  VLSHRT
```

Figure 305. OPERCNTL as shipped in RACFICE

```
SORT    FIELDS=(ACC_EVT_USER_ID,A)
INCLUDE COND=(ACC_EVENT_TYPE,EQ,C'ACCESS',AND,
              ACC_AUTH_OPER,EQ,C'YES')
OPTION  VLSHRT
```

Figure 306. OPRTCNTL as modified to use ICETOOL symbols

Observe the differences between the OPERCNTL and OPRTCNTL. The ADUSYMBL member has defined the symbolic ACC_EVT_USER as being '63,8,CH', or, stated in English, ACC_EVT_USER is the string found at column 63 for a length of 8 and the type of data is character data. Therefore, the FIELDS definition on the SORT statement can be changed to the symbolic of ACC_EVT_USER. The resulting output is identical and is shown in Figure 480 on page 407.

Chapter 7. Java for OS/390 Security Services

Access control to OS/390 resources from a Java application can be handled with the Java for OS/390 Security Services. This package was first introduced in JDK 1.1.6 (running on OS/390 Version 2 Release 4 or above) and provided access to a minimum set of existing OS/390 UNIX services dealing with security issues.

In the second release, shipped with JDK 1.1.8 for OS/390, three new methods were added to the `PlatformUser` class that allow you to implement proper user authentication: `authenticate`, `isUserInGroup` and `checkPassword`. These methods can be particularly useful in Java server applications on OS/390 which cannot use the OS/390 Web server's security mechanisms, such as RMI or socket servers.

7.1 Overview

The Java for OS/390 Security Services allow you to:

- See if the Security Server is active.
- Extract the user ID in effect for the current running thread.
- Check the user ID in effect for access rights to a resource.
- Authenticate a user ID and password (JDK 1.1.8).
- Check if a user ID is a member of a group (JDK 1.1.8).
- Change a user's password (JDK 1.1.8).

7.2 Installation

The Java for OS/390 Security Services package comes as a jar file RACF.jar. Ensure that it is part of the CLASSPATH variable. If you have installed JDK 1.1.8 for OS/390 the file is placed in the JAVA_HOME/lib/ subdirectory, where JAVA_HOME denotes the directory of your JDK installation.

The Java for OS/390 Security Services package calls native methods which are implemented in the shared library libSecurityServices.so. It has to be placed in a directory that is part of the LIBPATH variable. With JDK 1.1.8 the shared library is located in JAVA_HOME/lib/mvs/native_threads.

7.3 The classes in detail

The com.ibm.os390.security package consists of one interface and five classes.

7.3.1 PlatformAccesLevel

This interface is a placeholder for named constants used by `accessLevel` parameter of methods in `PlatformAccessControl` class.

7.3.2 PlatformReturned

This class is a helper used by the `PlatformAccessControl` and `PlatformUser` classes. It provides an output structure which is filled with various error codes

and values by the called OS/390 security service, such as an access control check.

The “meat” of the Java for OS/390 Security Services consists of four additional classes, each of them representing a platform entity of OS/390. Methods of each class represent an API to manipulate this platform entity. They are wrappers of OS/390 UNIX services which are in turn handled by a Security Server for OS/390 that implements SAF interfaces (such as RACF).

Note: These methods come as static methods intended to be invoked without the need to instantiate an object of the class.

7.3.3 PlatformSecurityServer

This class represents the OS/390 Security Server (such as RACF). Methods of this class are called to confirm the Security Server or a specific class of resources is active, for example:

```
if (PlatformSecurityServer.isActive())
  System.out.println("Security Server is active.");
else
  return;
```

7.3.4 PlatformAccessControl

This class represents the access control function of the OS/390 Security Server. It can be used to check the permission of the current user to a specific resource. If the current thread has a security context, the thread user ID is checked in an access control check. If not, the user ID associated with the current process is checked. For example, to check if the current user has READ acess to the resource named BPX.SERVER of resource type FACILITY, you would code:

```
PlatformReturned pr  =
   PlatformAccessControl.checkPermission("FACILITY" ,
                                         "BPX.SERVER" ,
                                      PlatformAccessLevel.READ ) ;
```

If the user is authorized for the specific resource, `checkPermission()` returns null; otherwise, `pr` is filled with error codes and messages.

7.3.5 PlatformThread

This class wraps OS/390 UNIX thread-level functions. It represents a UNIX thread that is mapped onto an OS/390 task (TCB). The method `getUserName` is useful to extract the user associated with the current thread:

```
String currentUser = PlatformThread.getUserName();
```

Note: At the time of writing, `PlatformThread.getUserName` was the only implemented method. In future releases of this package, `PlatformUser` will provide methods that allow you to switch the context of the current thread to a specific user. This means that a Java application will be able to support multiple threads, each running under its own authority! For example, a Java server application could reuse already granted permissions in a DB2 database, since every client accesses DB2 in its own security context.

7.3.6 PlatformUser

This class represents an OS/390 user ID which is the basis for OS/390 platform authentication and access control. The methods described in the following are not included in the first version of the Java for OS/390 Security Services.

The `authenticate` method validates a user ID and a password:

```
PlatformReturned pr = PlatformUser.authenticate( "anyUser","Password");

if (pr == null)
{
   System.out.println("Password check was successful");
}
else
{
    System.out.println("An error ocurred ...");
    System.out.println( "success: " + pr.success +
                        "\nerrno: " + pr.errno +
                        "\nerrno2: " + pr.errno2 +
                        "\nerrnoMsg: " + pr.errnoMsg );
}
```

The `isUserInGroup` method checks if a user is a member of a specific user group. Make sure to enter the user and the group name in uppercase:

```
boolean isMember = PlatformUser.isUserInGroup("ANYUSER","ANYGROUP");
if (isMember)
{
   System.out.println("ANYUSER is a member of ANYGROUP");
}
else
{
   System.out.println("ANYUSER is not a member of ANYGROUP");
}
```

The `changePassword` method is used to change a user's password. The user ID and the current password are validated before the password is changed to the new value:

```
PlatformReturned pr = PlatformUser.changePassword("anyUser",
                                                  "oldPwd",
                                                  "newPwd");
if (pr == null)
{
   System.out.println("Password change successful.\n");
}
else
{
   System.out.println("An error ocurred ...");
   System.out.println( "success: " + pr.success +
                    . . .
}
```

Important

The methods of `PlatformUser` and `PlatformAccessControl` wrap strongly protected OS/390 UNIX services. A program that calls these services, in our case the Java Virtual Machine, needs to be explicitly authorized to do so. This requires that the files in the `mvs/native_threads` directories in the `JAVA_HOME/bin` and `JAVA_HOME/lib` directories have their *program controlled attribute bits* set. They do not have to be APF-authorized, however. Ask your system administrator to make the appropriate changes, if you do not have the authority to do so.

Chapter 8. DB2 Version 6 external security enhancements

This chapter introduces the enhancements made to the DB2 External Security Module IRR@XACS, as supplied in SYS1.SAMPLIB, in support of DB2 version 6.

The DB2 External Security Module was first introduced at the DB2 Version 5 level, and allowed customers to use an external security manager (for example, RACF) for the protection of DB2 objects. This support is described in the redbook, *Ready for e-Business; OS/390 Security Server Enhancements*, SG24-5158.

DB2 Version 6 introduces new objects that can be protected using an external security manager (for example, RACF). It also introduces other security enhancements that we discuss in the next sections.

The enhancements to the DB2 External Security Module IRR@XACS are provided by APAR OW38710. OS/390 Security Server Version 2 Release 8 has these enhancements already incorporated.

DB2 Version 6 introduces the following new objects:

- User Defined Distinct Types
- User Defined Functions
- Stored Procedures
- Schemas

These new objects have new privileges to go with them, and they can now be protected using RACF profiles for the new objectclasses, which we discuss in 8.3, “New objectclasses and profiles” on page 243.

DB2 Version 6 introduces a so-called “trigger” level privilege on a DB2 table, which we discuss in 8.4, “Trigger privilege protection” on page 246.

DB2 Version 6 also introduces changes to the handling of ownership of certain objects, which we discuss in 8.5, “Security impact of ownership changes” on page 246.

8.1 IBM Class Descriptor Table enhancements

The IBM-supplied RACF Class Descriptor Table (CDT) has been updated to add new classes for the protection of the new objects. These new classes are:

MDSNUT for the protection of User Defined Distinct Types objects

MDSNUF for the protection of User Defined Function object

MDSNSP for the protection of Stored Procedure objects

MDSNSC for the protection of Schema objects

Note: The member class names as supplied in the IBM CDT also have a grouping class, GDSNUT for grouping User Defined Distinct Type objects together.

Note: The class names represent the usage of the DB2 External Security Module in *multi-subsystem scope*; for more information on this topic see the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.2 Installing the RACF/DB2 External Security Module

The OS/390 Security Server provides the RACF/DB2 External Security Module as an assembler source module. It resides in the IRR@XACS member of SYS1.SAMPLIB. Before you can use RACF with DB2 objects and authorities, you need to install the RACF/DB2 External Security Module using the following procedures:

1. Copy IRR@XACS to a private library, using DSNX@XAC as the member name.
2. Customize the RACF/DB2 External Security Module.

   Note: This step is optional and is necessary only if you want to modify the customization options from their default values.
3. Define classes for the RACF/DB2 External Security Module.

   Note: This step is optional and is necessary only when you have modified the customization options from their default values in the previous step.
4. When the previous steps have completed, the security administrator should define the appropriate profiles and activate the necessary classes.
5. Assemble and link-edit the RACF/DB2 External Security Module.

The RACF/DB2 External Security Module DSNX@XAC has to be placed in the hlq.SDSNEXIT load library. The next time the DB2 subsystem is restarted, the exit becomes active and issues the following messages, as shown in Figure 307.

```
S DB2UDBM1
$HASP100 DB2UDBM1 ON STCINRDR
IEF695I START DB2UDBM1 WITH JOBNAME DB2UDBM1 IS ASSIGNED TO USER STC
   , GROUP SYS1
$HASP373 DB2UDBM1 STARTED
IEF403I DB2UDBM1 - STARTED - TIME=12.45.10
IRR908I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB2U HAS 1
        A MODULE VERSION OF OW38710  AND A MODULE LENGTH OF 00004ED0.
IRR909I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB2U 902 2
        IS USING OPTIONS: &CLASSOPT=2
                          &CLASSNMT=DSN
                          &CHAROPT=1
                          &PCELLCT=50
                          &SCELLCT=50
IRR910I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB2U 903 3
        INITIATED RACLIST FOR CLASSES:
         MDSNDB   MDSNPK   MDSNPN   MDSNBP   MDSNCL
         MDSNTS   MDSNSG   MDSNTB   MDSNSM   MDSNSC
         MDSNUT   MDSNUF   MDSNSP   DSNADM
IRR911I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB2U 904 4
        SUCCESSFULLY RACLISTED CLASSES:
         MDSNDB   MDSNPK   MDSNPN   MDSNBP   MDSNCL
         MDSNTS   MDSNSG   MDSNTB   MDSNSM   MDSNSC
         MDSNUT   MDSNUF   MDSNSP   DSNADM
```

Figure 307. DB2 startup messages indicating usage of RACF/DB2 interface

1 IRR908I - indicates the version (PTF level) of the RACF/DB2 external security module and the size of the module.

2 IRR909I - indicates the RACF/DB2 options in effect.

3 IRR910I - indicates the RACF DB2 classes being targeted for a RACLIST.

4 IRR911I - indicates a successful RACLIST of the RACF DB2 classes.

If you are migrating from a previous release you have profiles already defined in the RACF DB2 classes; when it is your first attempt to use these classes, profiles need to be defined.

Note: When no profile is defined, security checking reverts to DB2 internal security checking.

The next sections discuss the new classes and profiles.

8.3 New objectclasses and profiles

As previously stated, each new DB2 object has its own set of classes, a so-called member and grouping class.

8.3.1 MDSNUT

The MDSNUT class protects DB2 User Defined Distinct Type objects. The construction of a RACF profile in this class is:

```
subsystem.schema-name.type-name.USAGE

subsystem indicates the DB2 subsystem name when running in Multi Subsystem
Scope
```

The `USAGE` privilege is the only privilege that can be protected using RACF profiles in the MDSNUT class. An example profile might look like:

```
RDEF MDSNUT DB2P.MYSCHEM.MYTYPE.USAGE
PERMIT DB2P.MYSCHEM.MYTYPE.USAGE CLASS(MDSNUT) ID(GRAAFF) ACC(READ)
```

For a complete discussion of access checking for DB2 user defined distinct types, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.3.2 MDSNUF

The MDSNUF class protects DB2 User Defined Function objects. The construction of a RACF profile in this class is:

```
subsystem.schema-name.function-name.privilege-name

subsystem indicates the DB2 subsystem name when running in Multi Subsystem
Scope

Schema indicates the schema name

privilege-name indicates the privilege to be permitted
```

The privileges are:

- EXECUTE
- DISPLAY
- START
- STOP

Note: The START and STOP privileges are DB2 operator privileges. They are not GRANTable.

The EXECUTE and DISPLAY privileges can be protected using RACF profiles in the MDSNUF class. An example profile might look like:

```
RDEF MDSNUF DB2P.MYSCHEM.MYFUNC.EXECUTE
PERMIT DB2P.MYSCHEM.MYFUNC.EXECUTE CLASS(MDSNUF) ID(GRAAFF) ACC(READ)

RDEF MDSNUF DB2P.MYSCHEM.MYFUNC.DISPLAY
PERMIT DB2P.MYSCHEM.MYFUNC.DISPLAY CLASS(MDSNUF) ID(GRAAFF) ACC(READ)
```

The START and STOP privileges are checked against the following RACF profiles in the DSNADM class:

- subsystem.SYSOPR
- subsystem.SYSCTRL
- subsystem.SYSADM

For a complete discussion of access checking for DB2 user-defined functions, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.3.3 MDSNSP

The MDSNSP class protects so-called DB2 Stored Procedure objects. The construction of a RACF profile in this class is:

```
subsystem.schema-name.proc-name.privilege-name

subsystem indicates the DB2 subsystem name when running in Multi Subsystem
Scope
```

The privileges are:

- EXECUTE
- DISPLAY
- START
- STOP

Note: The START and STOP privileges are DB2 operator privileges. They are not GRANTable.

The EXECUTE and DISPLAY privileges can be protected using RACF profiles in the MDSNSP class. An example profile might look like:

```
RDEF MDSNUF DB2P.MYSCHEM.MYPROC.EXECUTE
PERMIT DB2P.MYSCHEM.MYPROC.EXECUTE CLASS(MDSNUF) ID(GRAAFF) ACC(READ)

RDEF MDSNUF DB2P.MYSCHEM.MYPROC.DISPLAY
PERMIT DB2P.MYSCHEM.MYPROC.DISPLAY CLASS(MDSNUF) ID(GRAAFF) ACC(READ)
```

The START and STOP privileges are checked against the following RACF profiles in the DSNADM class:

- subsystem.SYSOPR
- subsystem.SYSCTRL
- subsystem.SYSADM

For a complete discussion of access checking for DB2 stored procedures, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.3.4 MDSNSC

The MDSNSC class protects DB2 Schema objects. The construction of a RACF profile in this class is:

```
subsystem.schema-name.object-name.privilege-name

subsystem indicates the DB2 subsystem name when running in Multi Subsystem
Scope
```

The schema privileges are:

- ALTERIN
- COMMENT ON
- CREATEIN
- DROPIN
- CHANGE NAME QUALIFIER

Note: The privileges COMMENT ON and CHANGE NAME QUALIFIER are *not* explicitly GRANTable.

These privileges can be protected using RACF profiles in the MDSNSC class. An example profile might look like:

```
RDEF MDSNUF DB2P.MYSCHEM.MYOBJ.ALTERIN
PERMIT DB2P.MYSCHEM.MYOBJ.ALTERIN CLASS(MDSNUF) ID(GRAAFF) ACC(READ)

RDEF MDSNUF DB2P.MYSCHEM.CREATEIN
PERMIT DB2P.MYSCHEM.CREATEIN CLASS(MDSNUF) ID(GRAAFF) ACC(READ)

RDEF MDSNUF DB2P.MYSCHEM.MYOBJ.DROPIN
ERMIT DB2P.MYSCHEM.MYOBJ.DROPIN CLASS(MDSNUF) ID(GRAAFF) ACC(READ)
```

The CHANGE NAME QUALIFIER privilege is checked against the following RACF profiles in the DSNADM class:

- subsystem.SYSCTRL
- subsystem.SYSADM

For a complete discussion of access checking for DB2 schema, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.4 Trigger privilege protection

DB2 Version 6 introduces a new privilege called “trigger” that you can use on tables. Triggers are sets of SQL statements that execute when a certain event occurs in a DB2 table. The event can be an insert, update, or delete operation.

The DB2 External Security Module has been enhanced to support triggers on tables. The RACF profile in the MDSNTB class now allows for a suffix of trigger:

```
DB2-subsystem.table-owner.table-name.TRIGGER
```

For a complete discussion of access checking for DB2 triggers on table resources, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.5 Security impact of ownership changes

When the user is the owner of a DB2 object, the user may have some implicit privileges, but not all privileges associated with the object. The RACF/DB2 External Security Module supports certain implicit privileges of ownership for DB2 objects and associated privileges, as shown in Table 9.

Table 9. Ownership and implicit privileges overview

DB2 object	Owner field	Implicit privileges
Package	XAPLREL1	BINDAUT and COPYAUT
Plan	XAPLOWNQ	BINDAUT
Schema	XAPLREL1	ALTERIN, COMMENT ON, and DROPIN
Stored procedure*	XAPLREL1	DISPLAY, START, and STOP
Table	XAPLOWNQ	All privileges except CRTSYAUT, DRPSYAUT, CRTVUAUT, and CNVRTAUT
User-defined distinct type	XAPLREL1	USAGE
User-defined function*	XAPLREL1	DISPLAY, START, and STOP

To check authorization for the privileges associated with implicit ownership, the RACF/DB2 External Security Module first checks to see if the ACEEUSER matches the value passed in the owner field. If these two fields are equal, the RACF/DB2 External Security Module authorizes access and returns a return code 0 in EXPLRC1 and reason code 13 in EXPLRC2. If this check fails, a check is made to see if XAPLUCHK equals the owner field (“does the current SQL ID equal the owner of the object?”). If these two fields are equal, access is allowed. If this check fails, profile checking will occur.

For a complete discussion of access checking for DB2 resources and ownership, see Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

8.6 Matching schema names

Certain privileges associated with schema objects (such as user-defined functions, user-defined distinct types, and stored procedures) can be granted if the user matches the schema name. The schema name is a short SQL identifier used as a qualifier in the name of schema objects and creates a logical grouping of these objects. It is often, but not always, a DB2 authorization ID.

For applicable privileges, the RACF/DB2 External Security Module will look for a match on schema name before checking RACF profiles. The RACF/DB2 external security module first checks to see if the ACEEUSER matches the schema name in field XAPLOWNQ. If these two fields are equal, the RACF/DB2 external security module authorizes access and returns a return code 0 in EXPLRC1 and reason code 14 in EXPLRC2. If the ACEEUSER equals XAPLOWNQ, no further checking occurs. If this check fails, profile checking will occur.

See Appendix D of the *OS/390 Security Server Security Administrator's Guide*, SC28-1915 for a complete discussion.

Part 3. IBM Communication Server for OS/390 security enhancements

Chapter 9. OS/390 Firewall Technologies enhancements

This chapter describes the enhancements made to the OS/390 Firewall Technologies.

For an introduction to the OS/390 Firewall Technologies, see *Stay Cool on OS/390: Installing Firewall Technologies,* SG24-2046.

9.1 Administration enhancements

OS/390 Security Server Release 7 introduces a Java-based graphical user interface (GUI) for administration of the OS/390 Firewall Technology. Before this release the OS/390 Firewall Technologies were administered by UNIX line commands. The GUI will be a welcome addition to the tools to administer the OS/390 Firewall Technologies. You are still able to use the UNIX line commands though, if you prefer.

The Java-based GUI is called the configuration client. The configuration client allows an administrator to perform remote configuration and administration. The configuration client runs on AIX, Windows 95 or Windows NT.

To ensure confidentiality and integrity, the configuration connection between the GUI and the new configuration server is authenticated using RACF and all data is protected by Secure Sockets Layer (SSL).

The configuration server is the configuration client's graphical user interface (GUI) to the Firewall. This server processes requests from the AIX, Windows 95 or Windows NT configuration clients, as shown in Figure 308. Once it is set up, it is considered part of the Firewall machine. The configuration client allows an authorized user to log on to the configuration server and perform specific configuration functions.

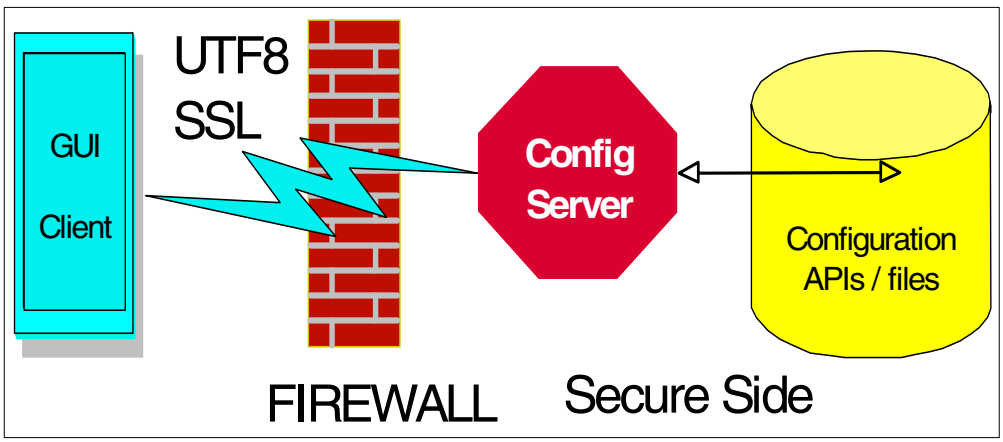


Figure 308. OS/390 Firewall Technologies configuration client and server overview

9.1.1 Installation of the configuration client

The configuration client code for Windows 95 and Windows NT is located in /usr/lpp/fw/bin/fwtech.zip.

The installation of the configuration client consists of the following steps:

1. Download the FWTECH.ZIP file to your workstation.
2. Unzip the FWTECH.ZIP file.
3. Run the SETUP utility.
4. Set up IP Filter rules to allow remote administration.
5. Set up SSL for the Configuration Server to use.
6. Authorize the client to use the Configuration Client

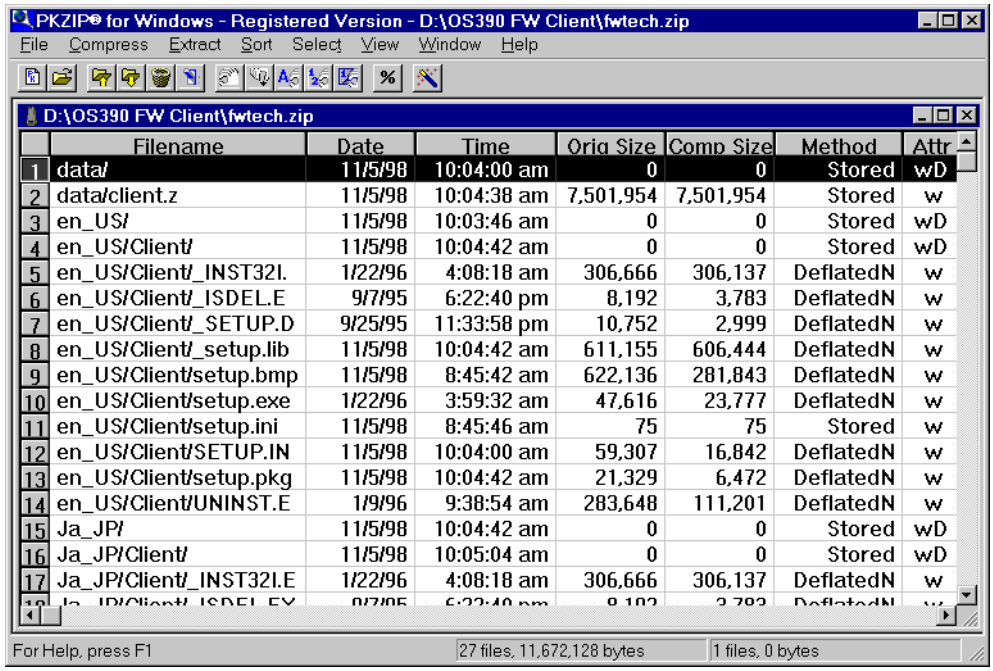
9.1.1.1 Download FWTECH.ZIP

Transmit this file to the Windows machine using FTP or a similar facility. If FTP is used to transmit the file, the `BIN` option must be set before the `GET` or `PUT` is issued.

9.1.1.2 Unzip the FWTECH.ZIP

To unzip the FWTECH.ZIP file, use any popular zip tool that can handle long file names, like Winzip32 at `www.winzip.com`

In our example the FWTECH.ZIP file was downloaded to the *OS/390 FW Client* directory. We selected this directory and opened the archive file, FWTECH.ZIP, as shown in Figure 309.



	Filename	Date	Time	Orig Size	Comp Size	Method	Attr
1	data/	11/5/98	10:04:00 am	0	0	Stored	wD
2	data/client.z	11/5/98	10:04:38 am	7,501,954	7,501,954	Stored	w
3	en_US/	11/5/98	10:03:46 am	0	0	Stored	wD
4	en_US/Client/	11/5/98	10:04:42 am	0	0	Stored	wD
5	en_US/Client/_INST32I.	1/22/96	4:08:18 am	306,666	306,137	DeflatedN	w
6	en_US/Client/_ISDEL.E	9/7/95	6:22:40 pm	8,192	3,783	DeflatedN	w
7	en_US/Client/_SETUP.D	9/25/95	11:33:58 pm	10,752	2,999	DeflatedN	w
8	en_US/Client/_setup.lib	11/5/98	10:04:42 am	611,155	606,444	DeflatedN	w
9	en_US/Client/setup.bmp	11/5/98	8:45:42 am	622,136	281,843	DeflatedN	w
10	en_US/Client/setup.exe	1/22/96	3:59:32 am	47,616	23,777	DeflatedN	w
11	en_US/Client/setup.ini	11/5/98	8:45:46 am	75	75	Stored	w
12	en_US/Client/SETUP.IN	11/5/98	10:04:00 am	59,307	16,842	DeflatedN	w
13	en_US/Client/setup.pkg	11/5/98	10:04:42 am	21,329	6,472	DeflatedN	w
14	en_US/Client/UNINST.E	1/9/96	9:38:54 am	283,648	111,201	DeflatedN	w
15	Ja_JP/	11/5/98	10:04:42 am	0	0	Stored	wD
16	Ja_JP/Client/	11/5/98	10:05:04 am	0	0	Stored	wD
17	Ja_JP/Client/_INST32I.E	1/22/96	4:08:18 am	306,666	306,137	DeflatedN	w

Figure 309. PKZIP Example screen

Select **Extract** from the main menu, which returns the pop-up window shown in Figure 310 on page 253.

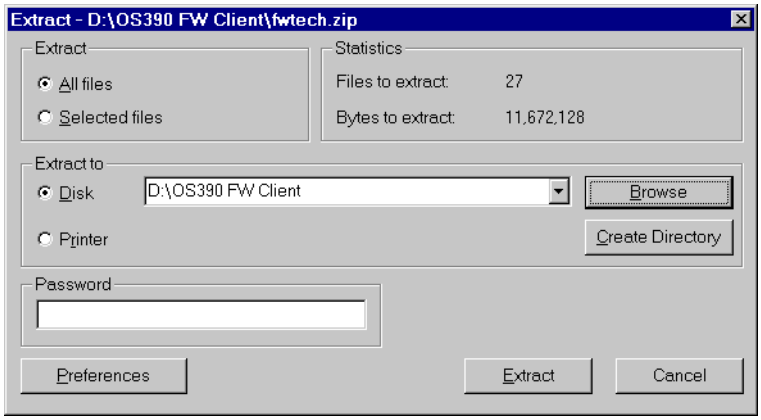


Figure 310. PKZIP prompt for extraction of files

After selecting the directory you want to extract the files to, click **Extract**. The files are extracted to the specified directory. When all files are extracted click **Done**, as shown in Figure 311.

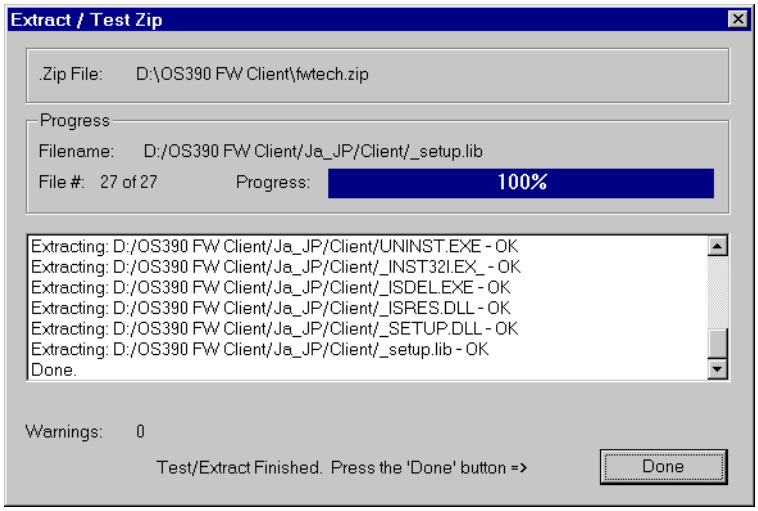


Figure 311. PKZIP extract done window

9.1.1.3 Run the SETUP utility

Now that all files are extracted, continue with the actual installation. Execute the file `X:\OS390 FW Client\en_US\Client\setup.exe`, where `X` is the drive we installed the code from. We used the Windows NT Run menu option to execute the `setup.exe` program, as shown in Figure 312 on page 254.

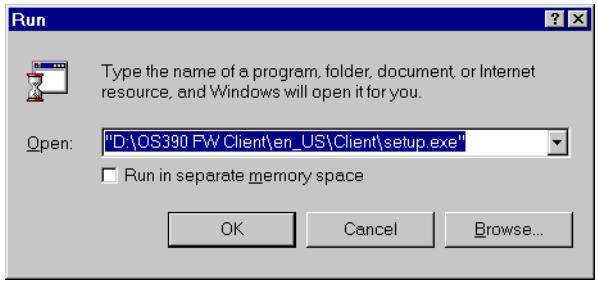


Figure 312. Windows NT Run Menu

Attention

This information relates to the US version of the Configuration Client. There is also code supplied for a Japanese version of the Configuration Client.

Click **OK** on the Windows NT Run window; the screen shown in Figure 313 is presented.

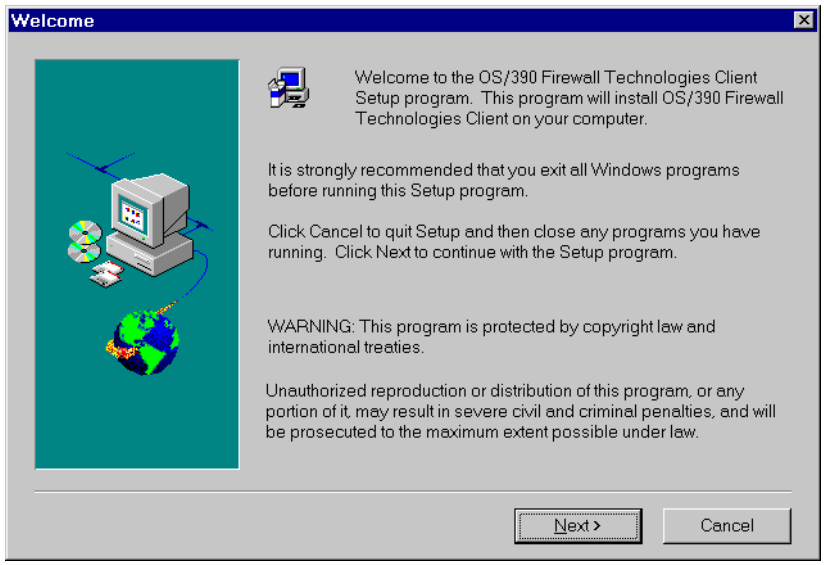


Figure 313. OS/390 Firewall Technologies Client Setup welcome

Click **Next** to continue the installation. The next window is displayed to confirm installation of the OS/390 Firewall Technologies Configuration Client, as shown in Figure 314 on page 255.

Figure 314. OS/390 Firewall Technologies Configuration Client information window (1)

Click **OK**; the next window reminds you that you must use a frame-enabled browser, like Internet Explorer 4.0 or Netscape Navigator 4.0, as shown in Figure 315.

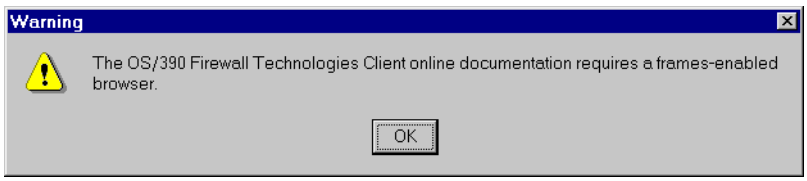


Figure 315. OS/390 Firewall Technologies Configuration Client information window (2)

Click **OK** to continue the installation. The next window asks for the location where you want the configuration client to be installed, as shown in Figure 316.

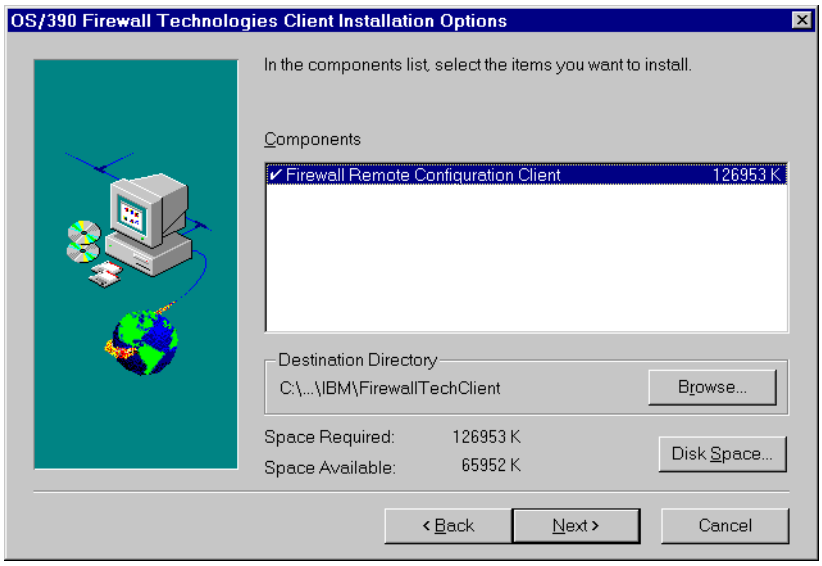


Figure 316. OS/390 Firewall Technology Configuration Client installation window

The default location for configuration client software is C:\Program Files\IBM\FirewallTechClient. You can change this location to any directory or drive you choose by clicking the **Browse** button. We changed our specification to the D drive rather then the C drive. Continued the installation by clicking **Next**.

The next window, shown in Figure 317 on page 256, reconfirms your choices.

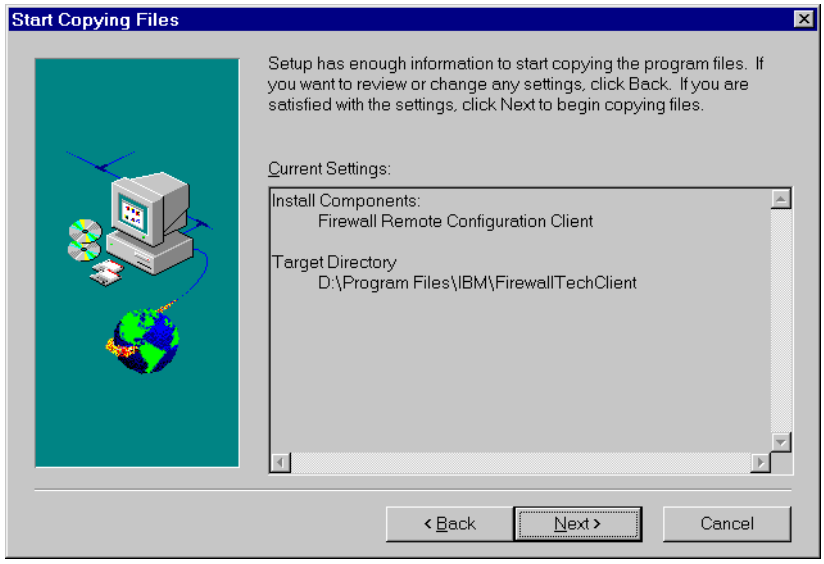


Figure 317. OS/390 Firewall Technologies installation confirmation window

Click **Next** to continue the installation; the files are then copied to the directory specified and a menu item will be added to the start menu under the programs option, called the *OS/390 Firewall Technologies Client.* When the installation ends, an information window will be presented, as shown in Figure 318, to confirm the completion of the installation.

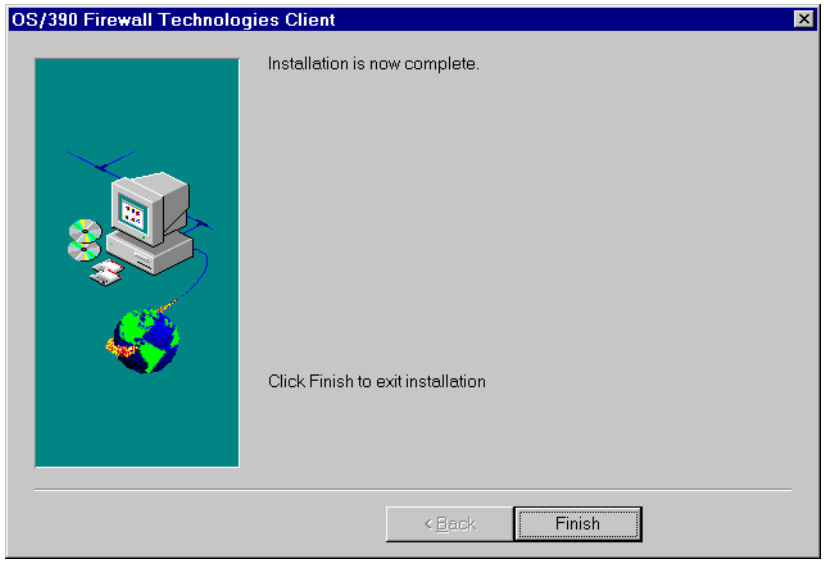


Figure 318. OS/390 Firewall Technologies Configuration Client installation complete

This completes the installation of the OS/390 Firewall Technologies configuration client.

9.1.1.4 Set up IP Filter rules to allow remote administration

The configuration client talks to the configuration server to make changes to the firewall configuration. We have to allow the configuration client to make these requests to the configuration server on port 1014.

Before we can really use the configuration client, we have to define two filter rules allowing traffic to come through port 1014. Our OS/390 system has two interfaces, a secure interface at address 9.12.2.21 and an unsecure interface at address 9.12.14.247. Our system configuration is shown in Figure 319.

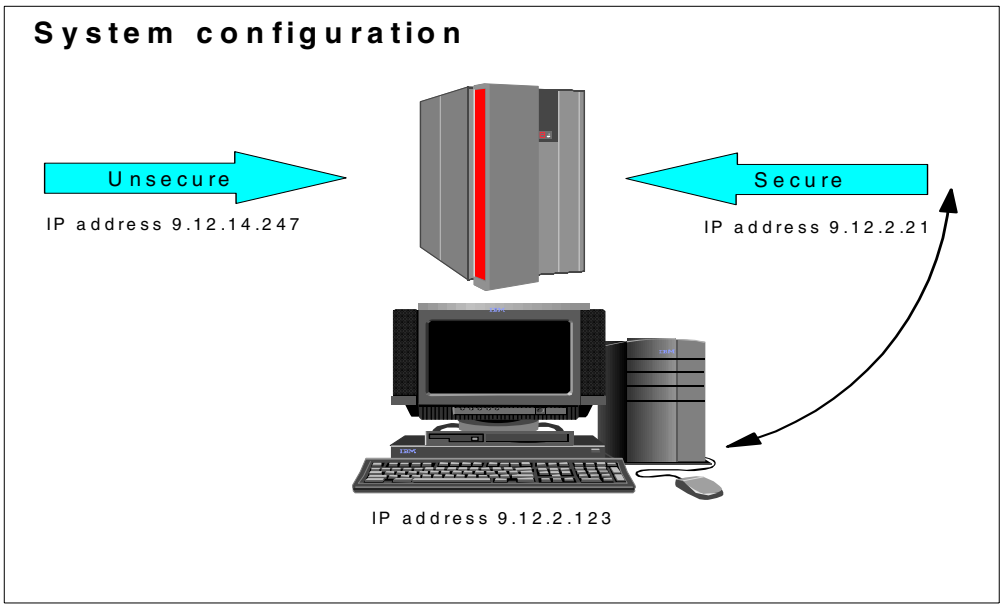


Figure 319. System configuration

The following steps show how to customize a set of filter rules to permit specific configuration traffic to the secure interface from our internal network. You have to go to the UNIX System Services shell environment using the authorized user ID defined previously and issue the commands shown in Figure 320.

```
GRAAFF @ SC57:/u/graaff>fwfrule cmd=add type=permit name='remote admin in
secure' protocol=tcp srcopcode=gt srcport=1023 destopcode=eq \
> destport=1014 interface=secure routing=local direction=inbound log=yes 1
GRAAFF:/u/graaff>fwfrule cmd=list id=510 2
               id = 510
             type = permit
             name = remote admin in secure
             desc =
         protocol = tcp
        srcopcode = gt
          srcport = 1023
       destopcode = eq
         destport = 1014
        interface = secure
          routing = local
        direction = inbound
              log = yes
           tunnel =
         fragment =

GRAAFF @ SC57:/u/graaff>
```

Figure 320. Defining firewall configuration client rules for inbound traffic

Figure 320 notes:

1 `fwrule` command creates a rule that allows any kind of traffic over the secure interface to the local host.

2 List the rule you have created to get the ID. It will be used in the `fwservice` command.

The filter rule defined previously allowed *remote administration* traffic coming in; now we also need to define the filter rule that allows the *response* traffic to go back to the client. The command to perform that operation is shown in Figure 321.

```
GRAAFF @ SC57:/u/graaff>fwfrule cmd=add type=permit name='remote admin out
secure' protocol=tcp srcopcode=eq srcport=1014 destopcode=gt \
> destport=1023 interface=secure routing=local direction=outbound log=yes
GRAAFF @ SC57:/fwlog>fwfrule cmd=list id=511
               id = 511
             type = permit
             name = remote admin out secure
             desc =
         protocol = tcp
        srcopcode = eq
          srcport = 1014
       destopcode = gt
         destport = 1023
        interface = secure
          routing = local
        direction = outbound
              log = yes
           tunnel =
         fragment =

GRAAFF @ SC57:/fwlog>
```

Figure 321. Defining firewall configuration client rules for outbound traffic

Note: The `log=yes` parameter on `FWFRULE` is not required, but we wanted logging to be performed.

```
GRAAFF @ SC57:/fwlog>fwservice cmd=create name='Remote Administration' desc='All
ow Remote Administration' rulelist=510/f,511/b log=yes     3
GRAAFF @ SC57:/fwlog>fwservice cmd=list name='Remote Administration' 4
               id = 503
             name = Remote Administration
             desc = Allow Remote Administration
         rulelist = 510/f,511/b
              log = yes
         fragment =
           tunnel =
             time =
            month =
              day =
          weekday =
       timefilter =

GRAAFF @ SC57:/fwlog>
```

Figure 322. Defining firewall configuration client services

Figure 322 notes:

3 Create a service that contains the rule created previously.

4 List the service you have created to get the ID. It will be used in the `fwconns` command.

We already had definitions made for network objects that represented our network and a *connection* between the networks. The definitions are shown in Figure 323.

```
GRAAFF @ SC57:/fwlog>fwnwobj cmd=list id=503
               id = 503
             type = VPN
             name = secure
             desc =
             addr = 9.12.2.21 6
             mask = 255.255.255.255
        startaddr =
          endaddr =

GRAAFF @ SC57:/fwlog>fwnwobj cmd=list id=505
               id = 505
             type = Network
             name = Everybody
             desc = everybody coming in
             addr = 0.0.0.0 5
             mask = 0.0.0.0
        startaddr =
          endaddr =
GRAAFF @ SC57:/fwlog>fwconns cmd=list id=502 7
               id = 502
             name = Remote Admin from anywhere
             desc =
           source = 505
      destination = 501
      servicelist = 503
        sockslist =
```

Figure 323. Firewall configuration client Network and Connection objects

Figure 323 notes:

5 A Network object that represents your intranet. In this case, we used the everybody.

6 A Network object that represents the host where the firewall server is running.

7 A connection associating the two Network objects with the service.

After completing these steps, update the filter rules to allow the new rules to take effect. To update the filter rules, use the `FWFILTER CMD=UPDATE` command, as show in Figure 324 on page 260.

```
GRAAFF @ SC57:/fwlog>fwfilter cmd=update    8
ICAC1577i Processing firewall TCP/IP stack ETCPIP:


Filter support (level 2.80) initialized at 16:55:10 on Jan-31-2000


ICAC1531w Unable to inform the sock daemon to refresh configuration data.    9
GRAAFF @ SC57:/fwlog>
```

Figure 324. Updating the firewall filter rules

Figure 324 notes:

8 Update the filter rules.

9 You will receive this warning message unless you have the SOCKS server running. If you do not need it, ignore this message.

All other configurations related to the IKE function will be made using the firewall configuration client GUI. For more information about the firewall command see *OS/390 Firewall Technologies Guide and Reference*, SC24-5835.

9.1.1.5 SSL setup for the firewall configuration client

The firewall configuration server uses the Secure Sockets Layer (SSL) protocol of OS/390 for communication between the GUI clients and the server.

The administrator must run the GSKKYMAN utility to create a new key database, if one was not already created. The administrator needs to use the “Create a self-signed certificate” option to create and store a Self-Signed Certificate Version 3, then use the “Store encrypted database password” option. See the*OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference*, SC24-5877 for information on how to use the GSKKYMAN utility.

GSKKYMAN uses the DLLs that are installed with System SSL and must have access to these at run time. GSKKYMAN must also have access to the message catalogs. The library /bin includes a symbolic link to /usr/lpp/gskssl/bin/gskkyman, therefore, if your PATH environment variable contains this directory, you will find the GSKKYMAN utility. If your PATH environment variable does not contain this directory, add /usr/lpp/gskssl/bin to your PATH using the following:

```
PATH=$PATH:/usr/lpp/gskssl/bin
```

The libraries /usr/lib/nls/msg/C and /usr/lib/nls/msg/En_US.IBM-1047 (and /usr/lib/nls/msg/Ja_JP for JCPT272 installations) include symbolic links to the message catalogs for GSKKYMAN. If they do not include these links, add /usr/lpp/gskssl/lib/nls/msg to your NLSPATH using the following command:

```
export NLSPATH=$NLSPATH:/usr/lpp/gskssl/lib/nls/msg/%L/%N
```

This setting assumes that your environment has the LANG environment variable set to En_US.IBM-1047 (Ja_JP for JCPT272 installations that expect Japanese messages and prompts). If LANG is not set properly, set the NLSPATH environment variable using the following command:

```
export NLSPATH=$NLSPATH:/usr/lpp/gskssl/lib/nls/msg/En_US.IBM-1047/%N
```

Or for JCPT272 installations that expect Japanese messages and prompts:

```
export NLSPATH=$NLSPATH:/usr/lpp/gskssl/lib/nls/msg/Ja_JP/%N
```

The DLLs for System SSL are installed into a partitioned data set (PDS). These DLLs are not installed into the LINKLIB or LPALIB by default. To access these DLLs, if they have not been placed in LINKLIB or LPALIB, you must set the STEPLIB environment variable to find the DLLs. Consult your system programmer for the high-level qualifier of the System SSL PDS. In the following example, the high-level qualifier for the System SSL PDS is GSK. In the following command, replace the value to match your installation:

```
export STEPLIB=GSK.SGSKLOAD
```

Using the GSKKYMAN utility to create a key database for the configuration server:

```
GIANCA @ RA03:/u/gianca/firewall>gskkyman


            IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 1
Enter key database name or press ENTER for "key.kdb": fwcfgsrv.kdb
Enter password for the key database.......>
Enter password again for verification.....>
Should the password expire? (1 = yes, 0 = no) : 0

The database has been successfully created, do you want to continue to work with
 the database now? (1 = yes, 0 = no) : 1                              Enter your
```

Figure 325. GSKKYMAN utility: main menu

Enter option 1 to create a new kdb file and press Enter. Type the key database name and press Enter. We used fwcfgsrv.kdb. Enter the password for the key database file twice (press Enter after you type the password). Do not forget this password because each time you have to access this kdb file you will be prompted for this particular password. Type 0 and press Enter to not use an expired password. Type 1 and press Enter to continue to work with this database.

```
          Key database menu

Current key database is /u/gianca/firewall/os390/fwcfgsrv.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 5
Enter version number of the certificate to be created (1, 2, or 3):  3
Enter a label for this key.............> Firewall GUI Key
Select desired key size from the following options (512):
    1:    512
    2:    1024
Enter the number corresponding to the key size you want: 1
Enter certificate subject name fields in the following.
    Common Name (required).............> Firewall GUI Key
    Organization (required)............> IBM
    Organization Unit (optional).......> ITSO
    City/Locality (optional)...........> Cary
    State/Province (optional)..........> North Caroline
    Country Name (required 2 characters)..> US
Enter number of valid days for the certificate :  365
Do you want to set the key as the default in your key database? (1 = yes, 0 = no
): 1
Do you want to save the certificate to a file? (1=yes, 0=no): 0

Please wait while self-signed certificate is created...

Your request has completed successfully, exit gskkyman? (1=yes,0=no) : 0
```

Figure 326. GSKKYMAN: creating a self-signed certificate

Type `5` and press Enter to create a self-signed certificate. You can use another certificate importing a key into your database or creating a certificate request. For this case we used a self-signed certificate. Type `3` and press Enter to create a Version 3 certificate. The version number refers to the X.509 standard version number. Type a label for the key and press Enter. Type a common name and press Enter. Type an organization name and press Enter. All the other fields are optional. Then type the number of days this certificate should be valid and press Enter. We chose one year. Type `1` and press Enter to set this key as the default key in this database. Type `0` and press Enter to not save this certificate into a file. Type `0` and press Enter to return to the previous menu.

```
          Key database menu

Current key database is /u/gianca/firewall/os390/fwcfgsrv.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 11

The encrypted password has been stored in file /u/gianca/firewall/os390/fwcfgsrv
.sth

Yourrequesthascompletedsuccessfully,exitgskkyman?(1=yes,0=no)>:1
```

Figure 327. GSKKYMAN: storing an encrypted password

Type `11` and press Enter to store the encrypted database password in a stashed file. Finally, type `1` and press Enter to leave `GSKKYMAN`.

Now you have two files in the directory in which you were running `GSKKYMAN`: a kdb file and an sth file, whose file names in our example were fwcfgsrv.kdb and fwcfgsrv.sth respectively. The kdb file contains the keys and certificates you created and the sth file contains the database password. Copy these two files to the directory you specified in the fwdaemon daemonopts parameter for the CFGSRV server.

9.1.1.6 Using the configuration client

Start the GUI from the Windows start menu. You will see the following screen:

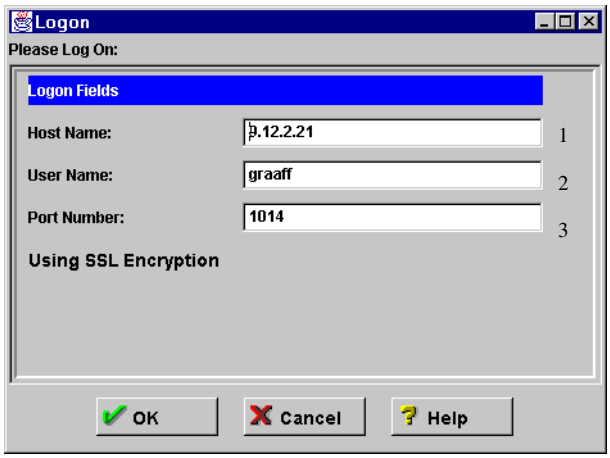


Figure 328. Firewall configuration client menu

Figure 328 notes:

1 This is the host IP address where the firewall is running.

2 The administrator user name you defined previously.

3 The port number you configured in the `FWDAEMON` command (-p parameter).

Note: Do not forget that this user must be permitted to the ICA.CFGSRV profile.

Click **OK** to continue. You will receive a password prompt:

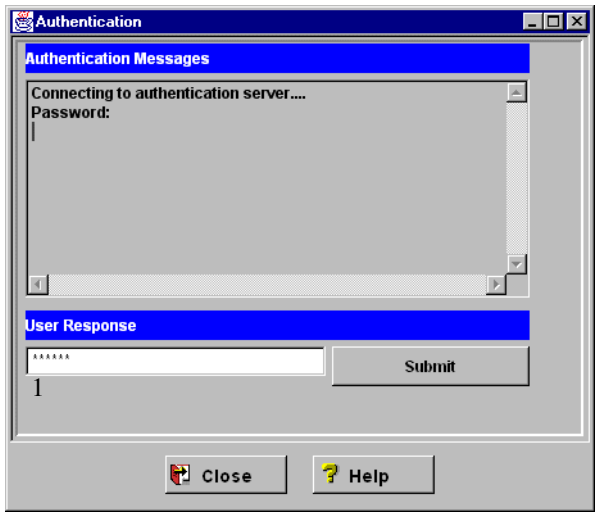


Figure 329. Firewall GUI password prompt

Figure 329 note:

1 This is the RACF password for the user ID.

Click **Submit** to continue. Now you will see the firewall configuration client main screen:

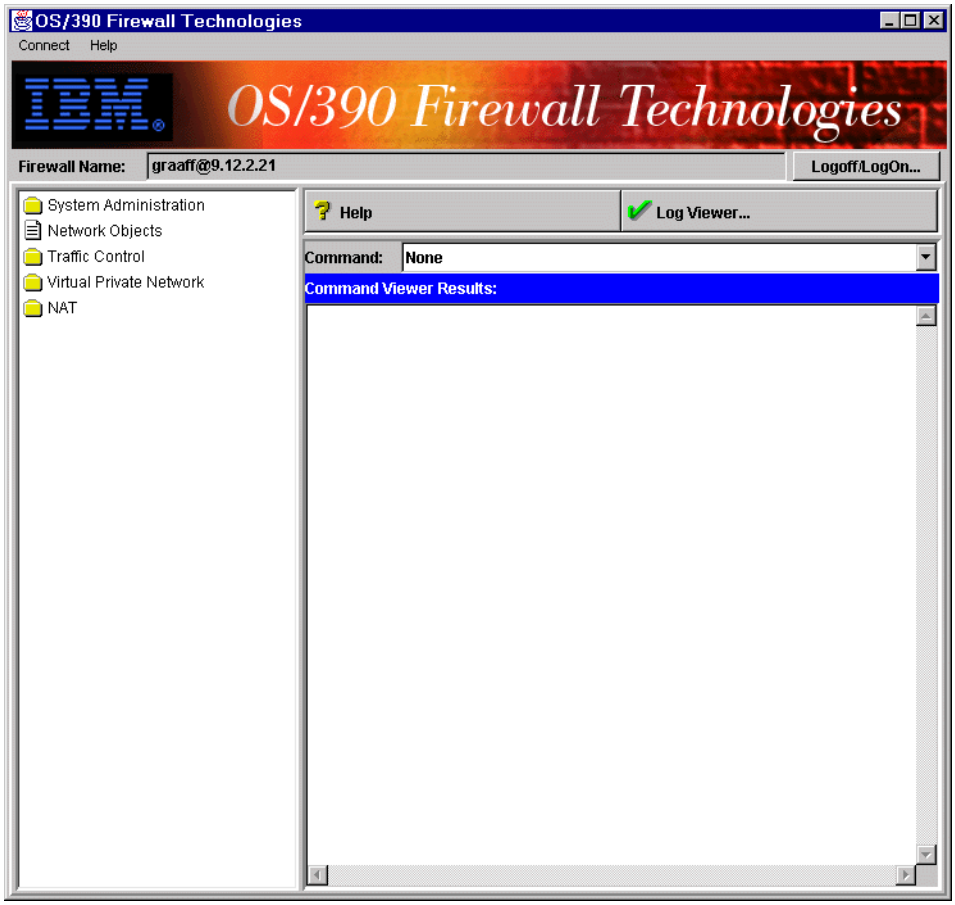


Figure 330. Firewall configuration client main screen

For more information about how to use the firewall configuration client see *OS/390 Firewall Technologies Guide and Reference*, SC24-5835.

9.2 IPSec enhancement

In OS/390 V2R8, we have the dynamic tunnels support based on the Internet Security Association and Key Management Protocol (ISAKMP) standards developed by the Internet Engineering Task Force (IETF). The dynamic tunnels provide a more reliable and secure connection than the manually configured tunnels because the key exchange is done automatically. The secure key negotiation and key refreshment is done in an industry-standard way.

The ISAKMP support is based on the following RFC standards:

- RFC 2407 - The Internet IP Domain of Interpretation for ISAKMP
- RFC 2408 - The Internet Security Association and Key Management Protocol
- RFC 2409 - The Internet Key Exchange

9.3 Firewall Technologies for OS/390

The OS/390 Firewall Technologies is a collection of programs that provide the firewall functions on OS/390, including support for virtual private networks (VPNs). The virtual private network further enhances the level of security of the system by providing a mechanism for data to be encrypted and/or authenticated between endpoints. The OS/390 Firewall Technologies provides the facilities to manually or dynamically define a virtual private network according to standards defined by IETF.

In essence, the OS/390 firewall consists of traditional firewall functions and support for VPNs. The OS/390 Firewall Technologies provides for:

- **IPSec, virtual private network, or tunneling**
  - Key management either manually or dynamically. Dynamic VPN support has been introduced by OS/390 V2R8. The Internet Key Exchange (IKE) authentication is done using either pre-shared keys or RSA Signatures.
  - S/390 hardware cryptographic facility will be used if available.
- **Application gateways (proxies)**
  - FTP proxy
- **Transparent gateways (SOCKS)**
  - SOCKS V4 server
- **Packet filtering**
  - Filter rules
- **Network Address Translation (NAT)**
  - Administrator-defined address mapping
  - Address translation in IP headers only
- **Logging**
  - Enhanced Syslog server
  - SMF records
- **Configuration and administration**
  - Compatible with AIX and Windows NT (command line and GUI).
  - Commands are valid in OE only, not from TSO.
  - Commands create intermediate files, which are used to update the online configuration.
  - An external security manager such as RACF is used to control authorization to maintain network security profiles. This is in line with general security concepts on the OS/390 servers.

These features may be used in any combination.

Although OS/390 Firewall Technologies provides various functions, in this book we focus only on the dynamic VPN functions. For more information on the other firewall functions on OS/390, see *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046 and *OS/390 Firewall Technologies Guide and Reference*, SC24-5835.

9.3.1 OS/390 Firewall Technologies enhancements

In OS/390 V2R7, the following enhancements were introduced by OS/390 Firewall Technologies:

- IP Security

  The IP security support has been upgraded to support the latest RFCs regarding the IPSec protocol (2401-2406 and 2410) in addition to the old RFCs. The encryption algorithms supported include Data Encryption Standard (DES) and Triple DES, as well as Commercial Data Masking Facility (CDMF). The algorithms used to perform authentication are Keyed Message Digest-5 and two versions of the Hashed-Based Message Authentication Code (HMAC), HMAC_SHA and HMAC_MD5.

- Configuration client GUI

  The product can now be administered through a Java-based interface that runs on AIX, Windows 95 and Windows NT platforms. It allows you to perform a remote configuration and administration in a secure way using the authentication provided by an external security manager, such as RACF, and Secure Sockets Layer (SSL) transport.

- Support of multiple stacks

  OS/390 Firewall Technologies now supports up to eight TCP/IP stacks.

The significant enhancement provided by OS/390 V2R8 in order to implement a secure network using the VPN function is:

- Dynamic VPN support

  OS/390 Firewall Technologies Version 2 Release 8 adds dynamic tunneling, which is support for the dynamic negotiation of keys and algorithms used to create IPSec Security Associations (SAs) between systems in a VPN. OS/390 Firewall Technologies has added a new component called ISAKMPD.

9.4 IPSec, virtual private network or tunneling

In this section, we provide a brief overview of the Security Architecture for the Internet Protocol (IPSec) and Internet Key Exchange (IKE) protocol. For a complete description of these two technologies, refer to the corresponding RFCs. Additional information is in *A Comprehensive Guide to Virtual Private Networks, Volume III: IBM Cross-Platform and Key Management Solutions*, SG24-5309 and *TCP/IP Tutorial and Technical Overview*, GG24-3376.

9.4.1 Internet security (IPSec)

IP Security enables secure communications over the Internet and within company networks by securing data traffic at the IP layer. This allows individual users or organizations to secure traffic for all applications, without having to make any modifications to the applications.

The IP Security (IPSec) Working Group of the Internet Engineering Task Force (IETF) has defined an open architecture and an open framework, known as *IPSec*. IPSec is called a framework because it provides a stable, long-lasting base for providing network layer security.

IPSec is a security protocol in the IP layer of TCP/IP that provides security services to ensure packet authentication, integrity, and confidentiality. IPSec is essentially an encapsulation protocol, namely one that defines the syntax and semantics of placing one packet inside another. IPSec protects your data traffic in three ways, using robust cryptographic techniques:

1. Authentication: the process by which the identity of a host or end point is verified
2. Integrity checking: the process of ensuring that no modifications were made to the data while in transit across the network
3. Encryption: the process to provide the data confidentiality using encryption algorithms that convert a message (plaintext) into gibberish (ciphertext) and back again

The principal IPSec protocols are:

- IP Authentication Header (AH), which provides data origin authentication and data integrity
- IP Encapsulating Security Payload (ESP), which provides data confidentiality, optional data origin authentication and data integrity

The protocol formats for AH and ESP are independent of the cryptographic algorithms used to perform the authentication and encryption. Once an IPSec security association is established between two systems, they can perform message encryption and message authentication. A security association is simply a set of items of information which, when shared between two communicating systems, enables the two systems to communicate in a desired way.

The OS/390 Firewall Technologies supports the original IPSec standards described in Request for Comments (RFCs) 1825 through 1829, as well as the latest standards described in RFCs 2401 through 2406 and 2410. The original standards continue to be supported to ensure backward compatibility.

Note that dynamic VPNs do not support doing IPSec with the old RFC headers and they do not support doing IPSec with the CDMF encryption algorithm. The following encryption algorithms can be used to encrypt/decrypt the data:

- DES_CBC_8: Data Encryption Standard level of encryption (uses a 56-bit key and a 64-bit initialization vector)
- 3DES_CBC: Data Encryption Standard level of encryption (executed three times and uses a 24-byte key)
- ESP_NULL: No encryption

9.4.2 Security associations

The concept of a security association (SA) is fundamental to IPSec. An SA is a unidirectional logical connection between two IPSec systems, uniquely identified by the following triple:

```
<Security Parameter Index, IP Destination Address, Security Protocol>
```

The definition of the members is as follows:

Security Parameter Index (SPI)
This is a 32-bit value used to identify different SAs with the same destination address and security protocol. The SPI is carried in the header of the security protocol (AH or ESP). The SPI has only local significance, as defined by the creator of the SA. The SPI values in the range 1 to 255 are reserved by the Internet Assigned Numbers Authority (IANA). The SPI value of 0 must be used for local implementation-specific purposes only. Generally the SPI is selected by the destination system during the SA establishment.

IP Destination Address
This address can be a unicast, broadcast or multicast address. However, currently SA management mechanisms are defined only for unicast addresses.

Security Protocol
This can be either AH or ESP.

An SA can be in either of two modes: transport or tunnel, depending on the mode of the protocol in that SA. Because SAs are simplex, for bidirectional communication between two IPSec systems, there must be two SAs defined, one in each direction.

An SA gives security services to the traffic carried by it either by using AH or ESP, but not both. In other words, for a connection that should be protected by both AH and ESP, two SAs must be defined for each direction. In this case, the set of SAs that define the connection is referred to as an *SA bundle*. The SAs in the bundle do not have to terminate at the same endpoint. For example, a mobile host could use an AH SA between itself and a firewall and a nested ESP SA that extends to a host behind the firewall.

A security association is possible between hosts that implement RFCs 1825-1829, or the new RFCs 2401-2406 and 2410.

9.4.3 Modes of operation

To set up a secure communication between two hosts, SAs must be negotiated and managed during the use of the tunnel. Key management is a complex issue for IP Security and new standards are being developed to ensure interoperability. Two types of tunnels are supported by the OS/390 Firewall Technologies, each using different key management techniques. They are:

- Manual tunnels (static, IETF-standard)
- ISAKMP tunnels (dynamic, IETF-standard)

Tunnel and key management
Manual tunnels provide backward compatibility to all header types and will interoperate with systems that do not have ISAKMP support. The disadvantage of manual tunnels is that the key values are static. In other words, the encryption and authentication keys are the same for the life of the tunnel and must be manually updated. The manual tunnel support has the widest choice of header and encryption options.

Dynamic tunnels are based the ISAKMP standards developed by the IETF. Dynamic tunnels negotiate and refresh security parameters and exchange keys

securely. Two types of authentication are supported - pre-shared key and digital signature.

The negotiation uses a two-phase approach. The first phase authenticates the communicating parties and specifies the method for security communications. In phase two, IP security associations (SAs) are negotiated and keys are exchanged. The following ISAKMP support is available:

- Authentication with pre-shared keys and digital signatures
- Use of main mode (identity protect mode) and aggressive mode
- Support for Diffie-Hellman groups 1 and 2
- ESP support for triple DES, DES, NULL, and authentication with HMAC_MD5 and HMAC_SHA
- AH support for HMAC_MD5 and HMAC_SHA

The decision to use manual tunnels or dynamic tunnels depends on the tunnel support of the remote end and the type of key management desired. Dynamic tunnels are preferable (when available) because they offer secure key negotiation and key refreshment in an industry-standard way. They also take advantage of the new IETF ESP and AH header types and provide replay protection.

If the remote end does not support dynamic tunnels, or uses one of the algorithms supported only by manual tunnels, manual tunnels should be used. Manual tunnels ensure interoperability with a large number of hosts. Because the keys are static and difficult to change and may be cumbersome to update, they are not as secure.

Tunnel modes

The OS/390 Firewall Technologies allows you to create manual and/or dynamic tunnels that operate in either transport mode or tunnel mode. One of the parameters that must be specified when defining a tunnel is its operational mode.

Tunnel mode protects the entire IP packet. The entire IP packet is encapsulated in an IPSec packet and a new IP protocol header is constructed and attached at the beginning of the IPSec packet to form a new IP packet. The source and destination addresses may or may not be the same as those in the encapsulated IP packets. This mode is typically used for a security association between two firewalls, or between a firewall and a remote system, for example, whenever either of the two communicating systems is not an endpoint of the tunnel. The source and destination addresses in the new IP header are the addresses of the tunnel's endpoints.

Transport mode only protects the transport-layer packet (such as a TCP or a UDP packet) inside an IP packet. In this mode the IP protocol header is first separated from the transport-layer packet, then the transport-layer packet is encapsulated in an IPSec packet. Then the IP protocol header is attached to the IPSec packet to form a new IP packet, and finally the length, protocol, and header checksum fields in the IP protocol header are modified accordingly. The source and destination addresses in the IP protocol header remain unchanged. This mode is used when the endpoints of the security association are the two communicating systems.

9.4.4 VPN customer scenarios

Figure 331 shows three business scenarios that are well suited to the implementation of a VPN solution.

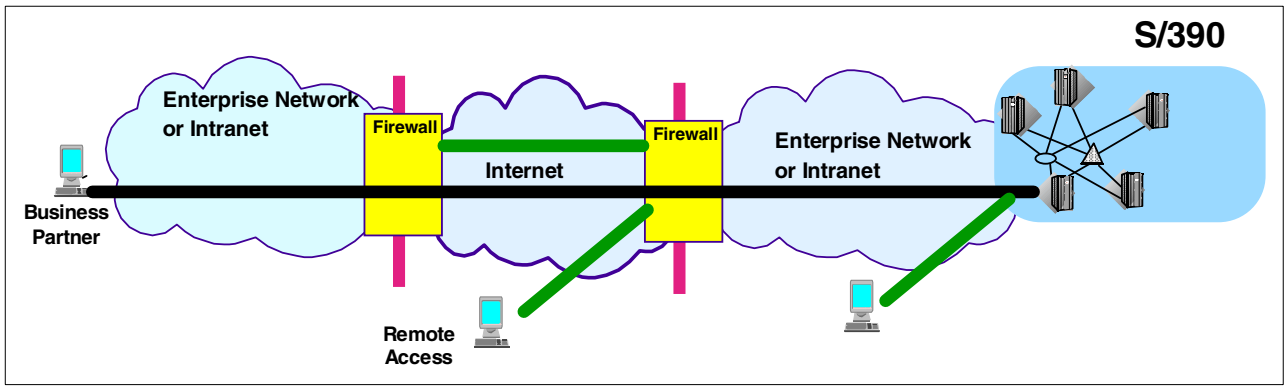


Figure 331. VPN scenarios for an e-business application

IPSec enables a variety of secure e-business configurations depending on your network topology and/or access methods:

- Branch office connection network

  The terminal residing in a branch office can connect to OS/390 systems in a corporate intranet by way of the Internet (branch office interconnect over the Internet). In this scenario the firewall-to-firewall protection will be used, that is, VPN will be established between the firewalls. The end-to-end tunnel also can be used. Tunnel mode will be used in this scenario.

- Remote access network

  In this scenario, the terminal will be a dial-up terminal to the Internet and the client-to-firewall protection will be used. Establishing a VPN between a terminal and a firewall, a user can access the OS/390 system in the intranet securely (remote access from the Internet). To establish the VPN connection, the tunnel mode operation will be selected.

- Business partner/supplier network

  In this scenario, the client could be a terminal in the network of a business partner or supplier, or in the corporate intranet itself. The client terminal can establish the VPN connection directly to OS/390 systems (end-to-end protection), and all application data is transferred through the VPN. Using this scenario, you can send the application data securely. Between the client and server, you can use transport mode to protect the application data. The tunnel mode operation also can be used between the intermediate firewalls and/or the end systems.

9.5 The Internet Key Exchange (IKE) framework overview

The critical elements of IP Security are security associations and the information that they provide with regard to identifying the partners of a secure communications channel, the cryptographic algorithms, and keys to be used. The ISAKMP protocol provides a framework for exchanging messages to automate the negotiation of security associations. The Internet Key Exchange (IKE) framework, also referred to as Internet Security Association Key Management

Protocol/Oakley, is used in conjunction with the ISAKMP protocol to automate the generation and refresh of cryptographic keys. The ability to perform these functions with minimal manual configuration of machines will be a crucial element as a VPN grows in size.

9.5.1 ISAKMP overview

The ISAKMP procedures deal with initializing the keys, so they must be capable of running over networks where no security can be assumed to exist. Therefore, the ISAKMP protocols use the most complex and processor-intensive operations in the IPSec protocol suite.

In addition, the ISAKMP methods have been designed with the explicit goals of providing protection against several well-known exposures:

- Denial-of-Service: The messages are constructed with unique pseudo-random numbers that can quickly identify and reject incorrect messages without the need to execute processor-intensive cryptographic operations.
- Man-in-the-Middle: Protection is provided against the common attacks such as deletion of messages, modification of messages, reflecting messages back to the sender, replying to old messages, and redirection of messages to unintended recipients.
- Lack of Perfect Forward Secrecy (PFS): Compromise of past keys provides no useful clues for breaking any other key, whether it occurred before or after the compromised key. That is, each refreshed key will be derived without any dependence on predecessor keys.

9.5.2 Operation overview

The robustness of any cryptography-based solution depends much more strongly on keeping the keys secret than it does on the actual details of the chosen cryptographic algorithms. Hence, the IETF IPSec working group has prescribed a set of extremely robust ISAKMP exchange protocols. The ISAKMP protocol is implemented at the application layer of TCP/IP and communicates using UDP port 500. The protocol uses a two-phase approach.

9.5.2.1 Phase 1 operations

This set of negotiations establishes a master secret from which all cryptographic keys will subsequently be derived for protecting the user's data traffic. In the most general case, public key cryptography is used to establish an ISAKMP security association between systems, and to establish the keys that are used to protect the ISAKMP messages that flow in the subsequent Phase 2 negotiations. Phase 1 is concerned only with establishing the protection suite for the ISAKMP messages themselves, but it does not establish any security associations or keys for protecting user data.

In Phase 1, the cryptographic operations are the most processor-intensive, but need to be done only infrequently, and a single Phase 1 exchange can be used to support multiple Phase 2 exchanges. As a rule, Phase 1 negotiations are executed once a day or maybe once a week, while Phase 2 negotiations can be executed as often as once every few minutes.

9.5.2.2 Phase 2 operations

Phase 2 exchanges are less complex, since they are used only after the security protection suite negotiated in Phase 1 is activated. A set of communicating systems negotiate the security associations and keys that will protect user data exchanges. Phase 2 ISAKMP messages are protected by the ISAKMP security association generated in Phase 1. Phase 2 negotiations generally occur more frequently than Phase 1.

9.5.2.3 Permanent identifiers

The ISAKMP protocol offers a solution even when the remote host's IP address is not known in advance. ISAKMP allows a remote host to identify itself by a permanent identifier, such as a name or an e-mail address. The ISAKMP Phase 1 exchanges then authenticate the remote host's permanent identity using public key cryptography:

- Certificates create a binding between the permanent identifier and a public key. Therefore, ISAKMP's certificate-based Phase 1 message exchanges can authenticate the remote host's permanent identify.
- Since the ISAKMP messages themselves are carried within IP datagrams, the ISAKMP partner (for example, a firewall or destination host) can associate the remote host's dynamic IP address with its authenticated permanent identity.

For detailed information on the ISAKMP protocol, refer to the corresponding RFCs and to *A Comprehensive Guide to Virtual Private Networks, Volume III: IBM Cross-Platform and Key Management Solutions*, SG24-5309 and *TCP/IP Tutorial and Technical Overview*, GG24-3376.

9.5.3 ISAKMP authentication

The ISAKMP protocol requires that ISAKMP peers authenticate themselves as part of Phase 1 processing. The ISAKMPD server shipped with OS/390 Firewall Technologies supports authentication via either pre-shared keys or RSA Signatures.

With pre-shared key authentication both ISAKMP peers agree to an arbitrary value. This value is included in the calculation that dynamically creates the secret encryption key that will be used to encrypt messages between the two peers. The ability to successfully send a message encrypted with this secret key proves that the pre-shared value is known.

In RSA signature-based authentication each ISAKMP peer signs a piece of information with its private key. If the peer can successfully verify the signature with the public key contained in the ISAKMP peer's certificate, and can also verify the server's certificate, then that ISAKMP server is the entity identified by the certificate. A certificate is a binding between an entity and the public key portion of an asymmetrical cryptographic key pair. A third party, called a Certificate Authority (CA), certifies the identity of the entity involved and that the entity does possess the associated private key.

The ISAKMP protocol requires that ISAKMP peers identify themselves as part of Phase 1 processing by sending a message containing their identity. In RSA signature-based authentication an ISAKMP server's identity is also contained in the subject name field and/or a subject alternate name extension of its certificate.

The identity contained in the certificate must match the identity sent as part of Phase 1 processing.

9.6 Implementing the dynamic tunnels on OS/390

To implement dynamic tunnels in OS/390 you have to implement the firewall services in OS/390. Firewall services implement IKE function in OS/390 in conjunction with SecureWay CS IP Services, Open Cryptographic Services Facility (OCSF), and Security Server.

The following high-level steps are required to define dynamic VPNs:

1. Define policies:
   - Key management
   - Data management
   - Dynamic tunnel
2. Define key server information
3. Define authentication data
4. Define anchor filter rules
5. Define locally activated dynamic connections

Each high-level step is comprised of a series of lower-level steps that are covered in detail in 9.7.1, “Creating a dynamic VPN connection using the GUI panels” on page 299.

9.6.1 OS/390 SecureWay CS IP services customization

To configure CS/390 IP to support IKE functions you need to do the following steps. If you need more information see *OS/390 Firewall Technologies Guide and Reference*, SC24-5835. At ITSO Raleigh, we used the TCPIPB stack in system RA03 to implement the firewall function.

9.6.1.1 Configure the TCP/IP profile

Add the device statements that you will use to connect the OS/390 with the networks. Refer to*OS/390 SecureWay Communications Server IP Configuration*, SC31-8513 for more information. Look at the example below:

```
DEVICE TR1B LCS      2020 AUTORESTART
LINK  TR1B  IBMTR    0    TR1B
```

If you want to start FWKERN automatically, insert the following AUTOLOG statement:

```
AUTOLOG
   FWKERN           ; OS/390 Firewall
ENDAUTOLOG
```

Note: If you are running a system that will be connected to the Internet or to another nonsecure network you should remove any AUTOLOG statement for the standard TCP/IP servers. They should start only after FWKERN.

Add the following PORT statements:

```
500 UDD OMVS                    ; OS/390 Firewall ISAKMP Server
514 UDP OMVS                    ; OS/390 Firewall Syslogd Server
```

```
1014 TCP OMVS                    ; OS/390 Firewall Configuration Client
```

Add the following definitions in the IPCONFIG block statement:

```
IPCONFIG
     FIREWALL
     DATAGRAMFWD
```

The FIREWALL keyword cannot be dynamically activated using the VARY OBEY console command. To activate the firewall function, you have to restart the TCP/IP stack.

9.6.1.2 Configure /etc/services

Create the /etc/services file if it does not exist, then add the definitions for the SYSLOG server and ISAKMP server as follows:

```
syslog   514/udp
isakmp   500/udp
```

9.6.1.3 Checking TCP/IP configuration

During the TCP/IP stack initialization you can check if the following messages appear in the system log:

```
EZZ0641I IP FORWARDING NOFWDMULTIPATH SUPPORT IS ENABLED
EZZ0349I FIREWALL SUPPORT IS ENABLED
```

You can check all the other configurations you made using the following commands:

```
Display TCPIP,TCPIPB,N,CONFIG 1
EZZ2500I NETSTAT CS V2R8 TCPIPB
TCP CONFIGURATION TABLE:
DEFAULTRCVBUFSIZE:  00065536  DEFAULTSNDBUFSIZE: 00065536
DEFLTMAXRCVBUFSIZE: 00262144
MAXRETRANSMITTIME:  120.000   MINRETRANSMITTIME: 0.500
ROUNDTRIPGAIN:      0.125     VARIANCEGAIN:      0.250
VARIANCEMULTIPLIER: 2.000     MAXSEGLIFETIME:    60.000
DEFAULTKEEPALIVE:   0.120     LOGPROTOERR:       00
TCPFLAGS:           10
UDP CONFIGURATION TABLE:
DEFAULTRCVBUFSIZE: 00016384  DEFAULTSNDBUFSIZE: 00016384
CHECKSUM:          00000001  LOGPROTOERR:       01
UDPFLAGS:          23
IP CONFIGURATION TABLE:
FORWARDING: YES 2   TIMETOLIVE: 00060  RSMTIMEOUT:  00015
FIREWALL:   00001 3 ARPTIMEOUT: 01200  MAXRSMSIZE:  65535
```

Figure 332. Report from Netstat CONFIG command

Figure 332 notes:

1 This command shows configuration information about the TCP/IP stack.

2 FORWARDING: YES means that TCP/IP will transfer data between networks. NO means that this stack will not transfer data between networks.

3 FIREWALL: 00001 means that the firewall function is active in this stack. 00000 means that this stack is not activated.

To check if the devices are well configured and ready, issue the following command:

```
Display TCPIP,TCPIPB,N,DEvlinks 1
EZZ2500I NETSTAT CS V2R8 TCPIPB 757
DEVNAME: LOOPBACK          DEVTYPE: LOOPBACK  DEVNUM: 0000
  LNKNAME: LOOPBACK          LNKTYPE: LOOPBACK    STATUS: READY
    NETNUM: 0   QUESIZE: 0   BYTEIN: 0000308050   BYTEOUT: 0000308050
  BSD ROUTING PARAMETERS:
    MTU SIZE: 00000             METRIC: 00
    DESTADDR: 0.0.0.0           SUBNETMASK: 0.0.0.0
  MULTICAST SPECIFIC:
    MULTICAST CAPABILITY: NO
DEVNAME: TR1B              DEVTYPE: LCS       DEVNUM: 2020
  LNKNAME: TR1B              LNKTYPE: TR          STATUS: READY 2
    NETNUM: 0   QUESIZE: 0   BYTEIN: 0001000461   BYTEOUT: 0001722471
    ARPMACADDRESS: NON-CANONICAL   SRBRIDGINGCAPABILITY: YES
    BROADCASTCAPABILITY: YES       BROADCASTTYPE: ALL RINGS
  BSD ROUTING PARAMETERS:
    MTU SIZE: 00000             METRIC: 00
    DESTADDR: 0.0.0.0           SUBNETMASK: 255.255.255.0
  MULTICAST SPECIFIC:
    MULTICAST CAPABILITY: YES
```

Figure 333. Report from Netstat DEvlinks command

Figure 333 notes:

1 This command displays information about devices and links defined in the TCP/IP stack.

2 STATUS: READY means that this device is ready and operational.

To check if the port statements are well configured, issue the following command:

```
Display TCPIP,TCPIPB,N,PORTList 1
EZZ2500I NETSTAT CS V2R8 TCPIPB 779
PORT# PROT USER     FLAGS RANGE
00007 TCP  MISCSERV A
00009 TCP  MISCSERV A
00019 TCP  MISCSERV A
00020 TCP  OMVS
00021 TCP  FTPD1    A
00021 TCP  FTPDB1   A
00025 TCP  SMTP     A
00053 TCP  OMVS     A
00080 TCP  OMVS     A
00111 TCP  OMVS     A
00443 TCP  OMVS     A
00500 UDP OMVS    A    2
00514 UDP OMVS    A    2
00515 TCP  T03ALPD  A
00750 TCP  MVSKERB  A
00751 TCP  ADM@SRV  A
00760 TCP  IOASNMP  A
01014 TCP OMVS    A     2
```

Figure 334. Report from Netstat PORTList command

Figure 334 notes:

1 This command shows information about port reservation in the TCP/IP stack.

2 Ports 500 and 514 over protocol UDP, and 1014 over TCP, are reserved for UNIX System Services application, that is FWKERN in this case.

9.6.2 UNIX System Services customization

There are some parameters in BPXPRMxx member in SYS1.PARMLIB that must be checked and changed if necessary. Check the following parameters:

- MAXPROCSYS: The firewall requires 11 processes to start its servers. The default is 200.
- MAXPROCUSER: The firewall requires 11 processes to start its servers. The default is 25.
- MAXFILEPROC: The firewall requires at least 25 open file descriptors for its servers, two for each concurrent connection to the SOCKS server and four for each concurrent connection to the FTP proxy server. The default is 64.
- MAXTHREADTASKS: The firewall requires approximately 10 threads for firewall servers, one thread for each concurrent connection to the SOCKS server, and one thread for each concurrent connection to the proxy FTP server. The default is 50.
- MAXTHREAD: The SOCKS and proxy FTP servers require one thread for each concurrent connection. The default is 200.
- MAXSOCKETS: The firewall requires approximately 25 sockets for firewall servers, two for each concurrent connection to the SOCKS server, and four for each concurrent connection to the FTP proxy server. ISAKMP requires one additional socket per interface defined to a firewall stack. The default is 64.

Check if you have AF_UNIX configured. If not, add the following statements to the BPXPRMxx file:

```
NETWORK DOMAINNAME(AF_UNIX)
        DOMAINNUMBER(1)
        MAXSOCKETS(100)
        TYPE(IBMUDS)
```

For more information about the BPXPRMxx configuration, see *OS/390 UNIX System Services Planning*, SC28-1890, and *OS/390 Initialization and Tuning Reference*, SC28-1752.

9.6.3 OS/390 Security Server and cryptographic services customization

The following procedures assume that you are performing them from a RACF administration ID. Table 10 provides a brief description of the RACF profiles we are using here.

Table 10. RACF profile description

Class	Profile	Description
FACILITY	BPX.SMF	Checks if the caller attempting to cut an SMF record is allowed to write an SMF record or test if an SMF type or subtype is being recorded.

Class	Profile	Description
FACILITY	BPX.DAEMON	Restricts access to the following services: seteuid, setuid, setruid, setreuid, and spawn. The caller of this service must be a superuser.
FACILITY	BPX.SERVER	Restricts the use of the pthread_security_np service. A user with read or write access to the BPX.SERVER FACILITY class profile can use this service. It creates or deletes the security environment for the caller's thread. This profile is also used to restrict the use of the BPX1ACK service, which determines access authority to an OS/390 resource.
FACILITY	BPX.SUPERUSER	Users with access to the BPX.SUPERUSER FACILITY class profile can switch to superuser authority (effective UID of 0).
FACILITY	BPX.FILEATTR.APF	Controls that users are allowed to set the APF-authorized attribute in an HFS file. This authority allows the user to create a program that will run APF-authorized. This is similar to the authority of allowing a programmer to update SYS1.LINKLIB or SYS1.LPALIB.
FACILITY	BPX.FILEATTR.PROGCTL	Controls that users are allowed to set the program-controlled attribute in an HFS file. Programs marked with this attribute can execute in server address spaces that run with a high level of authority.
FACILITY	ICA.CFGSRV	Permits access to the Firewall Configuration GUI.
FACILITY	IRR.DIGTCERT.ADD	Permission to add a certificate.
FACILITY	IRR.DIGTCERT.ADDRING	Permission to add a key ring.
FACILITY	IRR.DIGTCERT.CONNECT	Permission to connect to a key ring.
FACILITY	IRR.DIGTCERT.DELETE	Permission to delete a certificate.
FACILITY	IRR.DIGTCERT.DELRING	Permission to delete a key ring.
FACILITY	IRR.DIGTCERT.GENCERT	Permission to generate a certificate.
FACILITY	IRR.DIGTCERT.GENREQ	Permission to generate a certificate request.
FACILITY	IRR.DIGTCERT.LIST	Permission to list a certificate.
FACILITY	IRR.DIGTCERT.LISTRING	Permission to list a key ring.
FACILITY	CDS.CSSM.CRYPTO	Authorizes the daemon to call a Cryptographic Service Provider.
FACILITY	CDS.CSSM.DATALIB	Authorizes the daemon to call a Data Library (DL) Service Provider.

Class	Profile	Description
FACILITY	FWKERN.START.REQUEST	Permits FWKERN to issue the START console command and to start its servers. It also controls firewall administrator IDs that are allowed to issue the fwdaemon command to start and stop firewall servers.
STARTED	STC name	The STARTED class allows you to assign RACF identities to started procedures and jobs dynamically, using the RDEFINE and RALTER commands. Unlike the started procedures table, it does not require you to modify code or re-IPL in order to add or modify RACF identities for started procedures. It provides, in effect, a dynamic started procedures table.
CSFSERV	ICSF Services	Controlling use of Integrated Cryptographic Service Facility (ICSF) cryptographic services.

9.6.3.1 Defining users and groups

Define the firewall kernel address space to RACF or an equivalent security product:

- Define FWKERN user.
- Define the firewall kernel started procedure as a started task.

```
ADDGROUP FWGRP SUPGROUP(SYS1) OMVS(GID(nnn))
MKDIR '/u/fwkern' MODE(7,5,5)
ADDUSER FWKERN OMVS(HOME('/u/fwkern/') UID(0))
        DFLTGRP(FWGRP) AUTHORITY(CREATE) UACC(ALTER) PASSWORD(pw)
RDEFINE STARTED FWKERN STDATA(USER(FWKERN))
SETROPTS RACLIST(STARTED) REFRESH
```

Note: nnn is the installation-defined group ID for the FWGRP group that was identified in the previous steps; pw is the password for the FWKERN user ID. Choose this password with extreme care to avoid potential security exposure.

9.6.3.2 Granting users and groups authority to firewall objects

Create the FWKERN.START.REQUEST resource profile:

```
RDEFINE FACILITY FWKERN.START.REQUEST UACC(NONE)
PERMIT FWKERN.START.REQUEST CLASS(FACILITY) ID(FWKERN) ACCESS(UPDATE)
SETROPTS CLASSACT(FACILITY)
```

Permit FWKERN to access to start the servers:

```
RDEFINE STARTED ICAPSLOG.**  STDATA(USER(FWKERN) GROUP(FWGRP))
RDEFINE STARTED ICAPSOCK.**  STDATA(USER(FWKERN) GROUP(FWGRP))
RDEFINE STARTED ICAPPFTP.**  STDATA(USER(FWKERN) GROUP(FWGRP))
RDEFINE STARTED ICAPCFGS.**  STDATA(USER(FWKERN) GROUP(FWGRP))
RDEFINE STARTED ICAPSTAK.**  STDATA(USER(FWKERN) GROUP(FWGRP))
RDEFINE STARTED ICAPIKED.**  STDATA(USER(FWKERN) GROUP(FWGRP))
SETROPTS RACLIST(STARTED) REFRESH
```

Permit FWKERN read access to the TCP/IP data sets if necessary:

```
PERMIT 'TCPIP.**' ID(FWKERN) ACCESS(READ)
```

Permit FWKERN read access to the BPX.SMF facility (for logging):

```
RLIST FACILITY BPX.SMF ALL
RDEFINE FACILITY BPX.SMF UACC(NONE)
PERMIT BPX.SMF CLASS(FACILITY) ID(FWKERN) ACCESS(READ)
```

Permit FWKERN to read the BPX.DAEMON facility:

```
RLIST FACILITY BPX.DAEMON
RDEFINE FACILITY BPX.DAEMON UACC(NONE)
PERMIT BPX.DAEMON CLASS(FACILITY) ID(FWKERN) ACCESS(READ)
```

Once you activate the BPX.DAEMON facility, all programs loaded into the fwkern address space have to be RACF program controlled.

The firewall server programs must be marked as program controlled. It is also necessary to mark the System SSL library hlq.SGSKLOAD as program controlled. This can be done by using the following commands:

```
RALTER PROGRAM * ADDMEM('ICA.SICALMOD'/'******'/NOPADCHK) UACC(READ)
RALTER PROGRAM * ADDMEM('hlq.SGSKLOAD'/'******'/NOPADCHK) UACC(READ)
SETROPTS WHEN(PROGRAM) REFRESH
```

To use the configuration server, some setup is required.

Note: All user IDs that will use the firewall configuration GUI must be explicitly given permission to update the configuration through the configuration server. This includes user IDs that have superuser privileges or are members of the firewall group.

```
RDEFINE FACILITY ICA.CFGSRV UACC(NONE)
PERMIT ICA.CFGSRV CLASS(FACILITY) ID(userid) ACCESS(UPDATE)
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY) REFRESH
```

Adding firewall administrators to FWGRP:

If the user IDs that will administer the firewall are not superusers (UID=0), add them to the FWGRP group as follows:

```
CONNECT userid GROUP(FWGRP)
```

The ISAKMP server supports the ability to perform peer authentication using RSA signature mode. RSA signature mode requires that digital certificates be stored in RACF and connected to a key ring. RACF provides digital certificate and key ring support using the `RACDCERT` command. The authorizations necessary to perform the basic RACDCERT actions are shown below:

```
RDEFINE FACILITY IRR.DIGTCERT.ADD UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.ADDRING UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.CONNECT UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.GENCERT UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.GENREQ UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.ADD CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.ADDRING CLASS(FACILITY) ID(userid) ACC(UPDATE)
PERMIT IRR.DIGTCERT.CONNECT CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.GENREQ CLASS(FACILITY) ID(userid) ACC(CONTROL)
```

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACC(UPDATE)
SETROPTS RACLIST(FACILITY) REFRESH
```

Note: `userid` is the ID of the person who will be executing the `RACDCERT` command to store digital certificates.

Authority to the IRR.DIGTCERT.function resource in the FACILITY class allows a user to issue the `RACDCERT` command that is used to install and maintain digital certificates and key rings in RACF. To issue the `RACDCERT` command, users must have one of the following authorities:

- The SPECIAL attribute
- Sufficient authority to resource IRR.DIGTCERT.function in the FACILITY class.
  - READ access to IRR.DIGTCERT.function to issue the `RACDCERT` command for themselves.
  - UPDATE access to IRR.DIGTCERT.function to issue the `RACDCERT` command for others.
  - CONTROL access to IRR.DIGTCERT.function to issue the `RACDCERT` command for SITE and CERTAUTH certificates. This authority also has other uses.

Refer to the *OS/390 Security Server (RACF) Command Language Reference*, SC28-1919, for a complete description of the facilities and authorizations needed to create and modify digital certificates and key rings.

9.6.3.3 Installing OCSF code

To perform the cryptographic functions needed by the IKE function, you have to install and configure Open Cryptographic Services Facility (OCSF). See 2.1.1, “Installation of OCSF” on page 14 on how to install OCSF.

The ISAKMP firewall (IKE function) runs as an APF-authorized application. So, we have to turn on the APF-authorized extend attribute for the OCSF and OCEP dynamically loaded libraries.You can turn on the APF-authorized extended attribute using the `extattr +a` command.

Mark the OCSF programs in the OCSF UNIX Library as APF authorized and program controlled using the `extattr` command. To issue the `extattr` command, the user ID has to have access to a specific RACF class profile:

```
RDEFINE FACILITY BPX.FILEATTR.APF ACC(NONE)
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) USER(GIANCA) ACCESS(UPDATE)
RDEFINE FACILITY BPX.FILEATTR.PROGCTL ACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) USER(GIANCA) ACCESS(UPDATE)
SETROPTS CLASS(FACILITY) REFRESH
```

From the command prompt in the OS/390 UNIX shell, issue the following commands:

```
$ cd /usr/lpp/ocsf/lib
$ extattr +a *.dll  1
$ ls -E *.dll         2
-rwxr-xr-x  aps  2 OMVSKERN SYS1       49152 May 11 21:20 cdserprt.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1       86016 May 11 21:20 cdsibmut.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1       86016 May 11 21:20 cdskwtf.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1     4173824 May 11 21:20 cdsnspsp.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1      192512 May 11 21:20 cdsport.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1      188416 May 11 21:21 cdsrandm.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1      823296 May 11 21:19 cssm32.dll
lrwxrwxrwx       1 OMVSKERN SYS1          16 May 11 21:20 cssmmanp.dll -> cssmma
np_sl3.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1       36864 May 11 21:20 cssmmanp_sl3.dll
lrwxrwxrwx       1 OMVSKERN SYS1          16 May 11 21:20 cssmusep.dll -> cssmus
ep_sl3.dll
-rwxr-xr-x  aps  2 OMVSKERN SYS1       36864 May 11 21:20 cssmusep_sl3.dll

$ cd /usr/lpp/ocsf/addins
$ extattr +a *.so   1
$ ls -E *.so          2
-rwxr-xr-x  aps  2 OMVSKERN   SYS1      450560 May 11 21:20 ibmcca.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1      589824 May 11 21:20 ibmcl.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     1474560 May 11 21:21 ibmcl2.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     5701632 May 11 21:21 ibmdl2.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     6856704 May 11 21:20 ibmocepdl.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1      425984 May 11 21:20 ibmoceptp.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     1138688 May 11 21:20 ibmswcsp.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1       57344 May 11 21:21 ibmtp.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     3563520 May 11 21:21 ibmtp2.so
-rwxr-xr-x  aps  2 OMVSKERN   SYS1     1069056 May 11 21:21 ibmwkcsp.so
```

Figure 335. extattr shell command

Figure 335notes:

1 Give the APF authorization attributes to all files with the dll suffix in the /usr/lpp/ocsf/lib directory and files with the so suffix in /usr/lpp/ocsf/addins.

2 Using the `ls` command with the -E option, you can see the extended attributes of the HFS files. The `a` and `p` flags in the second column indicate that the files do have the API-authorized and program-controlled attribute.

Grant the permission to use OCSF services to the FWKERN user ID:

```
PERMIT CDS.CSSM CLASS(FACILITY) ID(FWKERN) ACC(READ)
PERMIT CDS.CSSM.CRYPTO CLASS(FACILITY) ID(FWKERN) ACC(READ)
PERMIT CDS.CSSM.DATALIB CLASS(FACILITY) ID(FWKERN) ACC(READ)
```

Permit FWKERN access to the BPX.SERVER:

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(FWKERN) ACC(READ)
```

Refresh the FACILITY class and PROGRAM class profile:

```
SETROPTS CLASS(FACILITY) REFRESH
```

9.6.3.4 Installing Open Cryptographic Enhanced Plug-Ins (OCEP)

The ISAKMP firewall (IKE function) also makes use of the OCEP functions, therefore you need to install OCEP as shown in 2.2.1, “Installation of OCEP” on page 20.

Give the FWKERN user ID permission to the following FACILITY class profiles:

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(FWKERN) ACC(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(FWKERN) ACC(READ)
SETROPTS CLASS(FACILITY) REFRESH
```

9.6.3.5 ICSF/MVS authorization

OS/390 Firewall Technologies can take advantage of the encryption and decryption functions available in the new generation of System/390 processors. The firewall uses several of these functions in two ways:

- Encrypting and decrypting TCP/IP packet data in an IP tunnel.
- Encrypting signature data included as part of the ISAKMP message flows. This encryption is only performed when RSA signature mode authentication is requested and the associated certificate was defined using the `RACDCERT` command that specified the ICSF keyword.

This support is provided by the combination of the Integrated Cryptographic Feature (ICRF) on the processor or the CMOS crypto chip available with all CMOS G4/G5/G6 processors and the Integrated Cryptographic Service Facility/MVS (ICSF) software product.

To use this support, ICFS must be started and running. It is preferable to do this prior to staring TCP/IP; however, it can also be done when TCP/IP is active.

Note: The remaining configuration is only applicable to the use of hardware crypto when encrypting and decrypting TCP/IP packet data in an IP tunnel.

If you plan to use this hardware crypto support and issue TCP/IP commands such as OPING from a user ID on the system where the firewall is running, this user ID must be permitted to access the ICSF cryptographic services if protected by an external security manager in the CSFSERV class. Perform the following steps to set up profiles in the CSFSERV resource class and permit users to access these profiles:

Define the appropriate profiles in the CSFSERV class:

```
RDEFINE CSFSERV service-name UACC(NONE)
```

The service names that the OS/390 Firewall Technologies uses are:

- CSFCKI
- CSFDEC1
- CSFENC1
- CSFRNG
- CSFCKM
- CSFOWH1

Note: If triple DES hardware crypto support is not available on your S/390 processor, the CSFCKM service is not used.

Permit user access to these profiles:

```
PERMIT profile-name CLASS(CSFSERV) ID(yourid) ACCESS(READ)
```

The CSFSERV class needs to be refreshed. This is done by the RACF administrator using the following RACF command:

```
SETROPTS RACLIST(CSFSERV)REFRESH
```

The MAXLEN installation option for hardware crypto determines the maximum length that can be used to encrypt and decrypt data using ICSF. Set the MAXLEN ICSF installation option to greater than 65527 as this is the maximum TCP/IP packet size.

9.6.4 OS/390 Firewall USS customization and starting

9.6.4.1 Copying shell scripts

OS/390 Firewall Technologies contains the following executable shell scripts:

```
fwlogmgmt
getmsg
```

Running shell scripts from locales that are not generated from code page IBM-1047 requires multiple copies of each shell script, one for each different locale's code page. You can use the `iconv` command to convert a shell script from one code page to another. For example, to convert the fwlogmgmt script to the Da_DK.IBM-277 locale, enter the following command:

```
iconv -f IBM-1047 -t Da_DK.IBM-277 /usr/lpp/fw/bin/fwlogmgmt > /tmp/fwlogmgmt
```

For more information about the `iconv` command, see *OS/390 UNIX System Services Command Reference*, SC28-1892.

For more information about customizing your locale, see *OS/390 UNIX System Services User's Guide*, SC28-1891, and *OS/390 UNIX System Services Planning*, SC28-1890.

9.6.4.2 Activate sample configuration files

Note: It is important to preserve the owner, group, and mode settings when copying sample files. This can be done using the -p option of the `cp` command from a superuser (UID=0). For example:

```
cp -p /usr/lpp/fw/etc/security/fwrules.cfg /etc/security/fwrules.cfg
```

The following sample configuration file is shipped with OS/390 Firewall Technologies in /usr/lpp/fw/etc:

- syslog.conf: Logging server configuration file

To use this sample, copy it into the /etc directory.

If you do not install the sample syslog.conf file before IPLing your system and starting the firewall, an existing syslog.conf file will be used if one exists. If none exists, default logging will be in effect. Default logging sends all messages with a priority of error and above to the OS/390 operator console (to file /dev/console). You must create the special character file from a superuser by issuing:

```
mknod /dev/console  c 9 0
```

See the *OS/390 UNIX System Services Command Reference*, SC28-1892, for further details on the `mknod` command.

In addition, the following files are shipped with OS/390 Firewall Technologies in /usr/lpp/fw/etc/security. They contain firewall default definitions:

- fwaudio.cfg - real audio configuration file
- fwdaemon.cfg - firewall servers configuration file
- fwobjects.cfg - network objects configuration file
- fwrules.cfg - firewall configured filter rules configuration file
- fwservices.cfg - services configuration file
- fwsocks.cfg - SOCKS configuration file
- logmgmt.cfg - log management configuration file
- fwahtran.cfg - AH transform configuration file
- fwesptran.cfg - ESP transform configuration file
- fwkeypol.cfg - key policy configuration file
- fwkeyprop.cfg - key proposal configuration file
- fwkeyring.cfg - key ring configuration file
- fwkeytran.cfg - key transform configuration file

If you are not migrating from a previous release of OS/390 Firewall Technologies, you must copy these files into the /etc/security directory during installation, if they are not already there.

If you are migrating from a previous release, use the command `fwmigrate` to preserve your current configuration, and copy the following files into the /etc/security directory, if they are not already there. Also see *OS/390 Firewall Technologies Guide and Reference*, SC24-5835 for more information.

- fwahtran.cfg
- fwesptran.cfg
- fwkeypol.cfg
- fwkeyprop.cfg
- fwkeyring.cfg
- fwkeytran.cfg

Whether or not you are migrating from a previous release of OS/390 Firewall Technologies, you must copy the following files into the /etc/security directory during installation:

- fwguicmds.En_US
- fwguicmds.Ja_JP (if Japanese version is installed)

In our configuration we changed the file /etc/syslog.conf. Our configuration is shown in Figure 336 on page 286.

```
 BROWSE -- /etc/syslog.conf ------------------------ Line 00000000 Col 001 034
 Command ===>                                          Scroll ===> CSR
******************************** Top of Data *********************************
#local0.* /tmp/firewall.local0.log
#local4.* /tmp/firewall.local4.log
#syslog.* /tmp/firewall.syslog.log
*.*      /tmp/firewall.all.log
******************************* Bottom of Data ********************************
```

Figure 336. SYSLOGD configuration file

Using this configuration all messages directed to the SYSLOGD server will be written in /tmp/firewall.all.log.

9.6.4.3 Defining firewall stack

To define the firewall stack we have to use the `fwstack` command in the UNIX System Services shell prompt:

```
$ fwstack cmd=add stack=tcpipb
```

Figure 337. Defining the TCPIPB stack to firewall kernel

In this release the firewall code can control many TCP/IP stacks in the same OS/390 image.

9.6.4.4 Configuring firewall servers

All the OS/390 Firewall Technologies servers run in their own address spaces. Servers are controlled by the control task running in the firewall kernel referred to as the FWKERN address space.

The FWKERN address space must be started before any of the servers are started. All requests to start, stop, or query the firewall servers (either collectively or individually) are made through the FWKERN control task through the `START`, `STOP`, or `MODIFY` commands, which you issue from the OS/390 operator console.

Use the `fwdaemon` command to list and change server configuration attributes, query server status, and start and stop servers.

At this time we only have to define four servers to the firewall stack. Go to the UNIX System Services shell prompt and use the `fwdaemon` command:

```
$ fwdaemon cmd=change daemon=syslogd started=yes
$ fwdaemon cmd=change daemon=fwstackd started=yes
$ fwdaemon cmd=change daemon=isakmpd started=yes
$ fwdaemon cmd=change daemon=cfgsrv started=yes \
     daemonopts="-f /etc/security/fwcfgsrv.kdb 1 -p 1014" 2
```

Figure 338. Defining firewall servers

Figure 338 notes:

1 Key database file for the SSL connection. When you create the key database for the firewall client, follow the steps in 9.1.1.5, “SSL setup for the firewall configuration client” on page 260.

2 TCP/IP port number where the server will be listening.

9.6.4.5 Start the firewall kernel

Before starting the firewall code you have to be sure that all previous configurations are available: the TCP/IP configuration, the BPXPRMxx configuration; ICA.SICALMOD and GSK.SGSKLOAD must be APF authorized and the firewall code at /usr/lpp/fw must be mounted and available. We insert both libraries in the LNKLSTxx to make them available to the firewall servers and to prevent updating many procedures.

You have to copy all members of the ICA.SICAPROC data set to a library in the JES concatenated started procedure libraries or concatenate this data set to JES.

Starting the firewall kernel:

```
S FWKERN
$HASP100 FWKERN   ON STCINRDR
IEF695I START FWKERN   WITH JOBNAME FWKERN   IS ASSIGNED TO USER FWKERN
 , GROUP OMVSGRP
$HASP373 FWKERN   STARTED
IEF403I FWKERN - STARTED - TIME=16.16.38
ICAM1057i Release 2.8.0 Service Level 0000000. Created on Jun 22 1999.
$HASP100 ICAPSLOG ON STCINRDR
$HASP373 ICAPSLOG STARTED
IEF403I ICAPSLOG - STARTED - TIME=16.16.42
ICAM1069i Daemon SYSLOGD has been started.
$HASP100 ICAPCFGS ON STCINRDR
$HASP373 ICAPCFGS STARTED
IEF403I ICAPCFGS - STARTED - TIME=16.16.48
ICAM1069i Daemon CFGSRV has been started.
$HASP100 ICAPIKED ON STCINRDR
$HASP373 ICAPIKED STARTED
IEF403I ICAPIKED - STARTED - TIME=16.16.57
ICAM1069i Daemon ISAKMPD has been started.
$HASP100 ICAPSTAK ON STCINRDR
$HASP373 ICAPSTAK STARTED
IEF403I ICAPSTAK - STARTED - TIME=16.17.18
ICAM1069i Daemon FWSTACKD has been started.
ICAM1003i FWKERN initialization complete.
```

Figure 339. Starting FWKERN

To start the firewall kernel go to the OS/390 console, type `S FWKERN` and press Enter. FWKERN will start all the firewall servers configured previously. You have to check the message ICAM1003i to be sure that all servers were started successfully.

There are some console commands you can use to check the firewall status, to stop or start a firewall server, and to stop FWKERN.

Some examples are shown in Figure 340 on page 288.

```
F FWKERN,QUERY ISAKMPD
ICAM1001i Firewall daemon ISAKMPD status is READY and process id is 145
335544366.
F FWKERN,QUERY SYSLOGD
ICAM1001i Firewall daemon SYSLOGD status is READY and process id is 147
33554455.
F FWKERN,QUERY FWSTACKD
ICAM1001i Firewall daemon FWSTACKD status is READY and process id is 149
50331695.
+JSX015 JESX     SLU=RMT3     T011,RTNCD=1000 LU NOT AVAILABLE
F FWKERN,QUERY CFGSRV
ICAM1001i Firewall daemon CFGSRV status is READY and process id is 152
67108917.
F FWKERN,QUERY LEVEL
ICAM1057i Release 2.8.0 Service Level 0000000. Created on Jun 22 1999.
```

Figure 340. Examples of FWKERN console commands

After starting the FWKERN check the socket connections that are open. You can use either the console command `Display TCPIP, TCPIPB, Netstat, SOCKETS,` or the UNIX shell command `netstat -p tcpipb -s`. We used the shell command, as shown in Figure 341 on page 289.

```
GIANCA @ RA03:/u/gianca>netstat -p tcpipb -s
MVS TCP/IP onetstat CS V2R8       TCPIP Name: TCPIPB           14:36:43
Sockets interface status:
Type   Bound to              Connected to          State    Conn
====   ========              ============          =====    ====
Name: FTPD1     Subtask: 007E7390
Stream 0.0.0.0..21           0.0.0.0..0            Listen   00000048
Name: HODSRV3   Subtask: 007E4E78
Stream 0.0.0.0..8999         0.0.0.0..0            Listen   00000021
Name: HODSRV3   Subtask: 007E7390
Stream 0.0.0.0..8989         0.0.0.0..0            Listen   0000003C
Name: ICAPCFGS  Subtask: 007E7420
Stream 0.0.0.0..1014  1      0.0.0.0..0            Listen   0000014A
Name: ICAPIKED  Subtask: 007E7420
Dgram  9.24.104.33..500      *..*                  UDP      0000014C
Dgram  192.168.100.100..500  *..*  2               UDP      0000014D
Dgram  172.16.233.4..500     *..*                  UDP      0000014E
Name: ICAPSLOG  Subtask: 007E7420
Dgram  0.0.0.0..514  3       *..*                  UDP      00000149
Name: TCPIPB    Subtask: 007D2AB0
Stream 0.0.0.0..9923         0.0.0.0..0            Listen   00000019
Name: TCPIPB    Subtask: 007D2CD0
Stream 0.0.0.0..8823         0.0.0.0..0            Listen   00000018
Name: TCPIPB    Subtask: 007D2E68
Stream 0.0.0.0..7723         0.0.0.0..0            Listen   00000017
Name: TCPIPB    Subtask: 007E1630
Stream 0.0.0.0..23           0.0.0.0..0            Listen   00000015
Name: TCPIPB    Subtask: 007E1AC8
Stream 0.0.0.0..6623         0.0.0.0..0            Listen   00000016
Name: TCPIPB    Subtask: 007E33F8
Stream 127.0.0.1..1025       127.0.0.1..1026       Establsh 00000013
Stream 0.0.0.0..1025         0.0.0.0..0            Listen   0000000C
Name: TCPIPB    Subtask: 007E3BF8
Dgram  0.0.0.0..1252         *..*                  UDP      000001FF
Name: TCPIPB    Subtask: 007EC1B0
Stream 127.0.0.1..1026       127.0.0.1..1025       Establsh 00000012
Name: WEBSRV    Subtask: 007EC9B0
Stream 0.0.0.0..80           0.0.0.0..0            Listen   00000020
```

Figure 341. Report from netstat -p OS/390 UNIX command

Figure 341 notes:

1 This is the firewall CFGSRV daemon listening on port 1014.

2 This is the firewall ISAKMPD daemon waiting for UPD packets in all active interfaces on port 500.

3 This is the firewall SYSLOGD daemon waiting for UDP packets in port 514.

See *OS/390 Firewall Technologies Guide and Reference*, SC24-5835 for more information about the commands and firewall configuration.

9.7 Dynamic tunnel scenario

Before we start the Dynamic VPN configuration we have to exchange all IKE parameters with our tunnel partners. In this case we used three partners: one AS/400, one AIX system, and one Windows NT system running an IKE client.

The AS/400 and the Windows NT systems are located in the same subnetwork of OS/390. The AIX system is located in another subnetwork. The configuration

client is located in another network. These three examples will show that the VPN dynamic tunnels can be used in any type of network. It really does not matter if the endpoints of a VPN are in the same or different networks. The VPN has to match your security issues. There are many cases when you would need a VPN inside your intranet. Look at Figure 342 to see our network scenario.

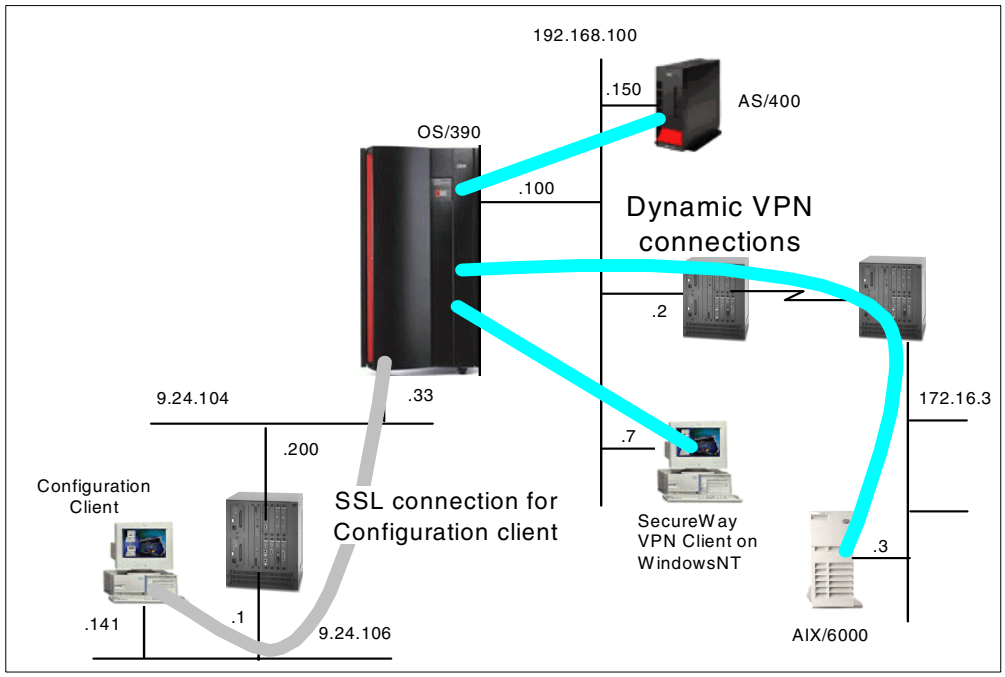


Figure 342. Network scenario of dynamic VPN implementation

The routing table in the TCPIPB stack is shown in Figure 343.

```
GIANCA @ RA03:/u/gianca>netstat -p tcpipb -r
MVS TCP/IP onetstat CS V2R8       TCPIP Name: TCPIPB           14:27:37
Destination      Gateway          Flags  Refcnt  Interface
-----------      -------          -----  ------  ---------
Defaultnet       192.168.100.2    G      000000  TR1B
9.24.104.0       0.0.0.0          U      000000  TR2B
9.24.106.0       9.24.104.200     UG     000001  TR2B
9.179.98.0       9.24.104.200     UG     000000  TR2B
172.16.233.3     0.0.0.0          UH     000000  EZASAMEMVS
172.16.233.28    0.0.0.0          UH     000000  EZAXCF28
172.16.233.39    0.0.0.0          UH     000000  EZAXCF39
192.168.10.0     9.24.104.5       UG     000000  TR2B
192.168.100.2    0.0.0.0          UH     000000  TR1B
192.168.100.150  0.0.0.0          UH     000001  TR1B
```

Figure 343. IKE function scenario: TCPIPB routing table

Now take a look at the sockets connection, shown in Figure 344 on page 291. Note that the IKE server is listening over all interfaces.

```
D TCPIP,TCPIPB,N,CON
EZZ2500I NETSTAT CS V2R8 TCPIPB 235
USER ID  CONN     LOCAL SOCKET           FOREIGN SOCKET         STATE
ICAPCFGS 000000F8 0.0.0.0..1014          0.0.0.0..0             LISTEN
INETD1   00000020 0.0.0.0..109           0.0.0.0..0             LISTEN
INETD1   0000001E 0.0.0.0..2023          0.0.0.0..0             LISTEN
INETD1   0000001F 0.0.0.0..110           0.0.0.0..0             LISTEN
TCPIPB   00000015 0.0.0.0..7723          0.0.0.0..0             LISTEN
TCPIPB   00000016 0.0.0.0..8823          0.0.0.0..0             LISTEN
TCPIPB   00000011 127.0.0.1..1026        127.0.0.1..1025        ESTABLS
TCPIPB   00000014 0.0.0.0..6623          0.0.0.0..0             LISTEN
TCPIPB   00000012 127.0.0.1..1025        127.0.0.1..1026        ESTABLS
TCPIPB   00000017 0.0.0.0..9923          0.0.0.0..0             LISTEN
TCPIPB   0000000C 0.0.0.0..1025          0.0.0.0..0             LISTEN
WEBSRVB  000000AF 0.0.0.0..80            0.0.0.0..0             LISTEN
ICAPIKED 000029C7 192.168.100.100..500   *..*                   UDP
ICAPIKED 000029C6 9.24.104.33..500       *..*                   UDP
ICAPIKED 000029C8 172.16.233.4..500      *..*                   UDP
ICAPSLOG 000029C3 0.0.0.0..514           *..*                   UDP
TCPIPB   00000100 0.0.0.0..1183          *..*                   UDP
16 OF 16 RECORDS DISPLAYED
```

Figure 344. Report from NETSTAT CON command

We created a planning worksheet to be used to organize the information. The content of the worksheet is shown in the following three tables.

Table 11. VPN planning worksheet - S/390 and AS/400

VPN Parameter	Value
Key Policy, Proposal, Transform:	
Initiator Negotiation	Main
Responder Negotiation	Main
Authentication Method	Pre-Shared Keys
Hash Algorithm	MD5
Encryption Algorithm	DES_CBC_8
Diffie-Hellman Group	Group 1
Maximum Key Lifetime	1440
Maximum Size Limit	1000
Key Lifetime Range	60-1440
Size Limit Range	1-1000
Data Policy, Proposal, AH and ESP Transform:	
PFS (Perfect Forward Secrecy)	Group 1
AH Encapsulation Mode	Not applicable
AH Authentication Algorithm	Not applicable
AH Maximum Data Lifetime	Not applicable
AH Maximum Size Limit	Not applicable

VPN Parameter	Value
AH Data Lifetime Range	Not applicable
AH Size Limit Range	Not applicable
ESP Encapsulation Mode	Transport
ESP Authentication Algorithm	HMAC_MD5
ESP Encryption Algorithm	DES_CBC_8
ESP Maximum Data Lifetime	60
ESP Maximum Size Limit	50000
ESP Data Lifetime Range	60-480
ESP Size Limit Range	1-50000
Dynamic Tunnel Policy:	
Initiation	Either
Connection Lifetime	0
Authentication Information:	
Remote Key Server	192.168.100.150
Authentication Method	Pre-Shared Keys
Shared Key	61626364
Certificate Authority:	
Racdcert Label	Not applicable
Key Ring:	
User ID	Not applicable
Key Ring Name	Not applicable
Dynamic Connection:	
Source	192.168.100.100
Destination	192.168.100.150
Source Port	0
Destination Port	0
Automatic Activation	No
Protocol	All
Remote Key Server	192.168.100.150
Key Servers:	
Local Key Server ID Type	IPV4
Local Key Server ID	192.168.100.100
Remote Key Server ID Type	IPV4
Remote Key Server ID	192.168.100.150

Table 12. VPN planning worksheet - S/390 and RS/6000

VPN Parameter	Value
Key Policy, Proposal, Transform:	
Initiator Negotiation	Main
Responder Negotiation	Main
Authentication Method	Pre-Shared Keys
Hash Algorithm	MD5
Encryption Algorithm	DES_CBC_8
Diffie-Hellman Group	Group 1
Maximum Key Lifetime	1440
Maximum Size Limit	0
Key Lifetime Range	60-1440
Size Limit Range	0-0
Data Policy, Proposal, AH and ESP Transform:	
PFS (Perfect Forward Secrecy)	None
AH Encapsulation Mode	Not applicable
AH Authentication Algorithm	Not applicable
AH Maximum Data Lifetime	Not applicable
AH Maximum Size Limit	Not applicable
AH Data Lifetime Range	Not applicable
AH Size Limit Range	Not applicable
ESP Encapsulation Mode	Transport
ESP Authentication Algorithm	HMAC_MD5
ESP Encryption Algorithm	DES_CBC_8
ESP Maximum Data Lifetime	60
ESP Maximum Size Limit	0
ESP Data Lifetime Range	60-480
ESP Size Limit Range	0-0
Dynamic Tunnel Policy:	
Initiation	Either
Connection Lifetime	0
Authentication Information:	
Remote Key Server	172.16.3.3
Authentication Method	Pre-Shared Keys

VPN Parameter	Value
Shared Key	3132333435363738
Certificate Authority:	
Racdcert Label	Not applicable
Key Ring:	
User ID	Not applicable
Key Ring Name	Not applicable
Dynamic Connection:	
Source	192.168.100.100
Destination	172.16.3.3
Source Port	0
Destination Port	0
Automatic Activation	No
Protocol	All
Remote Key Server	172.16.3.3
Key Servers:	
Local Key Server ID Type	IPV4
Local Key Server ID	192.168.100.100
Remote Key Server ID Type	IPV4
Remote Key Server ID	172.16.3.3

Table 13. VPN planning worksheet - S/390 and Windows NT (SecureWay VPN Client)

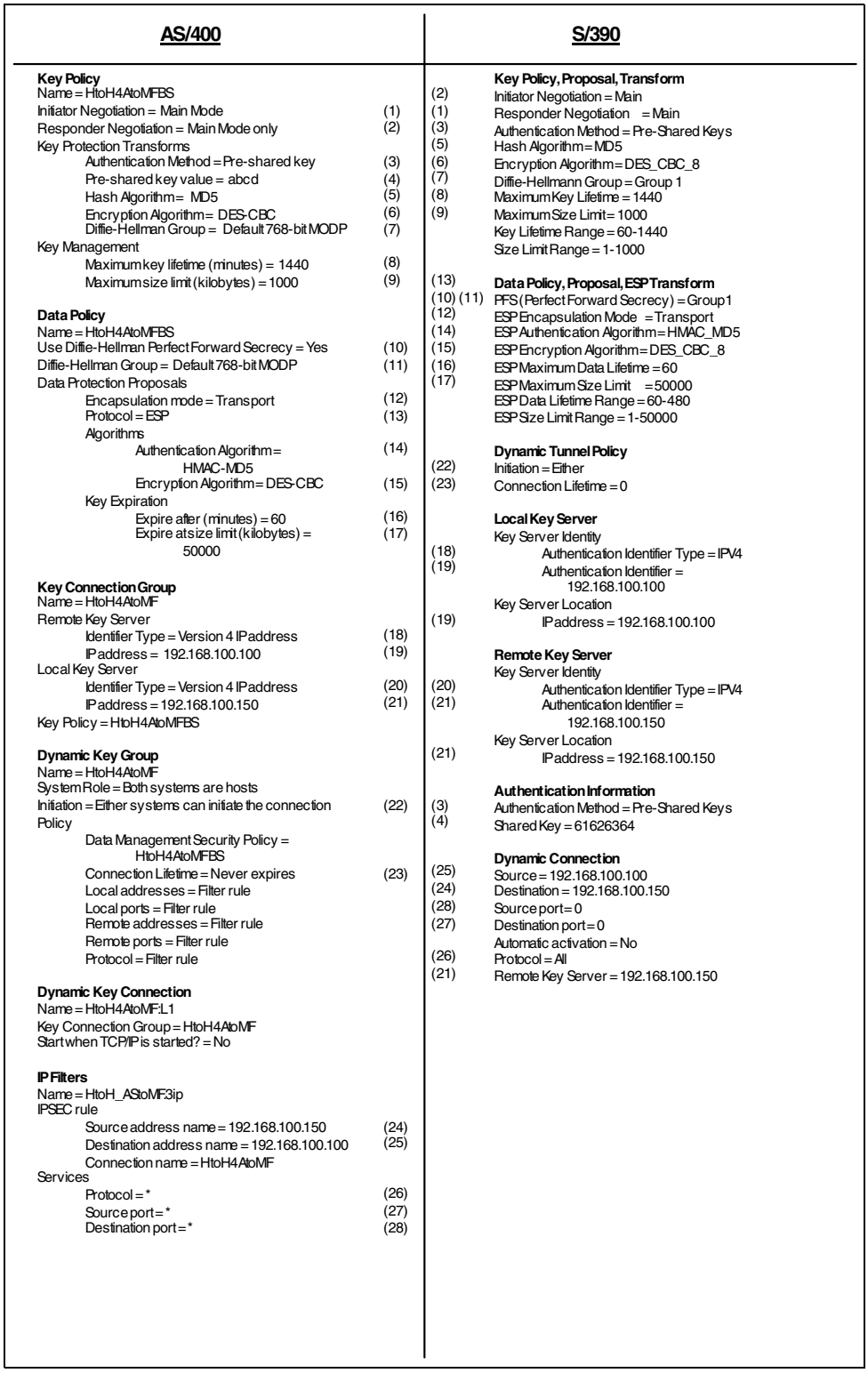
VPN Parameter	Value
Key Policy, Proposal, Transform:	
Initiator Negotiation	Main
Responder Negotiation	Main
Authentication Method	Pre-Shared Keys
Hash Algorithm	MD5
Encryption Algorithm	DES_CBC_8
Diffie-Hellman Group	Group 1
Maximum Key Lifetime	1440
Maximum Size Limit	0
Key Lifetime Range	1-1440
Size Limit Range	0-0

VPN Parameter	Value
Data Policy, Proposal, AH and ESP Transform:	
PFS (Perfect Forward Secrecy)	None
AH Encapsulation Mode	Not applicable
AH Authentication Algorithm	Not applicable
AH Maximum Data Lifetime	Not applicable
AH Maximum Size Limit	Not applicable
AH Data Lifetime Range	Not applicable
AH Size Limit Range	Not applicable
ESP Encapsulation Mode	Transport
ESP Authentication Algorithm	HMAC_MD5
ESP Encryption Algorithm	DES_CBC_8
ESP Maximum Data Lifetime	480
ESP Maximum Size Limit	0
ESP Data Lifetime Range	1-480
ESP Size Limit Range	0-0
Dynamic Tunnel Policy:	
Initiation	Either
Connection Lifetime	0
Authentication Information:	
Remote Key Server	192.168.100.7
Authentication Method	Pre-Shared Keys
Shared Key	3132333435363738
Certificate Authority:	
Racdcert Label	Not applicable
Key Ring:	
User ID	Not applicable
Key Ring Name	Not applicable
Dynamic Connection:	
Source	192.168.100.100
Destination	192.168.100.7
Source Port	0
Destination Port	0
Automatic Activation	No
Protocol	All

VPN Parameter	Value
Remote Key Server	192.168.100.7
Key Servers:	
Local Key Server ID Type	IPV4
Local Key Server ID	192.168.100.100
Remote Key Server ID Type	IPV4
Remote Key Server ID	192.168.100.7

Using the template in Appendix E, “VPN planning worksheet” on page 417, specify the information as part of the planning for your dynamic VPN tunnels. Create a worksheet for each TCP/IP stack you plan to configure with a dynamic tunnel.

Figure 345 on page 297 shows example of a cross-reference table utilized to match the parameters between the OS/390 and AS/400 systems.



AS/400		S/390	
Key Policy			**Key Policy, Proposal, Transform**
Name = HtoH4AtoMFBS		(2)	Initiator Negotiation = Main
Initiator Negotiation = Main Mode	(1)	(1)	Responder Negotiation = Main
Responder Negotiation = Main Mode only	(2)	(3)	Authentication Method = Pre-Shared Keys
Key Protection Transforms		(5)	Hash Algorithm = MD5
Authentication Method = Pre-shared key	(3)	(6)	Encryption Algorithm = DES_CBC_8
Pre-shared key value = abcd	(4)	(7)	Diffie-Hellmann Group = Group 1
Hash Algorithm = MD5	(5)	(8)	Maximum Key Lifetime = 1440
Encryption Algorithm = DES-CBC	(6)	(9)	Maximum Size Limit = 1000
Diffie-Hellman Group = Default 768-bit MODP	(7)		Key Lifetime Range = 60-1440
Key Management			Size Limit Range = 1-1000
Maximum key lifetime (minutes) = 1440	(8)		
Maximum size limit (kilobytes) = 1000	(9)	(13)	**Data Policy, Proposal, ESP Transform**
		(10) (11)	PFS (Perfect Forward Secrecy) = Group1
Data Policy		(12)	ESP Encapsulation Mode = Transport
Name = HtoH4AtoMFBS		(14)	ESP Authentication Algorithm = HMAC_MD5
Use Diffie-Hellman Perfect Forward Secrecy = Yes	(10)	(15)	ESP Encryption Algorithm = DES_CBC_8
Diffie-Hellman Group = Default 768-bit MODP	(11)	(16)	ESP Maximum Data Lifetime = 60
Data Protection Proposals		(17)	ESP Maximum Size Limit = 50000
Encapsulation mode = Transport	(12)		ESP Data Lifetime Range = 60-480
Protocol = ESP	(13)		ESP Size Limit Range = 1-50000
Algorithms			
Authentication Algorithm = HMAC-MD5	(14)		**Dynamic Tunnel Policy**
Encryption Algorithm = DES-CBC	(15)	(22)	Initiation = Either
Key Expiration		(23)	Connection Lifetime = 0
Expire after (minutes) = 60	(16)		
Expire at size limit (kilobytes) = 50000	(17)		**Local Key Server**
			Key Server Identity
Key Connection Group		(18)	Authentication Identifier Type = IPV4
Name = HtoH4AtoMF		(19)	Authentication Identifier = 192.168.100.100
Remote Key Server			Key Server Location
Identifier Type = Version 4 IP address	(18)	(19)	IP address = 192.168.100.100
IP address = 192.168.100.100	(19)		
Local Key Server			**Remote Key Server**
Identifier Type = Version 4 IP address	(20)		Key Server Identity
IP address = 192.168.100.150	(21)	(20)	Authentication Identifier Type = IPV4
Key Policy = HtoH4AtoMFBS		(21)	Authentication Identifier = 192.168.100.150
			Key Server Location
Dynamic Key Group		(21)	IP address = 192.168.100.150
Name = HtoH4AtoMF			
System Role = Both systems are hosts			**Authentication Information**
Initiation = Either systems can initiate the connection	(22)	(3)	Authentication Method = Pre-Shared Keys
Policy		(4)	Shared Key = 61626364
Data Management Security Policy = HtoH4AtoMFBS			
Connection Lifetime = Never expires	(23)		**Dynamic Connection**
Local addresses = Filter rule		(25)	Source = 192.168.100.100
Local ports = Filter rule		(24)	Destination = 192.168.100.150
Remote addresses = Filter rule		(28)	Source port = 0
Remote ports = Filter rule		(27)	Destination port = 0
Protocol = Filter rule			Automatic activation = No
		(26)	Protocol = All
Dynamic Key Connection		(21)	Remote Key Server = 192.168.100.150
Name = HtoH4AtoMF:L1			
Key Connection Group = HtoH4AtoMF			
Start when TCP/IP is started? = No			
IP Filters			
Name = HtoH_AStoMF3ip			
IPSEC rule			
Source address name = 192.168.100.150	(24)		
Destination address name = 192.168.100.100	(25)		
Connection name = HtoH4AtoMF			
Services			
Protocol = *	(26)		
Source port = *	(27)		
Destination port = *	(28)		

Figure 345. OS/390 and AS/400 system VPN configuration cross-reference table

These tables are very useful because the dynamic VPN definitions have many parameters. Using these tables can save a lot of time defining the VPN connection.

To create a dynamic VPN connection we have to create all the objects shown in Figure 346, and identified in the list following the figure.

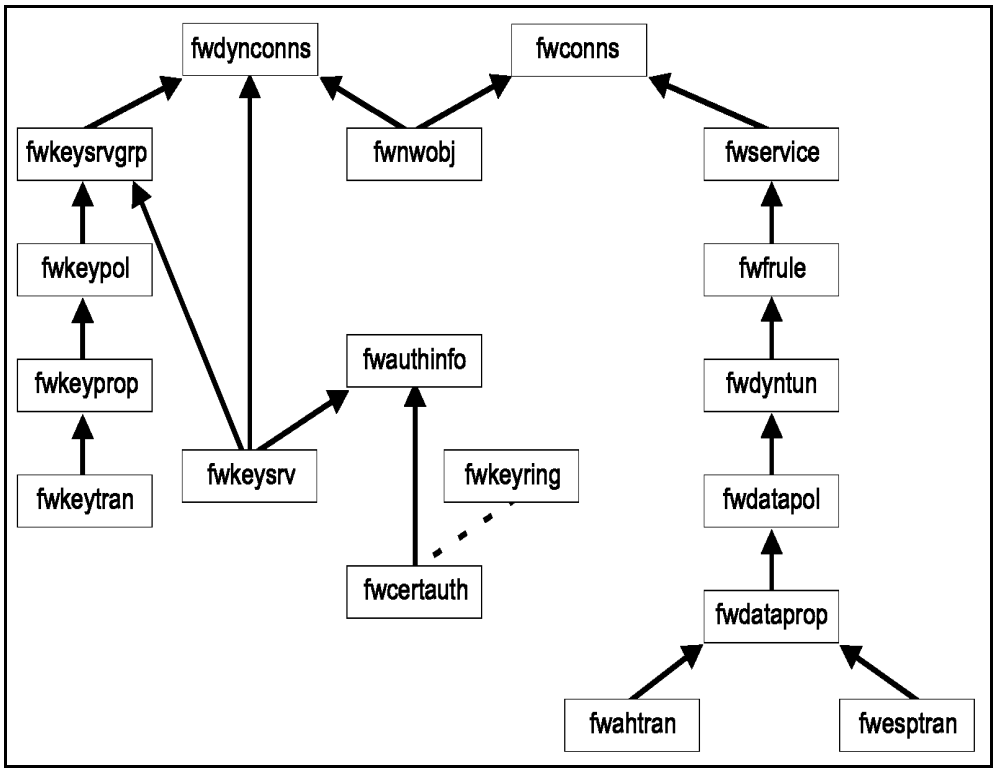


Figure 346. Firewall IKE objects relationship

- fwesptran: ESP transform object
- fwahtran: AH transform object
- fwdataprop: Data Proposal object
- fwdatapol: Data Policy object
- fwdyntun: Dynamic Tunnel Policy object
- fwfrule: Filter Rule object
- fwconns: Connection object
- fwnwobj: Network object
- fwdynconns: VPN Dynamic Connection object
- fwkeysrvgrp: Key Server Group object
- fwkeypol: Key Policy object
- fwkeyprop: Key Proposal object
- fwkeytran: Key Transform object
- fwkeysrv: Key Server object
- fwauthinfo: Authentication Information object
- fwcertauth: Certificate Authority object
- fwkeyring: Key Ring object

Figure 346 on page 298 shows the relationship between objects when you are defining a dynamic tunnel. Note that, in a particular configuration, not all objects have to be configured. For detailed information on these objects, consult *OS/390 Firewall Technologies Guide and Reference*, SC24-5835.

9.7.1 Creating a dynamic VPN connection using the GUI panels

Now we will use Table 11 on page 291 to create all VPN definitions that are necessary for these specific connections: OS/390 and OS/400.

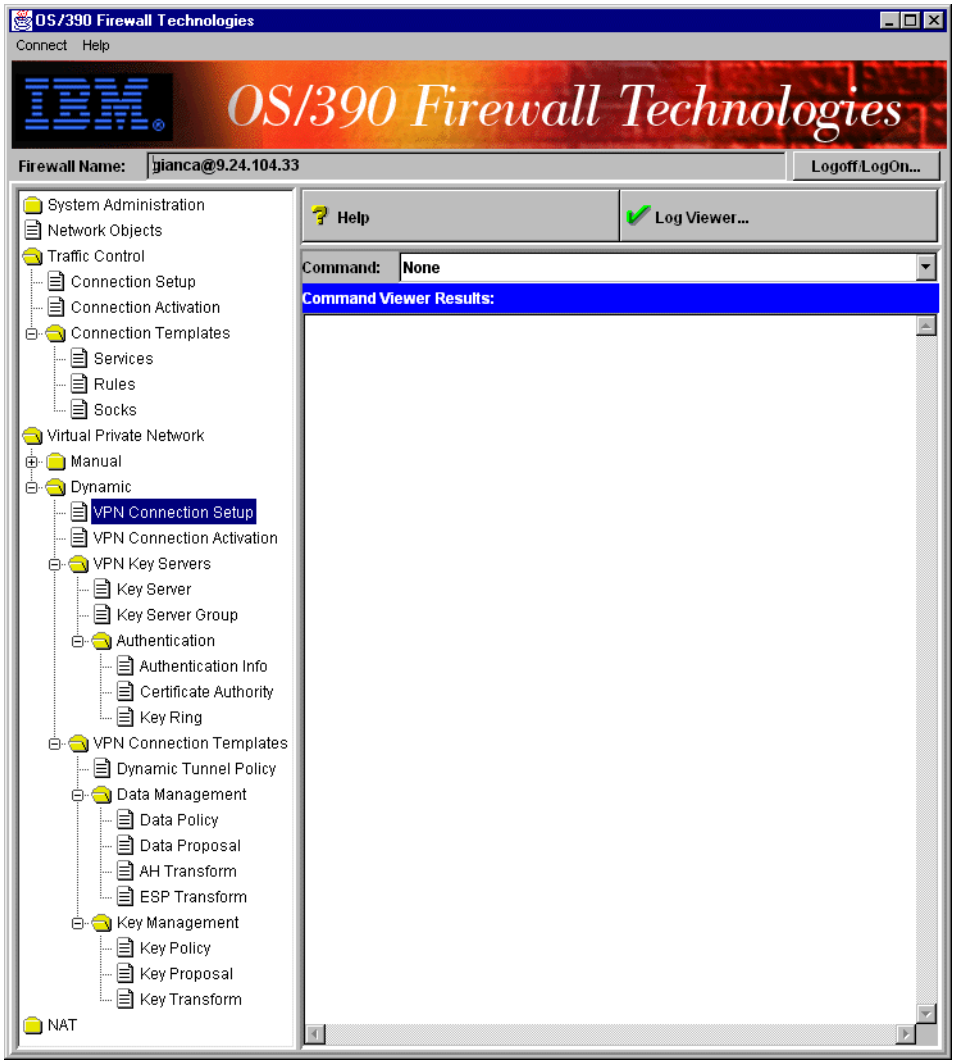


Figure 347. Firewall configuration client main screen: dynamic VPN definition

Log on to the configuration client. At the main screen double-click **Traffic Control**, then expand the **Connection Templates** tree. Double-click **Virtual Private Networks**; expand the **Dynamic** tree; expand the **VPN Key Servers** tree; expand the **Authentication** and **VPN Connection Templates** trees. Now expand the **Data Management** and **Key Management** trees. You will see the screen shown in Figure 347.

9.7.1.1 Key Management definition

We will start defining the Key Management object. We have to create a Key Transform object, and then associate it with a Key Proposal. Next we have to define a Key Policy using the Key Proposal. Do this with the following steps:

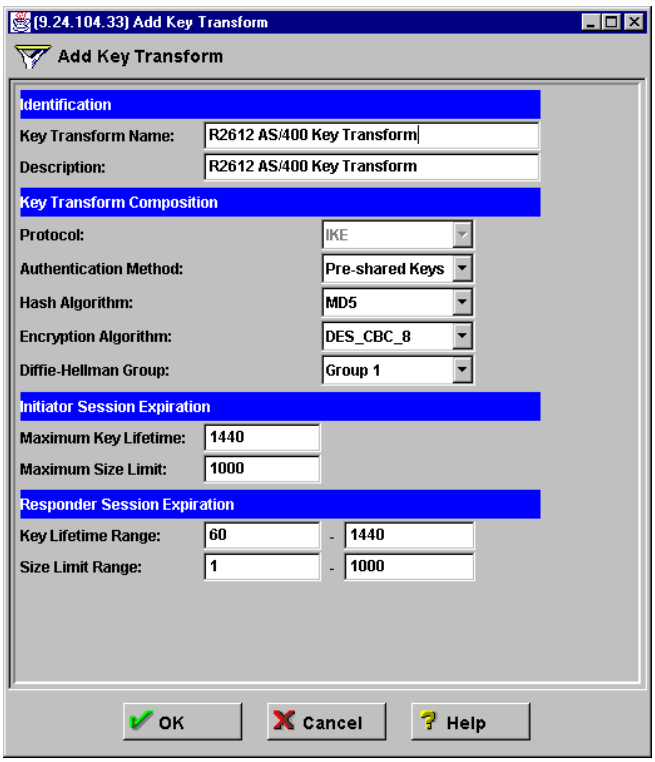


Figure 348. Adding Key Transform on OS/390

At the main screen double-click **Key Transform** and at the next screen double-click **NEW** to add a Key Transform. You will see the screen shown in Figure 348. Fill in the fields as shown, using values from Table 11 on page 291; click **OK**. Click **Close** and return to the main screen.

Key Transform

The key transform object defines the protection mechanisms used to secure subsequent exchanges between key servers. Key transforms specify how to use the Internet Key Exchange (IKE) protocol and include an authentication method and algorithm, a Diffie-Hellman group, and an encryption algorithm.

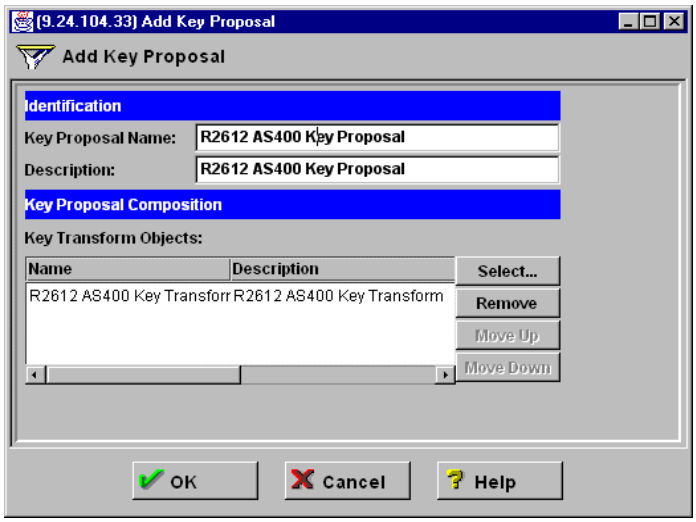


Figure 349. Adding Key Proposal on OS/390

At the main screen double-click **Key Proposal**, then double-click **NEW** to add a Key Proposal. You will see the screen shown in Figure 349. Click **Select ...** and choose the Key Transform created in Figure 348 on page 300. Fill in the fields as shown and click **OK**. Now click **Close** and return to the main screen.

Key Proposal

The key proposal object contains an ordered list of key transforms that will be proposed during key management negotiation. This ordering is important when acting as an initiator of a dynamic connection. In this case, the key transforms are sent to the remote key server in the initiator's order of preference as defined in the key proposal definition. When acting as responder, the initiator's ordering takes precedence.

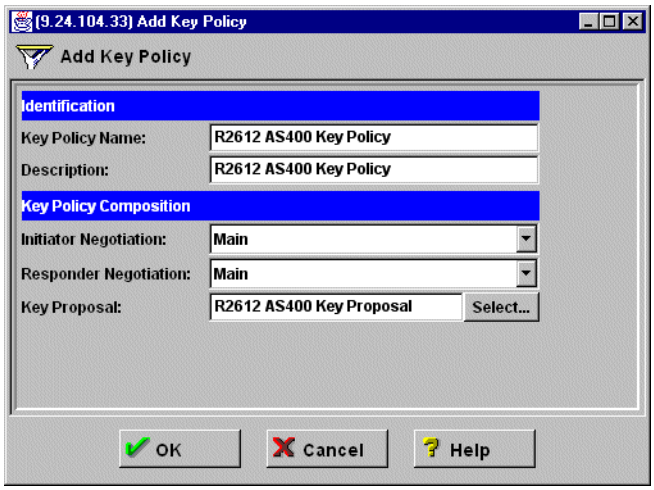


Figure 350. Adding Key Policy on OS/390

Now, at the main screen, double-click **Key Policy**, then double-click **NEW** to add a Key Policy. You will see the screen shown in Figure 350. Fill in the fields as shown, using values from Table 11 on page 291. Click **Select ...** and choose the

Key Proposal defined in Figure 349 on page 301. Then click **OK.** At the next screen click **Close** and return to the main screen.

Key Policy

The Key Policy object contains the information required when initiating or responding to a key management security negotiation. The Key Policy defines this system's initiator and responder negotiation modes and the key proposal.

9.7.1.2 Data Management definition

Now we will define the Data Management objects. We have to define an AH and ESP Transform object, associate them to a Data Proposal object, and then create a Data Policy using the Data Proposal object. Do this with the following steps:

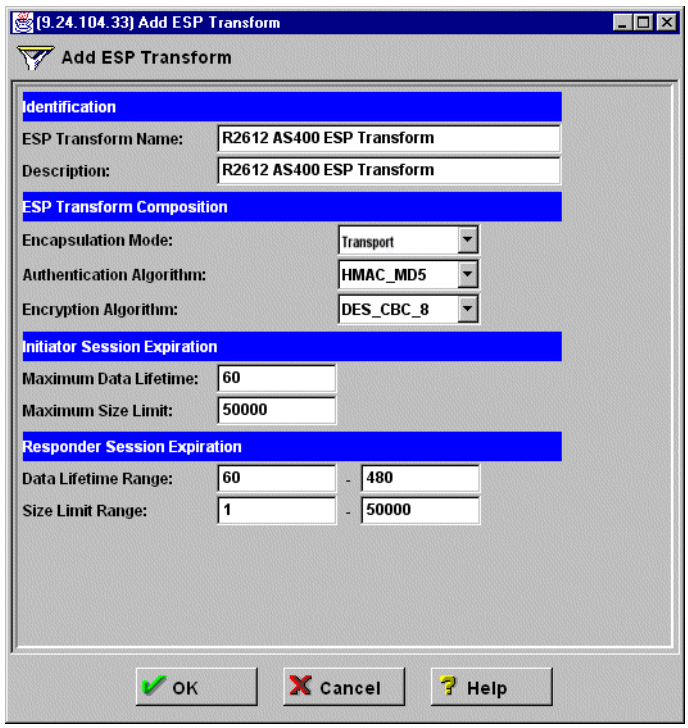


Figure 351. Adding ESP Transform on OS/390

At the main screen, double-click **ESP Transform**, then double-click **NEW** to add an ESP Transform. You will see the screen shown in Figure 351. Fill in the fields as shown, with values from Table 11 on page 291. Then click **OK**. At the next screen click **Close** and return to the main screen.

ESP Transform

The ESP Transform object defines protection mechanisms used to secure exchanges between the data endpoints through encryption and optionally with authentication.

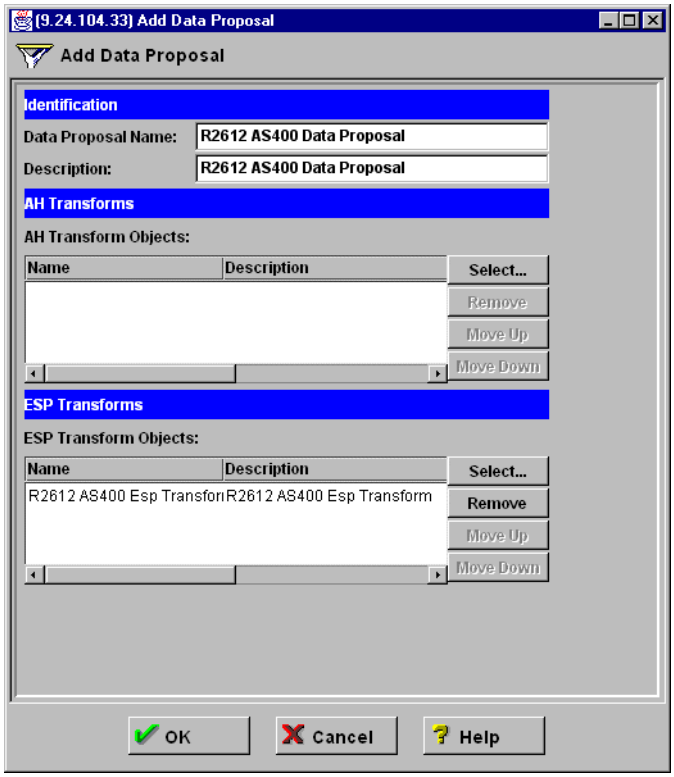


Figure 352. Adding Data Proposal on OS/390

At the main screen, double-click **Data Proposal**, then double-click **NEW** to add a Data Proposal. You will see the screen shown in Figure 352. Fill in the fields as shown. In the ESP Transforms section click **Select ...** and choose the ESP Transform created in Figure 351 on page 302. In this particular case we are not using AH Transform objects. Then click **OK**. At the next screen click **Close** and return to the main screen.

Data Proposal

The Data Proposal contains ordered lists of AH and ESP transforms used when negotiating a security association for data transmission. This ordering is important when acting as an initiator of a dynamic connection. In this case the ESP and AH transforms are sent to the remote key server in the initiator's order of preference as defined in the data proposal definition. When acting as responder, the initiator's ordering takes precedence.

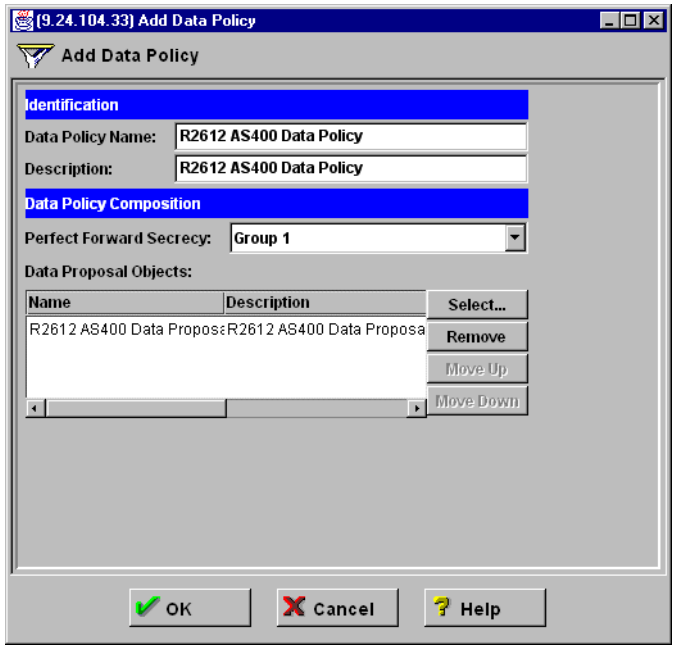


Figure 353. Adding Data Policy on OS/390

At the main screen, double-click **Data Policy**, then double-click **NEW** to add a Data Policy. You will see the screen shown in Figure 353. Fill in the fields as shown, using values from Table 11 on page 291. Click **Select ...** and choose the Data Proposal created in Figure 352 on page 303. Then click **OK.** At the next screen click **Close** and return to the main screen.

Data Policy

The Data Policy object defines information required when negotiating keys for data exchanges. This information includes the perfect forward secrecy selection and list of data proposals. The ordering is important when acting as an initiator of a dynamic connection. In this case, the policies are sent to the remote key server in the initiator's order of preference as defined in the data policy definition. When acting as responder, the initiator's ordering takes precedence.

9.7.1.3 Dynamic Tunnel Policy definition

The Dynamic Tunnel Policy object will be associated with an anchor filter Rule object. To define the Dynamic Tunnel Policy object complete the following steps:

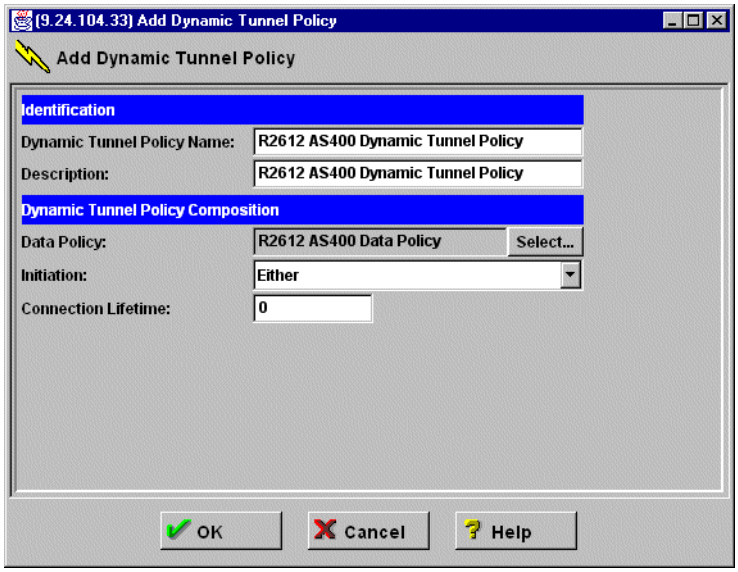


Figure 354. Adding Dynamic Tunnel Policy on OS/390

At the main screen, double-click **Dynamic Tunnel Policy**, then double-click **NEW** to add a Dynamic Tunnel Policy. You will see the screen shown in Figure 354. Fill in the fields as shown, using values from Table 11 on page 291. Click **Select ...** and choose the Data Policy created in Figure 353 on page 304. Then click **OK**. At the next screen click **Close** and return to the main screen.

Dynamic Tunnel Policy

The Dynamic Tunnel Policy object defines generic information relative to a set of tunnels. This information includes the Data Policy object, Initiation role, and Connection Lifetime to use. Dynamic Tunnel Policy objects will be specified in filter Rule objects.

9.7.1.4 Key Server definitions

Now we will define the Key Server objects. We have to define two Key Server objects: the Local Key Server (OS/390) and the Remote Key Server (AS/400). Then, we have to define a Key Server Group using the Key Server definitions. Complete the following steps:

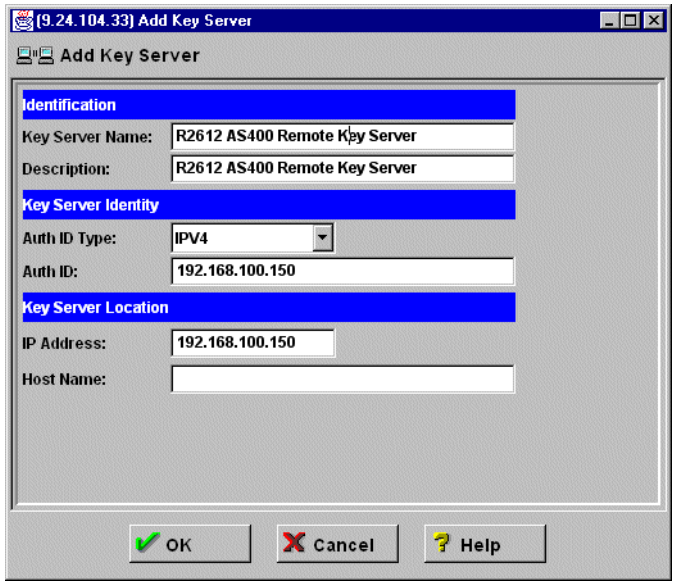


Figure 355. Adding Remote Key Server on OS/390

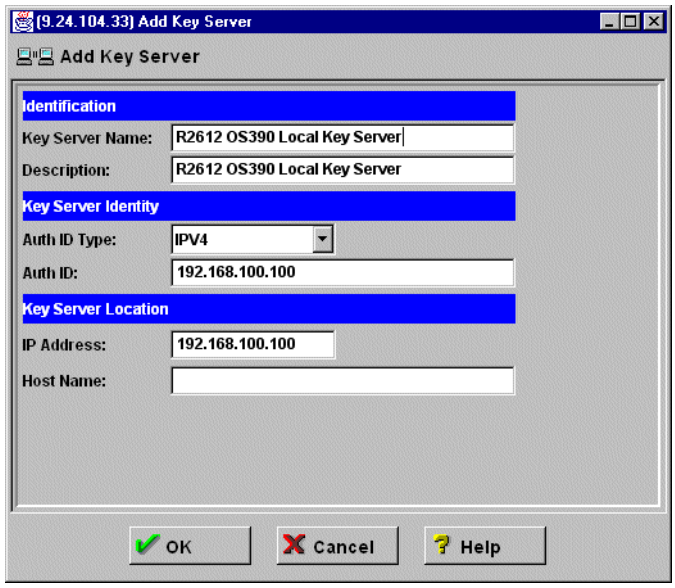


Figure 356. Adding Local Key Server on OS/390

At the main screen, double-click **Key Server**, then double-click **NEW** to add a Key Server. You will see the screen in Figure 355. Fill in the fields as shown in Table 11 on page 291 for the Remote Key Server definition, then click **OK**. Again, double-click **NEW** to add a Key Server. You will see the screen in Figure 356. Fill in the fields as shown in Table 11 on page 291 for the Local Key Server definition. Then click **OK.** At the next screen click **Close** and return to the main screen.

Key Server

The Key Server object defines information about key servers. Key servers negotiate security associations using the Internet Security Association Key Management Protocol (ISAKMP). A Key Server object defines an authentication identity by which a key server is known. A key server may have multiple identities.

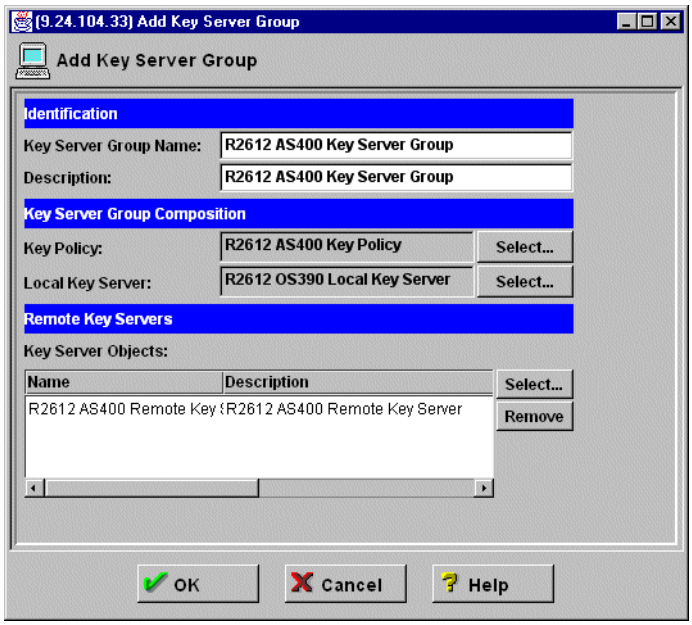


Figure 357. Adding Key Server Group on OS/390

At the main screen, double-click **Key Server Group**, then double-click **NEW** to add a Key Server Group. You will see the screen above. Click **Select ...** beside Key Policy and choose the Key Policy created in Figure 350 on page 301. Click **Select ...** beside Local Key Server and choose the Local Key Server created in Figure 356 on page 306. Click **Select ...** under Remote Key Server and choose the Remote Key Server created in Figure 355 on page 306. Then click **OK**. At the next screen click **Close** and return to the main screen.

Key Server Group

The Key Server Group object defines information about key server groups. A key server group is an association of a local key server with a list of remote key servers. This establishes which combinations of local and remote key servers can negotiate key management security associations and the key policy that will be used during these negotiations. Key server groups are ordered among themselves. Searches will find the first instance of the key server found in all key server group objects. Therefore, the key server groups must be ordered. Key server group objects will be specified in dynamic VPN connection objects.

9.7.1.5 Authentication Information definition

Here we will define the Authentication Information definition object. The Remote Key Server will be authenticated by following these steps:

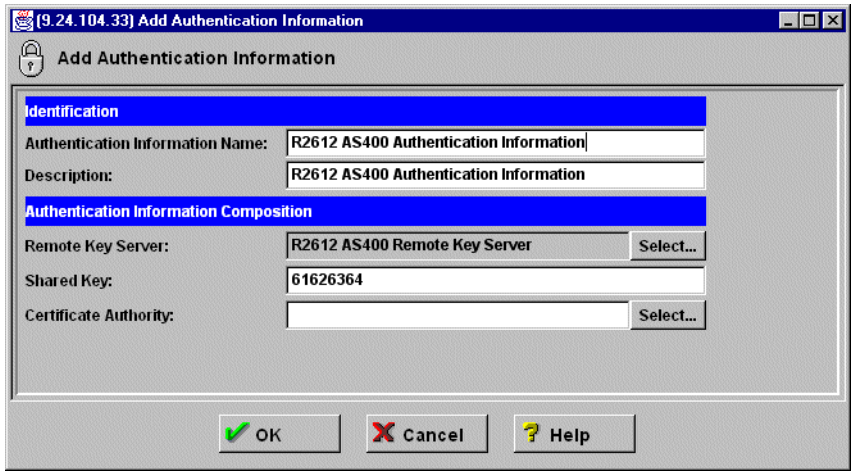


Figure 358. Adding Authentication Information on OS/390

At the main screen, double-click **Authentication Information**, then double-click **NEW** to add Authentication Information. You will see the screen shown in Figure 358. Fill in the fields as shown in Table 11 on page 291. Click **Select ...** and choose the Remote Key Server created in Figure 355 on page 306. Then click **OK.** At the next screen click **Close** and return to the main screen.

Note: The shared key must be entered as an even number of hex digits up to a length of 900 digits. The key 61626364 defined above is abcd in ASCII format.

Authentication Information

The Authentication Information object defines information required to authenticate the identity of the communicating peer. This information includes the remote key server object and the pre-shared key and/or certificate authority to use with it. Authentication information will be used when a dynamic connection is activated. The shared-key will always be typed in binary format (hexadecimal characters). The characters 61626364 in ASCII mean abcd.

9.7.1.6 Dynamic VPN connection definition

Now we will define the Dynamic VPN connection definition. We will associate the Network objects from both endpoints of the tunnel with the Remote Key Server and Key Server Group. Do this with the following steps:

We have to create two Network objects identifying the AS/400 and OS/390 systems:

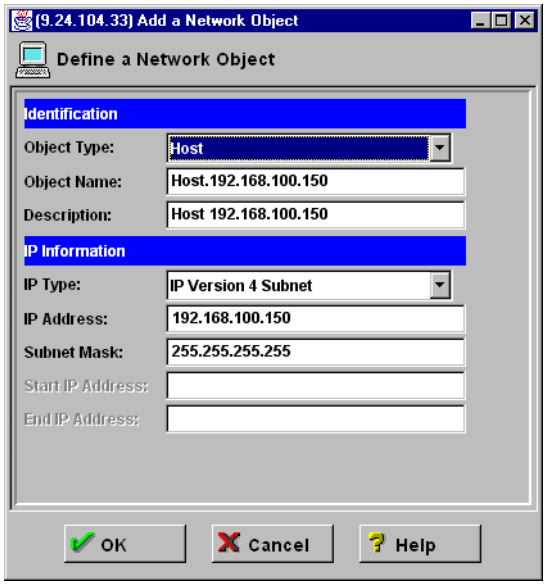


Figure 359. Adding Network object for AS/400

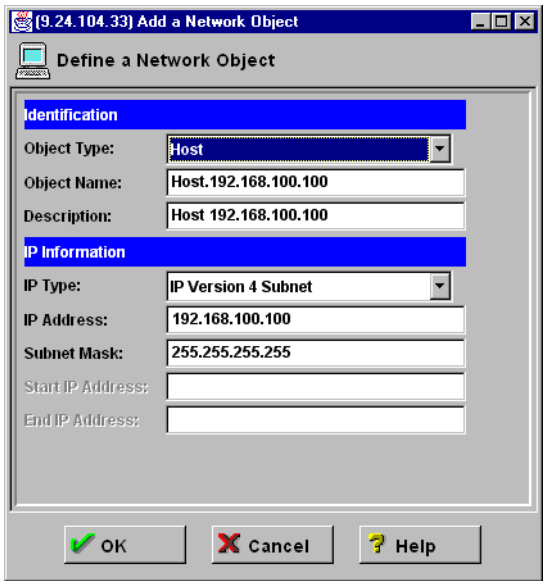


Figure 360. Adding Network object for OS/390

At the main screen, double-click **Network Objects**, then double-click **NEW** to add a Network object. You will see the screen in Figure 359. Fill in the fields as shown. Then click **OK**, then double-click **NEW** to add a Network object. You will see the screen in Figure 360. Fill in the fields as shown. Click **OK.** At the next screen click **Close** and return to the main screen.

Network Object

The Network Objects function allows you to maintain information about network addressable components on your network. This function acts as a central repository for use by other functions in the OS/390 system. Primarily, Network objects are used to designate source and destination addresses when you create your connections.

Now we will define the connection:

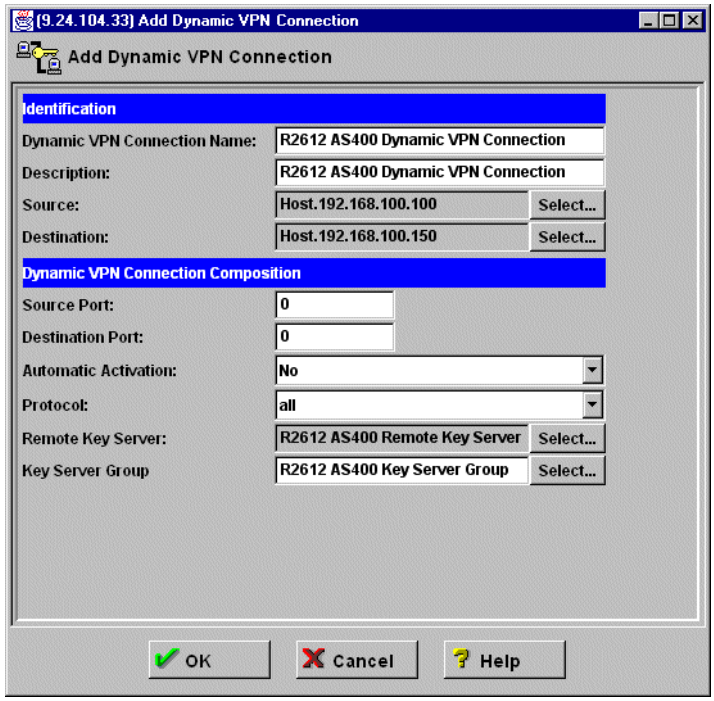


Figure 361. Adding Dynamic VPN connection on OS/390

At the main screen, double-click **VPN Connection Setup**, then double-click **NEW** to add a Dynamic VPN connection. You will see the screen shown in Figure 361. Fill in the fields as shown in Table 11 on page 291. Click **Select ...** beside Source and choose the OS/390 Network object created in Figure 360 on page 309. Click **Select ...** beside Destination and choose the AS/400 Network object created in Figure 359 on page 309. Click **Select ...** beside Remote Key Server and choose the Remote Key Server created in Figure 355 on page 306. Click **Select ...** beside Key Server Group and choose the Key Server Group created in Figure 357 on page 307. Then click **OK**. At the next screen click **Close** and return to the main screen.

Dynamic VPN Connection

The Dynamic VPN Connection object defines information that is used to activate a specific connection between data endpoints. This information includes the source and destination objects, source and destination ports, and protocol supported for this connection, along with the remote key server and key server group objects and an indicator of whether to auto-activate the connection.

9.7.1.7 Creating the anchor filter rules, services and connections

Now we have to create the anchor filter rule and the service definition associated with it to create the dynamic filter rules. Additionally, a connection definition which allows the tunnel endpoints to use AH and ESP protocol is required. Do this with the following steps:

In this anchor filter rule we will allow all types of traffic to flow in the tunnel. We will also associate this anchor filter rule with a Dynamic Tunnel Policy.

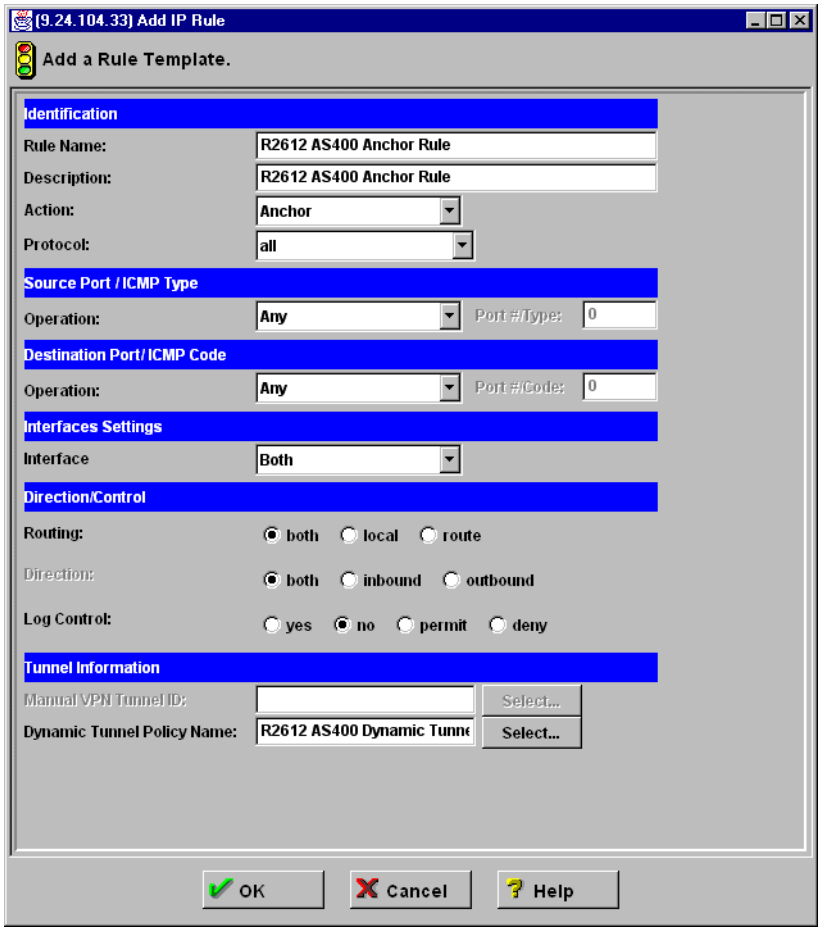


Figure 362. Creating anchor filter rule on OS/390

At the main screen, double-click **Connection Templates** and **Rules**, then double-click **NEW** to add a new Rule. You will see the screen shown in Figure 362. Fill in the fields as shown. Click **Select ...** beside Dynamic Tunnel Policy Name and choose the Dynamic Tunnel Policy created in Figure 354 on page 305. Then click **OK**. At the next screen click **Close** and return to the main screen.

Anchor filter rule

Rules on the OS/390 system are used to screen traffic passing through the system. Rules can be set up to either allow or disallow traffic on the basis of certain criteria. When a dynamic VPN connection is activated, the anchor filter rule is used to determine the placement of the dynamically generated filter rules among the static permit and deny rules.

We have to create a Services object to establish the anchor filter rule between the connection endpoints. Complete these steps:

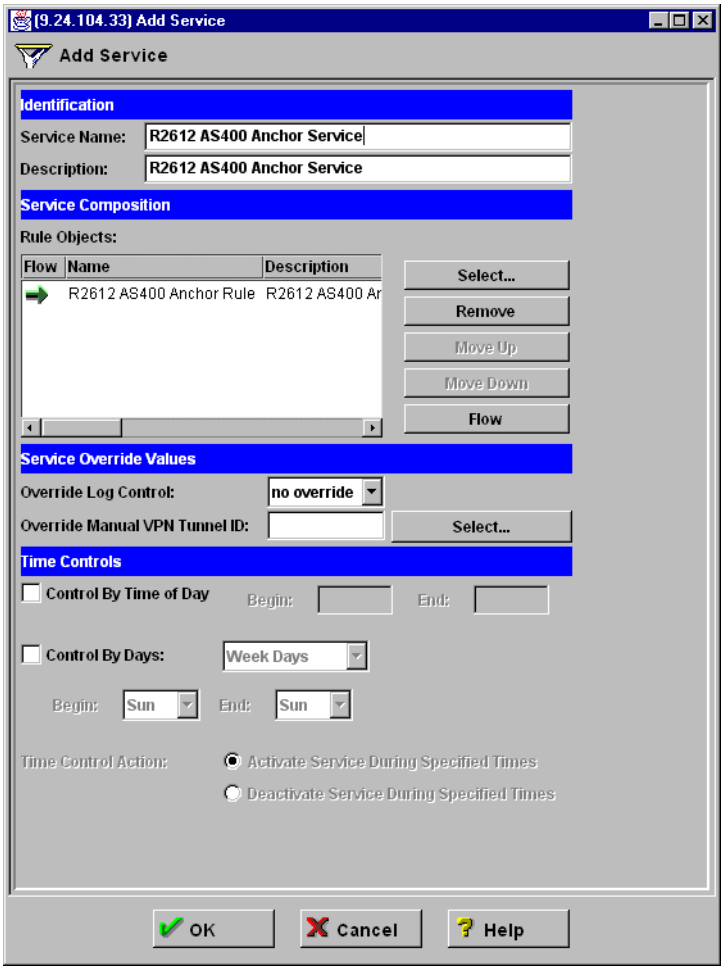


Figure 363. Adding Anchor Service on OS/390

At the main screen, double-click **Services**, then double-click **NEW** to add a new Service. You will see the screen shown in Figure 363. Fill in the fields as shown. Click **Select ...** under Rule Objects and choose the anchor filter rule created in Figure 362 on page 311. Then click **OK**.

Services

Services is a collection of rules or a set of instructions to permit or deny a particular type of traffic through the OS/390 system, for example, a Telnet session. Services figures prominently when defining connections, specifying the type of traffic that can or cannot take place between Network objects.

Now we will create the two connections to associate the services between the tunnel endpoints. Then we will activate the connection to create the filter rules. Do this with the following instructions:

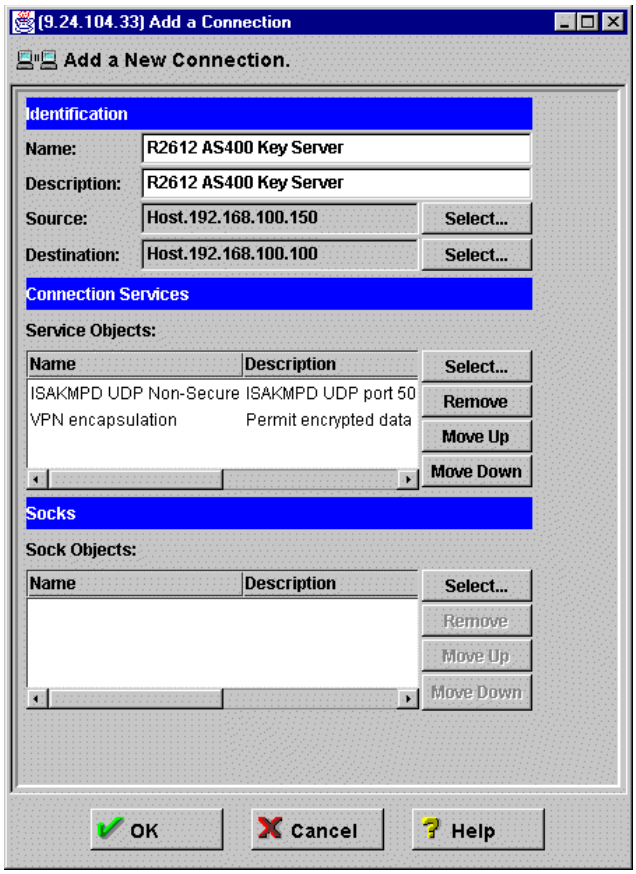


Figure 364. Adding Key Server Connection on OS/390

At the main screen, double-click **Connection Setup**, then double-click **NEW** to add a new Connection. You will see the screen shown in Figure 364. Fill in the fields as shown. Click **Select ...** beside Source and choose the Network object created in Figure 359 on page 309. Click **Select ...** beside Destination and choose the Network object created in Figure 360 on page 309. Then click **OK**.

The objects selected as the service object are predefined Service objects. These attributes are shown in Figure 365 and Figure 366 on page 314.

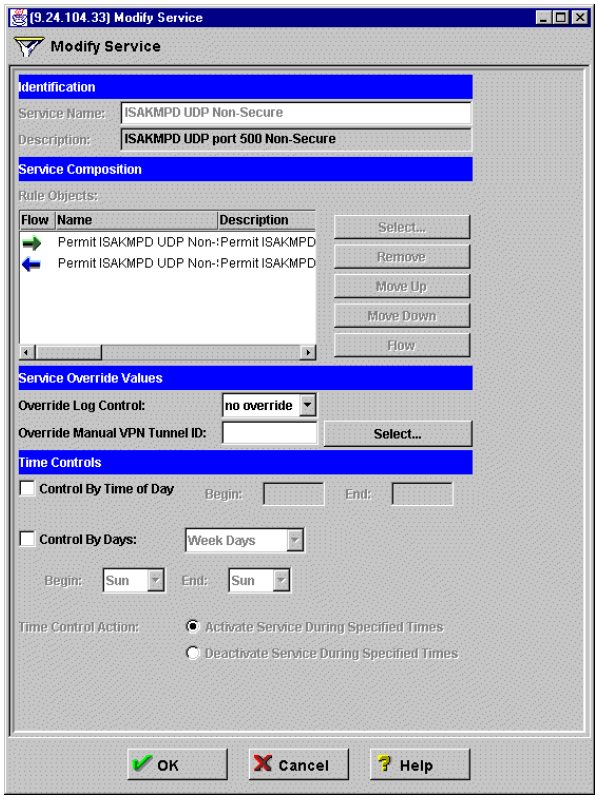


Figure 365. ISAKMPD UDP Non-Secure Service object

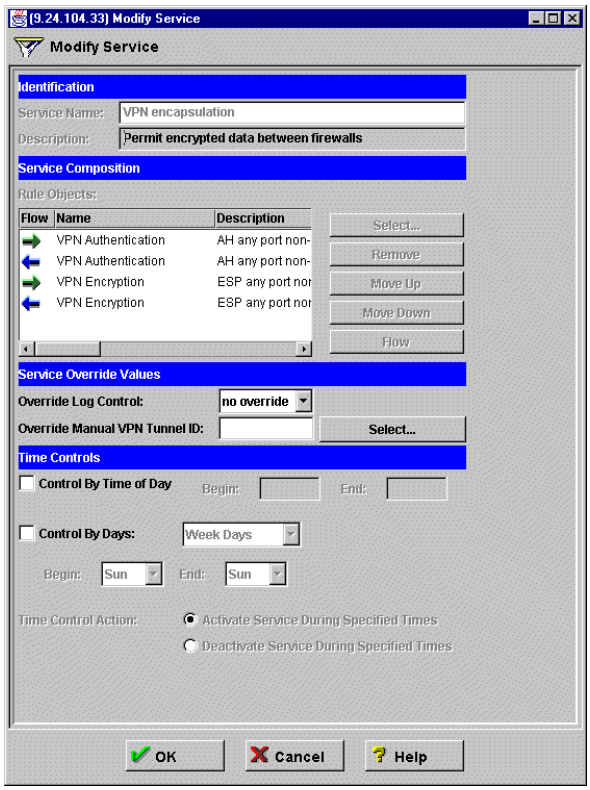


Figure 366. VPN encapsulation Service object

Next we define the connection which sets up the anchor filter rules for connection endpoints.

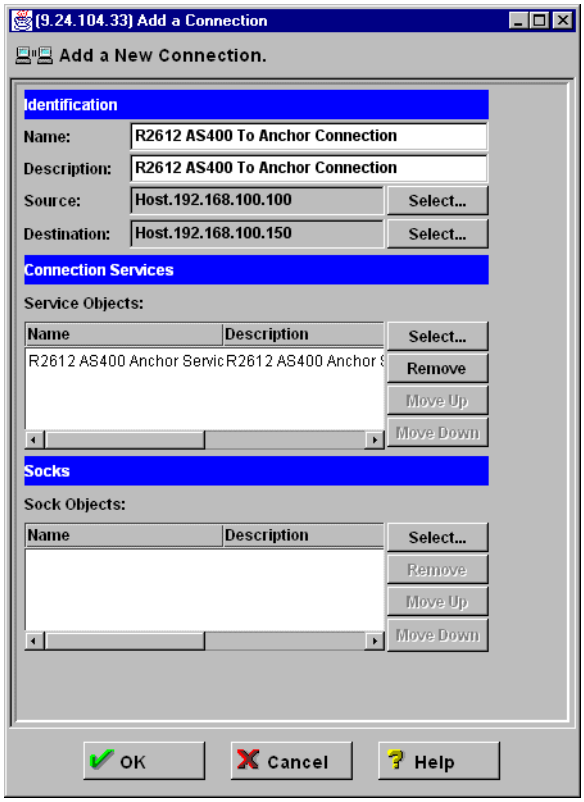


Figure 367. Adding Anchor Connection on OS/390

Double-click **NEW** to add a new Connection. You will see the screen shown in Figure 367. Fill in the fields as shown. Click **Select ...** beside Source and choose the Network object created in Figure 360 on page 309. Click **Select ...** beside Destination and choose the Network object created in Figure 359 on page 309. Then click **OK**.

Since an anchor filter rule generates both an inbound and outbound rule, you do not need have two connections that are defined for each direction.

Connections

The Connection function allows you to control the type of network traffic that can take place between two network entities that are connected through the OS/390 system. They permit or deny specified types of communications between two named endpoints or any type of Network object or group. After you have defined your Network objects and services, you create connections. In building connections, you will select one Network object to be the source and another Network object to be the destination for the traffic flow through the OS/390 system.

Note: The connection must be ordered in the following sequence: Key Server Connection and Anchor Connection. Check this in the Connections List screen and reorder the connection if necessary. Access HELP from the Connection Setup for more information.

9.7.1.8 Activating the filter rules and the dynamic VPN connection

Now we will activate the filter rules and the dynamic VPN connection. Do this with the following instructions:

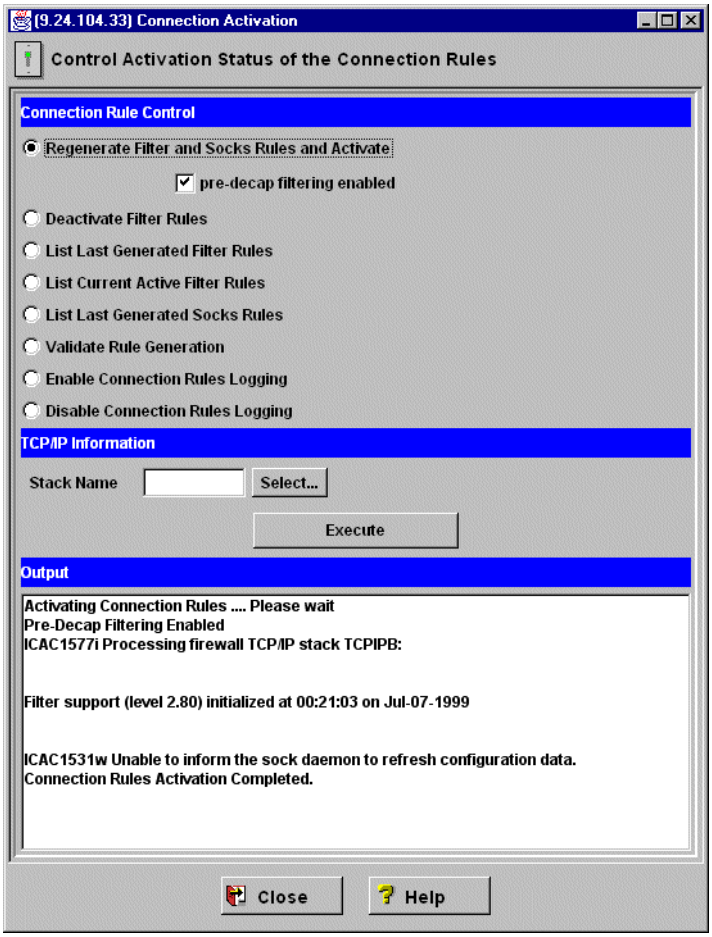


Figure 368. Connection Activation

At the main screen, double-click **Connection Activation**, then check the boxes shown in Figure 368 and click **Execute**. Check the messages in the output area for errors. Then click **Close**.

Connection Activation

Use this function to generate the rules based upon the configurations defined in the connection setup panel and all of its subsidiary configurations (for example, services, rule templates, and SOCKS templates). Rules can also be generated depending upon settings in the security policy panel. These rules become the active set through which the OS/390 system can evaluate network datagrams. If there is a set of connection rules already active, this procedure updates the active rules with the contents of the newly generated set. Feedback about a successful activation or any errors will be displayed in the output section.
Note: Checking the pre-decap filtering enabled check box results in AH and ESP packets being filtered before they are decapsulated. If there are concerns about AH and ESP packets in your network, then you may want to have the AH and ESP packets filtered before they are decapsulated.

Now we will activate the dynamic VPN connection using the following steps:

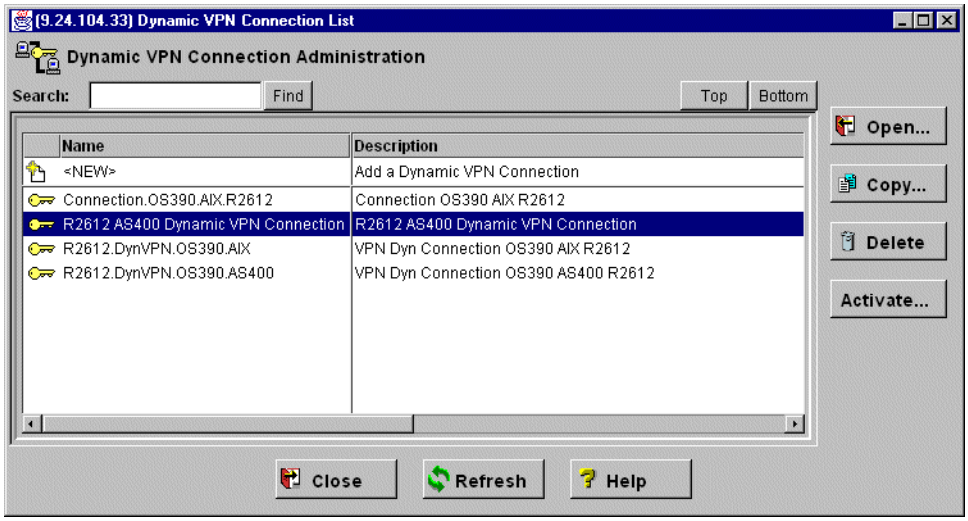


Figure 369. Dynamic VPN connection activation

At the main screen, double-click **VPN Connection Setup**, then choose the Dynamic VPN Connection created in Figure 361 on page 310 and click **Activate...**. You will receive the following message:

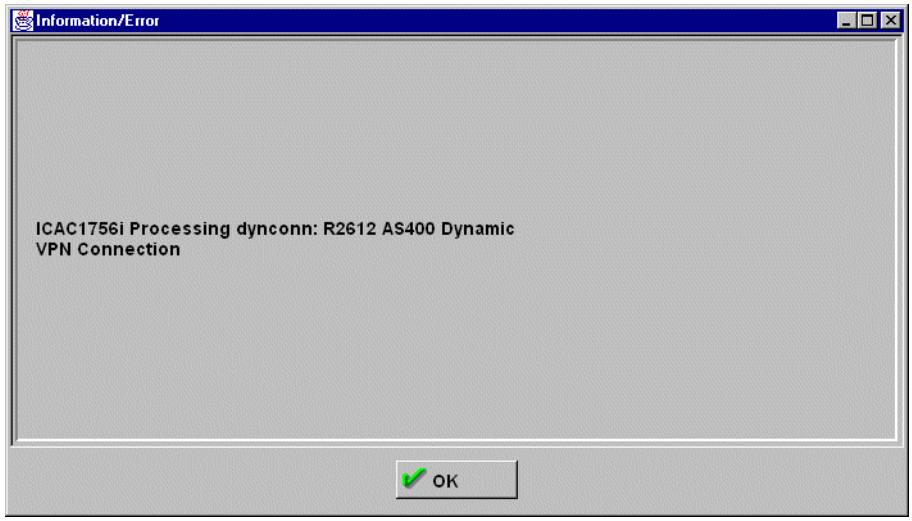


Figure 370. Dynamic connection activation message

This message only tells you that the activation of the dynamic VPN connections is in progress. You have to check the VPN connection Activation screen to see if the tunnel was activated. Follow these steps:

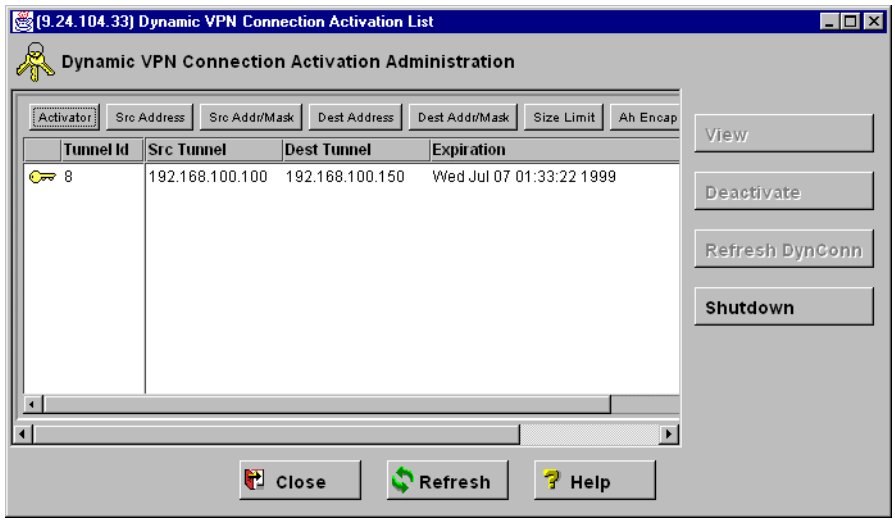


Figure 371. Dynamic Connection Activation List

To check all the parameters in the tunnel, double-click the tunnel, or select the tunnel and click **View**. You will receive the following screen showing all the definitions:

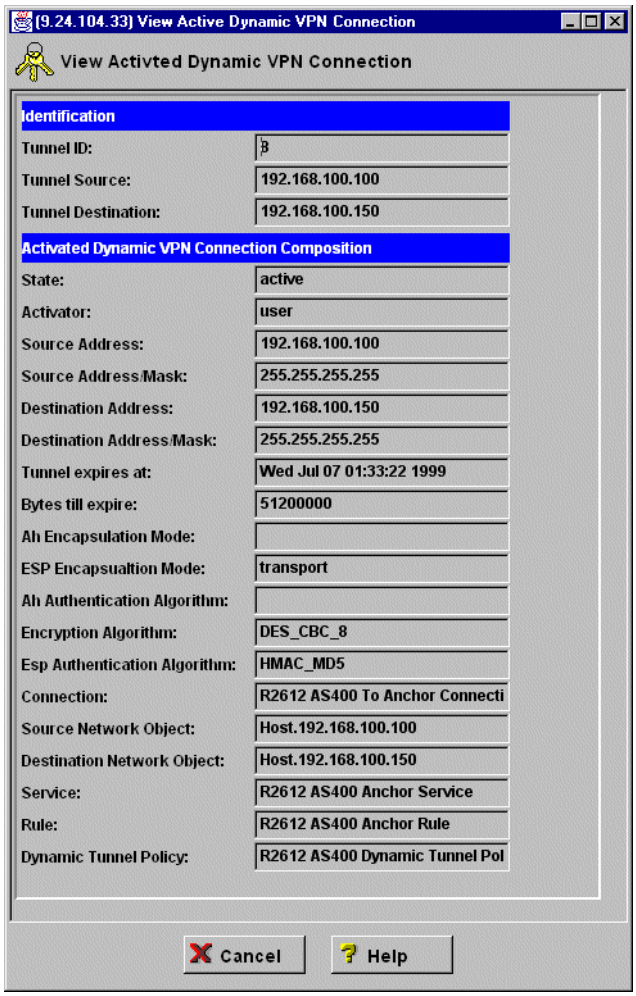


Figure 372. View Activated Dynamic VPN Connection

Now you can start the traffic between the two hosts. All types of traffic will be permitted.

9.7.1.9 Using the firewall log to check the tunnel status

There are some messages in the firewall log that can help you check the activation of the tunnel, if the traffic is flowing through the tunnel, and so on. You can browse the log file using the OBROWSE command in a TSO session or using the Log Viewer in the configuration client. These messages are in the file /tmp/firewall.all.log as shown in Figure 336 on page 286.

```
ICA8233i;507;510; 1
ICA8227i;0000000008;192.168.100.100;255.255.255.255;ALL;ALL;192.168.100.150; 2
ICA1073i;TCPIPB;R:p; o:;192.168.100.100;s:;192.168.100.100;d:;192.168.100.150; 3
p:;icmp;t:;8;c:;0;r:;l;a:;n;f:;y;T:;0000000512:0000000507:0000000510:0000000510:
00000510:0000000510:0000000530:0000000501:0000000008"†"*;AH:;0;ESP:;0;l:;284;
ICA1073i;TCPIPB;R:p; i:;192.168.100.100;s:;192.168.100.150;d:;192.168.100.100; 4
p:;icmp;t:;0;c:;0;r:;l;a:;n;f:;y;T:;0000000512:0000000507:0000000510:0000000510:
000000510:0000000510:0000000530:0000000501:0000000008"†"*;AH:;0;ESP:;0;l:;284;
```

Figure 373. Checking firewall log messages

Figure 373 notes:

1 This message indicates that an attempt to create a dynamic connection between the two Network objects is in progress. Checking these two Network objects in the UNIX System Services shell you will see:

```
GIANCA @ RA03:/u/gianca>fwnwobj cmd=list id=507 format=long
               id = 507
             type = Host
             name = Host.192.168.100.100
             desc = Host 192.168.100.100
             addr = 192.168.100.100
             mask = 255.255.255.255
        startaddr =
          endaddr =

GIANCA @ RA03:/u/gianca>fwnwobj cmd=list id=510 format=long
               id = 510
             type = Host
             name = Host.192.168.100.150
             desc = Host 192.168.100.150
             addr = 192.168.100.150
             mask = 255.255.255.255
        startaddr =
          endaddr =
```

Figure 374. Displaying tunnel endpoints Network objects

The two Network objects are the tunnel endpoints.

2 This message indicates that tunnel ID 8 was created.

3, 4 These messages show a PING (ICMP protocol) between the tunnel endpoints. Look tat the first ICMP message is from the OS/390 host and the second ICMP message is from the AS/400 host. These messages also indicate that the traffic is flowing through tunnel ID 8.

See *OS/390 Firewall Technologies Guide and Reference*, SC24-5835 for a more detailed explanation.

9.7.2 Creating a dynamic VPN using the shell commands

Now we will use the shell commands to define the tunnel based on Table 12 on page 293, between OS/390 and AIX/6000. You can either create a script and execute the script file or execute the commands from an OVMS shell prompt.

First we define the Key Management objects: Key Transform, Key Proposal, and a Key Policy. A Key Policy contains a Key Proposal that contains one or more Key Transform objects.

```
fwkeytran cmd=add    name="R2612 AIX Key Transform" \
                     desc="R2612 AIX Key Transform" \
                     prot=IKE authmeth=SHAREDKEY \
                     hashalg=MD5 encralg=DES_CBC_8 dhgrp=GROUP1 \
                     itime=1440 isize=0 rtime=60-1440 rsize=0-0
fwkeyprop cmd=create name="R2612 AIX Key Proposal" \
                     desc="R2612 AIX Key Proposal" \
                     keytranlist="R2612 AIX Key Transform"
fwkeypol  cmd=add    name="R2612 AIX Key Policy" \
                     desc="R2612 AIX Key Policy" \
                     imode=MAIN rmode=MAIN \
                     keyprop="R2612 AIX Key Proposal"
```

Figure 375. Key Management definition

Next we define the Data Management objects: ESP Transform, Data Proposal, and Data Policy. A data policy contains one or more Data Proposal objects, each of which contains one or more ESP and AH Transform objects.

```
fwesptran cmd=add     name="R2612 AIX Esp Transform" \
                      desc="R2612 AIX Esp Transform" \
                      mode=TRANSPORT \
                      authalg=HMAC_MD5 encralg=DES_CBC_8 \
                      itime=60 isize=0 rtime=60-480 rsize=0-0
fwdataprop cmd=create name="R2612 AIX Data Proposal" \
                      desc="R2612 AIX Data Proposal" \
                      esptranlist="R2612 AIX Esp Transform"
fwdatapol cmd=create  name="R2612 AIX Data Policy" \
                      desc="R2612 AIX Data Policy" \
                      pfs=NONE \
                      dataproplist="R2612 AIX Data Proposal"
```

Figure 376. Data Management definition

Now we define the Dynamic Tunnel Policy. The Dynamic Tunnel Policy will be associated with an anchor filter rule and will be used to set attributes for a dynamic tunnel.

```
fwdyntun  cmd=add     name="R2612 AIX Dynamic Tunnel Policy" \
                      desc="R2612 AIX Dynamic Tunnel Policy" \
                      datapol="R2612 AIX Data Policy" \
                      init=EITHER connlifetime=0
```

Figure 377. Dynamic Tunnel Policy definition

Next we define the Key Server objects and the Key Server Group object. The Key Servers objects define the endpoints of a VPN connection. The Local Key Server for this connection will be the same as defined in Figure 356 on page 306.

```
fwkeysrv  cmd=add      name="R2612 AIX Remote Key Server" \
                       desc="R2612 AIX Remote Key Server" \
                       idtype=IPV4 \
                       authid=172.16.3.3 ipaddr=172.16.3.3
keysrvgrp cmd=create name="R2612 AIX Key Server Group" \
                       desc="R2612 AIX Key Server Group" \
                       keypol="R2612 AIX Key Policy" \
                       lockeysrv="R2612 OS390 Local Key Server" \
                       remkeysrvlist="R2612 AIX Remote Key Server"
```

Figure 378. Key Server and Key Server Group definition

Now we define the Authentication Information. This object is used to authenticate the key servers when you start a dynamic connection. You can use either Pre-Shared Keys authentication or RSA Signature (certificate based). The Shared-Key is always defined using a binary format. The characters `3132333435363738` in ASCII means `12345678`.

```
fwauthinfo cmd=add     name="R2612 AIX Authentication Information" \
                       desc="R2612 AIX Authentication Information" \
                       remkeysrv="R2612 AIX Remote Key Server" \
                       shkey=3132333435363738
```

Figure 379. Authentication Information definition

Next we define an anchor filter Rule object and two Service objects. The anchor filter rule needs to be associated with a Dynamic Tunnel Policy object that allows dynamic connections to be created.

```
$ fwdyntun cmd=LIST dyntun="R2612 AIX Dynamic Tunnel Policy"
               id = 504   1
             name = R2612 AIX Dynamic Tunnel Policy
             desc = R2612 AIX Dynamic Tunnel Policy
          datapol = 504
             init = either
     connlifetime = 0

$ fwfrule cmd=ADD \
          name="R2612 AIX Anchor Rule" \
          desc="R2612 AIX Anchor Rule" \
          type=ANCHOR protocol=ALL \
          srcopcode=ANY srcport=0 \
          destopcode=ANY destport=0 \
          interface=BOTH routing=BOTH log=YES \
          tunnel=504     1
```

Figure 380. Anchor filter rule definition

Figure 380 note:

1 You have to list the Dynamic Tunnel Policy to get the ID. This parameter will be used in the `fwfrule` command in the tunnel parameter.

Then we create two Service objects: one service establishes the anchor filter rule between the connection endpoints and the other establishes the permit filter rules for the Key Server traffic.

```
$ fwfrule  cmd=LIST name="R2612 AIX Anchor Rule"
              id = 531        1
            type = anchor
            name = R2612 AIX Anchor Rule
            desc = R2612 AIX Anchor Rule
        protocol = all
       srcopcode = any
         srcport = 0
      destopcode = any
        destport = 0
       interface = both
         routing = both
             log = yes
          tunnel = 504

$ fwservice cmd=CREATE \
         name="R2612 AIX Anchor Service" \
         desc="R2612 AIX Anchor Service" \
         rulelist=531/f  1

$ fwservice cmd=CREATE \
          name="R2612 AIX Key Server Service" \
          desc="R2612 AIX Key Server Service" \
          rulelist=136/b,136/f,5/f,11/f  2

$ fwservice cmd=LIST name="R2612 AIX Anchor Service"   3
              id = 512      3
            name = R2612 AIX Anchor Service
            desc = R2612 AIX Anchor Service
        rulelist = 531/f
             log = yes
        fragment =
          tunnel =
            time =
           month =
             day =
         weekday =
      timefilter =

$ fwservice cmd=LIST name="R2612 AIX Key Server Service" 3
              id = 513   3
            name = R2612 AIX Key Server Service
            desc = R2612 AIX Key Server Service
        rulelist = 136/f,136/b,5/f,11/f
             log = yes
        fragment =
          tunnel =
            time =
           month =
             day =
         weekday =
      timefilter =
```

Figure 381. Services definition

Figure 381 notes:

1 List the anchor filter rule to get its ID. It will be used to create the Anchor Service.

2 These rules are predefined. Rule 136 allows ISAKMPD traffic (UPD 500) in a nonsecure interface; rule 5 allows VPN Authentication, and rule 11 allows VPN Encryption.

3 List both services you have created and get their IDs. They will be used in the Connection definition.

```
$ fwnwobj cmd=ADD  name=Host.172.16.3.3 desc="Host 172.16.3.3" \
         type=HOST addr=172.16.3.3 mask=255.255.255.255

$ fwnwobj  cmd=list name=Host.172.16.3.3 1
512   Host         Host.172.16.3.3  Host 172.16.3.3

$ fwnwobj  cmd=list name=Host.192.168.100.100  1
507   Host         Host.192.168.100 Host 192.168.100.100

$ fwconns  cmd=CREATE name="R2612 AIX Key Server" \
         desc="R2612 AIX Key Server" \
         source=Host.172.16.3.3 \
         destination=Host.192.168.100.100 \
         servicelist=513

$ fwconns  cmd=CREATE
         name="R2612 AIX Anchor Connection" \
         desc="R2612 AIX Anchor Connection" \
         source=Host.192.168.100.100 \
         destination=Host.172.16.3.3 \
         servicelist=507

$ fwconns  cmd=MOVE \   2
         connection="R2612 AIX Anchor Connection" \
         after="R2612 AIX Key Server"
```

Figure 382. Network objects and Connection definition

Figure 382 notes:

1 List the two Network objects to get their IDs. They will be used to define the connections.

2 The Key Server Connection must be in front of the Anchor Connection. To be sure, move the two Anchor Connections after the Key Server Connection.

Now we create the Dynamic VPN connection. The Dynamic VPN connection object defines information that is used to activate a specific connection between data endpoints.

```
$ fwdynconns cmd=ADD \
             name="R2612 AIX Dynamic VPN Connection" \
             desc="R2612 AIX Dynamic VPN Connection" \
             src=Host.192.168.100.100 \
             dest=Host.172.16.3.3 \
             srcport=0 destport=0 prot=ALL \
             remkeysrv="R2612 AIX Remote Key Server"
             keysrvgrp="R2612 AIX Key Server Group"
```

Figure 383. Dynamic VPN connection definition

Finally, we can activate and check if the Dynamic VPN connection is active.

```
GIANCA @ RA03:/u/gianca/firewall>fwdynconns cmd=activate dynconnlist="R2612 AIX
Dynamic VPN Connection"
ICAC1756i Processing dynconn: R2612 AIX Dynamic VPN Connection
GIANCA @ RA03:/u/gianca/firewall>fwdynconns cmd=listactive
11         192.168.100.100 192.168.100.7   active  user
12         192.168.100.100 172.16.3.3      active  user
GIANCA @ RA03:/u/gianca/firewall>fwdynconns cmd=listactive tunlist=12 format=lon
g
         TunnelID = 12
           TunSrc = 192.168.100.100
          TunDest = 172.16.3.3
            State = active
        Activator = user
         SrcAddr1 = 192.168.100.100
         SrcAddr2 = 255.255.255.255
        DestAddr1 = 172.16.3.3
        DestAddr2 = 255.255.255.255
Tunnel expires at = Wed Jul 07 18:57:41 1999 EDT
Bytes till expire = 0
    AH Encap Mode =
   ESP Encap Mode = transport
      AH Auth Alg =
      Encrypt Alg = DES_CBC_8
     ESP Auth Alg = HMAC_MD5
           ConnID = 509
         SrcObjID = 507
        DestObjID = 512
            SvcID = 512
          FruleID = 531
         DynTunID = 504
```

Figure 384. Activating and checking the connection status

Chapter 10. Enabling SSL on Telnet

This chapter focuses on the functional enhancements for the Telnet server shipped with SecureWay Communications Server for OS/390 V2R8 IP, mainly the Client Authentication support and Telnet Takeover functions.

10.1 Telnet server client authentication support

In SecureWay Communications Server for OS/390 V2R8 IP, a significant functional enhancement have been made for the Telnet server from a security point of view: client authentication support.

Starting in CS for OS/390 V2R6 IP, the TN3270 Telnet server provided the Secure Sockets Layer (SSL) function which provides secure data transmission between a secure sockets port and an SSL-enabled Telnet client.

In CS for OS/390 V2R8 IP, this function has been enhanced to support SSL client authentication, which allows additional authentication and access control checking using client certificates at the TN3270 server. The client authentication support prevents a client from seeing and getting past the USSMSG without an installation-approved certificate.

The enhancement also provides different levels of TN3270 SSL client authentication support based on an installation option defined in the TCP/IP profile. In addition, it provides the capability to exploit RACF certificate support in three security levels.

10.1.1 SSL support overview

The SSL protocol begins with a handshake. During the handshake, the client authenticates the server, the server optionally authenticates the client, and the client and server agree on how to encrypt and decrypt information.

In an SSL-encrypted session, any data on a secure port is encrypted using the SSL protocol before it is sent to the client. Data received from the client is decrypted before it is sent to other processes, such as VTAM. The flows between Telnet and VTAM are unchanged.

Secure connections can be made through a secure port. When running with base TCP/IP, Telnet connections across ports defined as secure ports are protected only by way of MD5 or SHA hashing algorithms, and they support SSL V3 clients only. Encryption support by way of RC2, RC4, DES, or triple DES requires one of the optional features shown in Table 14 on page 328.

Table 14. Optional Telnet encryption features

FMID	SSLv3 Clients	SSLv2 Clients
HTCP380	NULL SHA NULL MD5 NULL NULL	Not supported
JTCP383	RC4 MD5 Export RC2 MD5 Export NULL SHA NULL MD5 NULL NULL	RC4 Export RC2 Export
JTCP382	DES SHA RC4 MD5 Export RC2 MD5 Export NULL SHA NULL MD5 NULL NULL	RC4 Export RC2 Export
JTCP38K	Triple DES SHA US DES SHA RC4 MD5 Export RC4 SHA US RC4 MD5 US RC2 MD5 Export NULL SHA NULL MD5 NULL NULL	Triple DES US DES US RC4 Export RC4 US RC2 Export RC2 US

For more information, refer to *OS/390 SecureWay Communications Server IP Migration*, SC31-8512.

You can find additional information about the concepts of cryptography and SSL from the following Web sites:

- About SSL protocol:

  `http://home.netscape.com/eng/ssl3/ssl-toc.html`

- About the encryption methodology:

  `http://www.verisign.com/repository/crptintr.html`
  `http://www.verisign.com/client/about/introCryp.html`

A key ring file is used to obtain certificate/key information. The file may be an MVS data set or an HFS file. If the file is an MVS file, you can use RACF for protection. For HFS file formats, restrict the file access to users with superuser authority.

Starting with V2R8 the Security Server discontinued shipping the MKKF utility. Therefore, the SSL support manages the certificates within TN3270 by using a new utility, GSKKYMAN, which is shipped with System SSL.

X.509 certificates are used by both the client and server when securing communications using System SSL. The client must verify the server's certificate based on the certificate of the Certificate Authority (CA) that signed the certificate or based on a self-signed certificate from the server. The server must verify the client's certificate (if requested) using the certificate of the CA that signed the

client's certificate. The client and the server then use the session keys and begin encrypted communications.

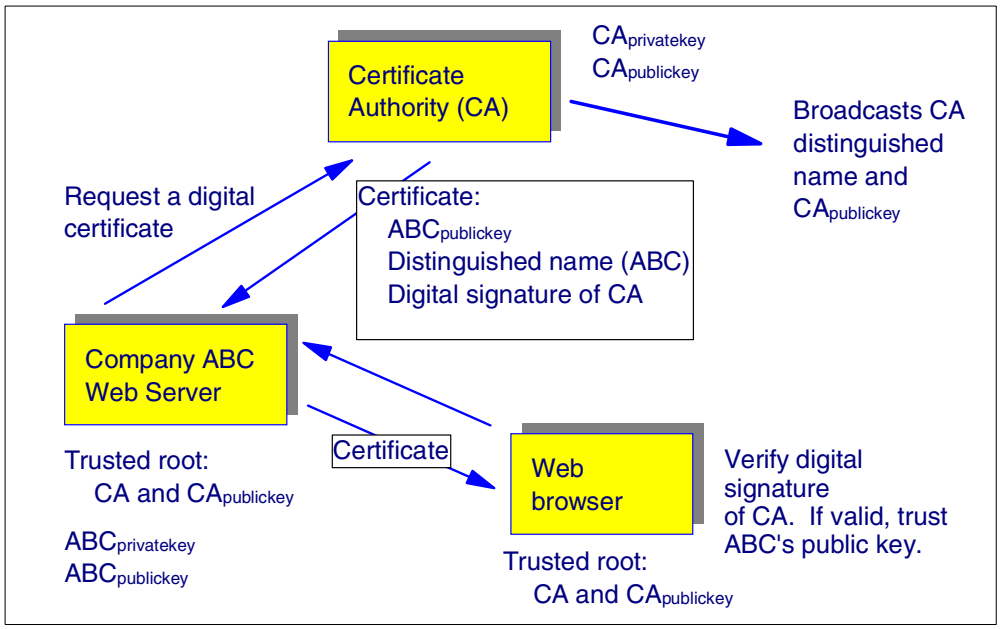


Figure 385. X.509 certificate overview

For more information on using the GSKKYMAN utility, refer to *OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference*, SC24-5877.

Multiple ports support
You can define as many as 255 listening ports for new session requests. For example, you can have 255 basic ports, 255 secure ports, or any combination of basic and secure ports, as long as the total number of ports does not exceed 255. You define ports using statements within the TELNETPARMS and BEGINVTAM information blocks. You can use the `VARY` and/or `DISPLAY` operator commands to select a particular type of port access or a specific port to display related information.

Server authentication
When using SSL to secure communications, the SSL authentication mechanism known as server authentication is used.

With server authentication, the Telnet server must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the Telnet server to the client application. The Telnet server supplies the client with the Telnet server's X.509 certificate during the initial SSL handshake. If the client validates the server's certificate, then a secure, encrypted communication channel is established between the Telnet server and the SSL-enabled client.

For server authentication to work, the Telnet server must have a private key and associated server certificate in the server's key ring file and a password stash file.

To conduct commercial business on the Internet, you might use a widely known certificate authority (CA), such as VeriSign, to get a high assurance server

certificate. For a relatively small private network within your own enterprise or group, you can issue your own server certificates, called self-signed certificates, for your own use using the GSKKYMAN utility.

Note: OS/390 Security Server will discontinue shipping the MKKF utility after OS/390 V2R7. Therefore, you have to use the GSKKYMAN utility shipped with the System Secure Sockets Layer (System SSL) element of OS/390 to implement the SSL function.

Client authentication
This solution enhances the SSL TN3270 offering (available in R6) by allowing additional authentication and access control checking using client certificates at the TN3270 server. This support prevents a client from seeing and getting past the USSMSG without an installation-approved certificate.

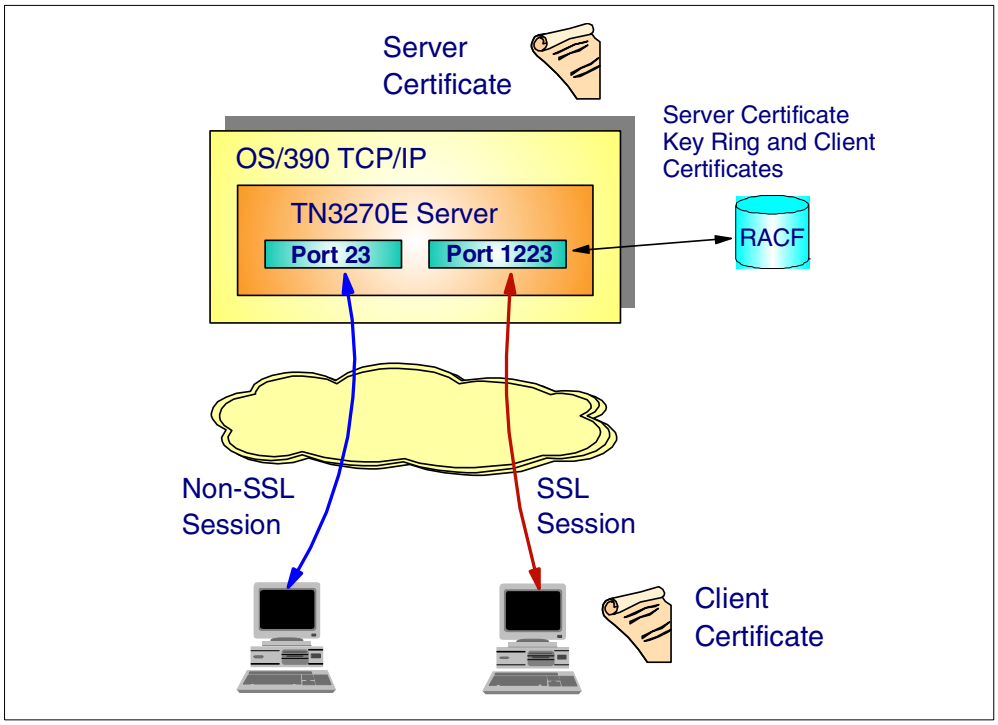


Figure 386. Telnet server client authentication support

The first level of support is to basically support SSL client authentication. In client authentication, the X.509 certificate for the client is supplied to the S/390 TN3270 server as part of the SSL handshake. To pass authentication, the Certificate Authority (CA) that signed the client certificate must be considered *trusted* by the server. That is, the certificate for the CA must be in the key ring used by the TN3270 server on the S/390. Note that the value of this option alone is determined based on which CAs are considered trusted. If the CA is a public CA and the certificate is in an easily obtained class, anyone can obtain such a certificate. Therefore, passing SSL client authentication does not provide much value-add unless coupled with the RACF support described below. If the CA is controlled by the enterprise, then the client that possesses such a certificate is at least known to the organization. Therefore, some level of access control is provided in this case.

The second level of support entails using a TN3270 client-supplied certificate as input to RACF at the TN3270 server to verify that the certificate maps to a user ID known to the system prior to issuing the USSMSG to the end user. This solution provides additional access control at the TN3270 server, which normally operates as pass-through and traditionally does no access control (unless the application is a RESTRICTAPPL or access is gained through a solicitor screen). This support ensures that the end user cannot get past the TN3270 server and attempt access to the SNA subsystem without a valid user ID on the TN3270 system.

This second level of checking returns a user ID and ensures that any user that does not have a certificate defined to the security product cannot gain access to the Telnet server. However, it also means that any user that has a certificate defined to the security product can access any Telnet port. The optional capability to restrict access to the TN3270 server on a port basis is also desirable.

To provide the capability to restrict access on a port basis, a new RACF class named SERVAUTH is provided. Under this class, the customer can specify the user IDs that are allowed to connect into a specific Telnet port by using a RACF profile name in the following format:

```
EZB.TN3270.sysname.tcpname.PORTnnnnn.
```

The user ID associated with the client certificate can then be checked against the SERVAUTH class profile entry for the Telnet port. The use of this RACF class is optional. If the SERVAUTH RACF class is active and a RACF profile for the port is defined, this level of RACF authorization will be verified prior to issuing the USSMSG to the end user. If the class is not active or there is no RACF profile protecting the port, this level of check is not required and the client is allowed to connect into Telnet as long as the client certificate was validated (as described above).

This function does not require any additional user ID and/or password entry by the client. This function does not attempt to alter how RESTRICTAPPL or the solicitor panel is processed.

RACF has supported client certificates since Version 2 Release 4. RACF Version 2 Release 8 includes the SERVAUTH class.

10.1.2 Implementing client authentication in OS/390

To implement client authentication in this release we have a new statement in TELNETPARMS BLOCK called CLIENTAUTH.

The client authentication in CS for OS/390 V2R8 IP Telnet server has three levels of security:

1. Without RACF checking, with only client certificate validation using a key database file
2. With RACF checking using only certificate validation
3. With RACF checking using certificate validation for limiting access on a port basis

In all three configurations you must have the key database containing the root certificate for the CA that issued the client certificate.

If you are using RACF you have to import the client certificate in the RACF database using the RACF `RACDCERT` command. If you want another level of security you can activate the SERVAUTH class to permit a connection of a specific user to a particular Telnet server. In this case the client certificate must be in the RACF database associated with a RACF user ID and this user ID must have permission to the SERVAUTH profile that identifies a particular Telnet server.

The implementation details are described later in this chapter.

Another requirement to implement the client authentication is the client TN3270 program emulator. It has to support the client authentication. At the ITSO we are using IBM Host On-Demand Version 4.

To implement HOD in OS/390 you have to configure a Web server, implement Java for OS/390 and install the HOD Service Manager code. In 10.1.3, “Implementation scenario” on page 332, we will show how to configure the HOD Service Manager in OS/390.

10.1.3 Implementation scenario

We have implemented all levels of client authentication in our system. We configured the five levels of authentication using HOD V.4 clients:

- Telnet session without the SSL security on TCP port 23.
- Telnet session using the SSL security only on TCP port 7723.
- Telnet connection using SSL with the client certificate verification without a security product such as RACF. TCP port 8823 is used for this connection.
- Telnet connection using SSL, the client certificate verification and RACF authorization without the SERVAUTH authorization through TCP port 9923.
- Telnet connection using SSL, the client certificate verification and RACF authorization with the SERVAUTH authorization. TCP port 6623 is configured for the highest security level.

Figure 387 on page 333 illustrates our network scenario for this particular implementation.

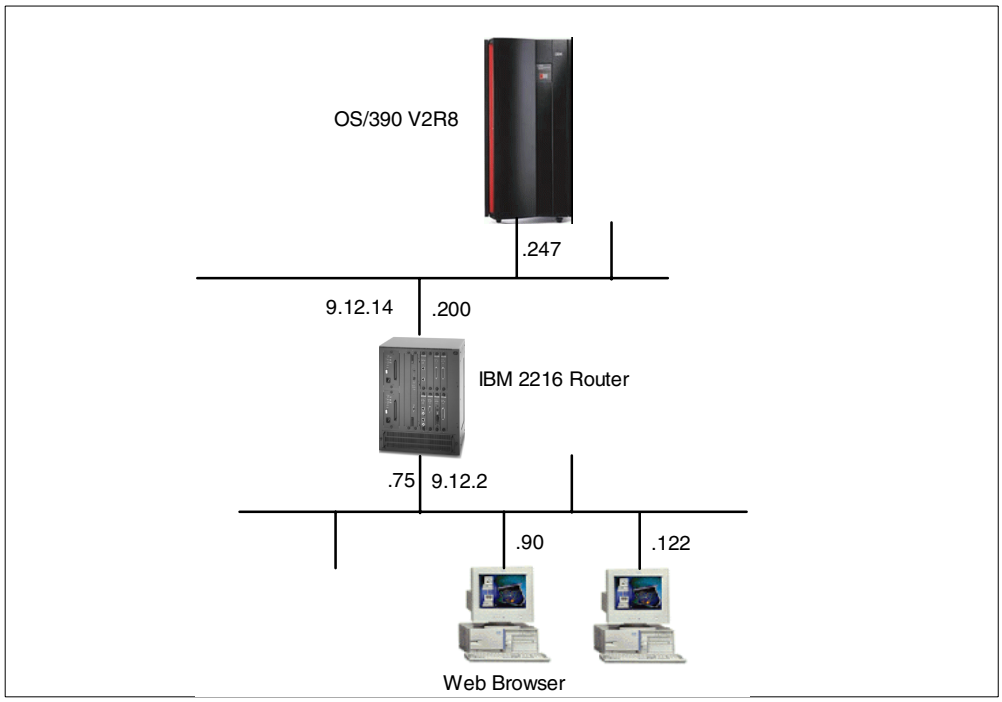


Figure 387. Client Authentication scenario network configuration

The 9672 machine is running OS/390 V2R8 with SecureWay CS for OS/390 V2R8 IP. The clients are running Windows NT Workstation and Windows 95.

10.1.3.1 Configuring the TCP/IP stack

To configure the client authentication in SecureWay IP Services you have to update the TCP/IP profile. Configure the TCP/IP profile data set using the CLIENTAUTH statement in the TELNETPARMS block, choosing the security level you want. The TELNETPARMS statements in our TCP/IP profile are shown in Figure 388 on page 334.

```
TELNETPARMS
   TKOSPECLU 3        1
   PORT 23
ENDTELNETPARMS

TELNETPARMS
   SECUREPORT 6623 KEYRING HFS /u/gianca/ssl/server/r2612.kdb
   CLIENTAUTH SAFCERT      2
   TKOSPECLU 3
ENDTELNETPARMS

TELNETPARMS
   SECUREPORT 7723 KEYRING HFS /u/gianca/ssl/server/r2612.kdb
   CLIENTAUTH NONE          3
   TKOSPECLU 3
ENDTELNETPARMS

TELNETPARMS
   SECUREPORT 8823 KEYRING HFS /u/gianca/ssl/server/r2612.kdb
   CLIENTAUTH SSLCERT        4
   TKOSPECLU 3
ENDTELNETPARMS

TELNETPARMS
   SECUREPORT 9923 KEYRING HFS /u/gianca/ssl/server/r2612.kdb 7
   CLIENTAUTH SAFCERT      5
   TKOSPECLU 3
ENDTELNETPARMS

BEGINVTAM
   PORT 23 6623 7723 8823 9923     6
   ALLOWAPPL *
   DEFAULTAPPL TSO
   IPGROUP IP1 255.0.0.0:9.0.0.0 ENDIPGROUP
   LUGROUP LU1 TCP03001..TCP03020 ENDLUGROUP
   PRTGROUP PR1 TCP03021..TCP03040 ENDPRTGROUP
   LUMAP LU1 IP1 GENERIC PR1
   PRTMAP PR1 IP1 GENERIC
   TELNETDEVICE 3287-1     ,SCS
ENDVTAM
```

Figure 388. TCPIPB stack TELNETPARMS statements

Figure 388 notes:

1 The TELNETPARMS for Telnet server without SSL.

2 This Telnet server requires client authentication using RACF and it has SERVAUTH class activated.

3 This Telnet server only requires an SSL connection.

4 This Telnet server requires client authentication without RACF security.

5 This Telnet server requires client authentication with RACF authorization but without SERVAUTH authorization.

> **Note**: If you use the SAF certificate verification, the definition in TCPIP.PROFILE is the same regardless of the usage of the SERVAUTH RACF authorization.

6 All Telnet servers are sharing the same BEGINVTAM block.

7 You have to create this key database before restarting the TCP/IP stack.

We can update the Telnet server configuration using the console command `VARY TCPIP,,OBEY` or by restarting the TCP/IP stack.

Now we will create the server key database using the GSKKYMAN utility. Go to an OMVS shell and follow the steps shown in Figure 389.

Note: Before you run the `gskkyman` command, you might need to make this file known to your OMVS environment. Issue the following command prior to issuing `gskkyman`:

```
export STEPLIB=GSK.OSV2R8.SGSKLOAD
```

```
GRAAFF @ SC57:/u/graaff>gskkyman

           IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 1
Enter key database name or press ENTER for "key.kdb": telnet.kdb    1
Enter password for the key database.......>
Enter password again for verification.....>
Should the password expire? (1 = yes, 0 = no) Ý1¨: 0

The database has been successfully created, do you want to continue to work with
 the database now? (1 = yes, 0 = no) Ý1¨: 1

           Key database menu

Current key database is /u/graaff/telnet.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 5
```

Figure 389. Creating a key database for the Telnet server

Figure 389 note:

1 This file will be used in the TCP/IP profile configuration.

```
Enter version number of the certificate to be created (1, 2, or 3) Ý3¨:  3   1
Enter a label for this key.....................> Telnet Server WTSC57
Select desired key size from the following options (512):
    1:    512
    2:    1024
Enter the number corresponding to the key size you want: 2
Enter certificate subject name fields in the following.
    Common Name (required).....................> wtsc57.itso.ibm.com
    Organization (required)....................> IBM
    Organization Unit (optional)...............> ITSO
    City/Locality (optional)...................> Poughkeepsie
    State/Province (optional)..................> New York
    Country Name (required 2 characters)..> US
Enter number of valid days for the certificate Ý365¨:  365
Do you want to set the key as the default in your key database? (1 = yes, 0 = no
) Ý1¨: 1   2
Do you want to save the certificate to a file? (1 = yes, 0 = no) Ý1¨: 1  3
Should the certificate binary data or Base64 encoded ASCII data be saved? (1 = A
SCII, 2 = binary) Ý1¨: 2   4
Enter certificate file name or press ENTER for "cert.crt": telnet.crt

Please wait while self-signed certificate is created...

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨: 1
GRAAFF @ SC57:/u/graaff>
 ===>
```

Figure 390. Creating a key pair and a self-signed certificate for the Telnet server

Figure 390 notes:

1 Choose Version 3. The version number refers to the X.509 standard version number.

2 You have to mark this key as default in the key database so that the Telnet server will select this certificate associated with this key in order to send to the client.

3 Save this certificate into a file. It will be used in an HOD configuration later.

4 Save the file in binary DER format. This is the format required in the HOD configuration.

The next step is to create a stash file where the password is stored in for the telnet server to open the key-ring. We again use the GSKKYMAN utility to do this, as shown in Figure 391 on page 337.

```
GRAAFF @ SC57:/u/graaff>gskkyman


           IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 2
Enter key database name or press ENTER for "key.kdb": telnet.kdb
Enter password for the key database.......>


           Key database menu

Current key database is /u/graaff/telnet.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 11

The encrypted password has been stored in file /u/graaff/telnet.sth

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨: 1
GRAAFF @ SC57:/u/graaff>ls -all
total 3544
drwxr-xr-x   5 LDAPPD   SYS1        8192 Apr 27 09:06 .   1
drwxr-xr-x  21 LDAPPD   SYS1        8192 Mar 22 13:17 ..
-rw-r--r--   1 LDAPPD   SYS1         611 Apr 27 09:00 telnet.crt
-rw-r--r--   1 LDAPPD   SYS1       65080 Apr 27 09:00 telnet.kdb
-rw-------   1 LDAPPD   SYS1         129 Apr 27 09:06 telnet.sth
```

Figure 391. Storing the key database password in an encrypted file

Figure 391 note:

1 After this we should have these three files in the directory.

Now we can start the TCP/IP stack.

```
S ITCPIP
IRR812I PROFILE ITCPIP.** (G) IN THE STARTED CLASS WAS USED 164
        TO START ITCPIP WITH JOBNAME ITCPIP.
$HASP100 ITCPIP   ON STCINRDR
IEF695I START ITCPIP   WITH JOBNAME ITCPIP   IS ASSIGNED TO USER
TCPIPU  , GROUP SYS1
$HASP373 ITCPIP   STARTED
IEF403I ITCPIP - STARTED - TIME=09.15.13
IEE252I MEMBER CTIEZB00 FOUND IN SYS1.PARMLIB
EZZ0300I OPENED PROFILE FILE DD:PROFILE
EZZ0309I PROFILE PROCESSING BEGINNING FOR DD:PROFILE
EZZ0316I PROFILE PROCESSING COMPLETE FOR FILE DD:PROFILE
EZZ0334I IP FORWARDING IS DISABLED
EZZ0338I TCP PORTS 1 THRU 1023 ARE RESERVED
EZZ0338I UDP PORTS 1 THRU 1023 ARE RESERVED
EZZ0336I NO LIMIT ON INCOMING UDP DATAGRAM QUEUE SET
EZZ4202I OPENEDITION-TCP/IP CONNECTION ESTABLISHED FOR ITCPIP
BPXF206I ROUTING INFORMATION FOR TRANSPORT DRIVER ITCPIP HAS BEEN
INITIALIZED OR UPDATED.
EZZ4314I INITIALIZATION COMPLETE FOR DEVICE OSA2200, LINK OSAT2200
EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE.
EZAIN11I ALL TCPIP INTERFACES FOR PROC ITCPIP ARE ACTIVE.
EZZ0400I TELNET/VTAM (SECOND PASS) BEGINNING FOR FILE: DD:PROFILE
EZZ6003I TELNET LISTENING ON PORT  6623    1
EZZ6003I TELNET LISTENING ON PORT  7723
EZZ6003I TELNET LISTENING ON PORT  8823
EZZ6003I TELNET LISTENING ON PORT  9923
EZZ6003I TELNET LISTENING ON PORT    23
EZZ0403I TELNET/VTAM (SECOND PASS) COMPLETE FOR FILE: DD:PROFILE
```

Figure 392. Starting ITCPIP stack

Figure 392 note:

1 You have to receive the EZZ6003I for each Telnet server you configured. It tells you that you have a Telnet server listening on a particular port.

10.1.3.2 Configuring HOD and JAVA

Now we will configure the HOD Service Manager on OS/390. HOD Version 4 has the following prerequisites:

- Java for OS/390 V1R1M6. If you do not have this version you can download the code from `http://www.s390.ibm.com/java`. At this site you will find the latest news, the Java code, and instructions to download and install Java on OS/390.
- Domino Go Webserver for OS/390 V5R0M0. We already have this software installed and configured in our system.
- OS/390 Version 2 Release 6 or later.
- To use the cryptographic services you must have one of the following features installed on OS/390:
  - OS/390 2.6 eNetwork Communications Server: IP Security SSL DES (56-bit export) FMID JTCP35L or later
  - OS/390 2.6 eNetwork Communications Server: IP Security Triple DES (US) FMID JTCP35K or later

Look at Table 14 on page 328 for more information about security features in OS/390.

These are the prerequisites to use the HOD Version 4 from a browser:

- Netscape Navigator 4.08 or 4.5 (Windows 95, Windows 98, Windows NT, UNIX)
- Netscape Navigator 4.04 (OS/2)
- Microsoft Internet Explorer 4.01 with SP1 or 5.0 (Windows 95, Windows 98 and Windows NT)

Use the following steps to install HOD without SMP/E:

Create a directory to mount the HFS that will hold the HOD code. If you do not have Java (JDK) code already installed you have to create a directory to hold the Java code too. Go to an OMVS shell and issue the command:

```
mkdir /usr/lpp/HOD
mkdir /usr/lpp/java
```

Create two HFS data sets, one for HOD and the other for Java, if necessary, using TSO or a JCL. You can use the JCL in Figure 393 to create the HFS data sets.

```
//HFS@CRE1  JOB (ACCTNUM,EXP),'PGMRNAME',
//     TIME=1440,NOTIFY=&SYSUID,REGION=6500K,
//     CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//ALLOCHFS  EXEC PGM=IEFBR14
//DD1       DD DSN=OMVS.RA03.HOD390,
//             DISP=(NEW,KEEP,DELETE),
//             STORCLAS=STANDARD,
//             DSNTYPE=HFS,
//             SPACE=(CYL,(600,250,20))
//DD1       DD DSN=OMVS.RA03.JDK116,
//             DISP=(NEW,KEEP,DELETE),
//             STORCLAS=STANDARD,
//             DSNTYPE=HFS,
//             SPACE=(CYL,(250,150,20))
//*
```

Figure 393. Sample HOD JCL to create an HFS file

In our environment the HOD V4 code has been restored from a CD-ROM. We uploaded the code in the /usr/lpp/HOD directory using FTP. If you use an SMP version of HOD follow the instructions in the program directory.

Then download the Java code to the /usr/lpp/java directory.

All the TAR files must be uploaded in binary format into an OS/390 system. The shell script to install the HOD code must be uploaded in ASCII format (converting to EBCDIC).

Check the HOD and Java directory. At ITSO Raleigh, the environment looks similar to Figure 394 on page 340.

```
GIANCA @ RA03:/tmp>cd /usr/lpp/HOD
GIANCA @ RA03:/usr/lpp/HOD>ls -al
drwx---rwx   3 KAKKY    DCEGRP      8192 Jul  8 14:36 .
drwxr-xr-x  27 2134     SYS1        8192 Jun 30 11:34 ..
-rw-r-----   1 GIANCA   DCEGRP   161034240 Jul  1 19:09 HOD40MVSCD.TAR
-rw-r-----   1 GIANCA   DCEGRP   12257280 Jul  1 19:00 HOD40SRV.TAR
-rw-r-----   1 GIANCA   DCEGRP   270233600 Jul  1 18:59 HOD40WWW.TAR
-rw-r-----   1 GIANCA   DCEGRP      2594 Jul  8 14:36 hod40mvs.sh
GIANCA @ RA03:/usr/lpp/HOD>cd /usr/lpp/java
GIANCA @ RA03:/usr/lpp/java>ls -al
drwxr-xr-x   3 KAKKY    DCEGRP      8192 Jun 17 09:54 .
drwxr-xr-x  27 2134     SYS1        8192 Jun 30 11:34 ..
-rw-r-----   1 GIANCA   DCEGRP   50669271 Jun 17 09:52 HJVA11D_TAR.Z
```

Figure 394. HOD and Java directories after code is downloaded

To install the HOD code run the `hod40mvs.sh`. It will create a directory named /usr/lpp/HOD/hostondemand. This directory contains all the HOD code: Java files, HTML files, help files, and user configuration files.

```
GIANCA @ RA03:/usr/lpp/HOD>ls -al
drwx---rwx   3 KAKKY    DCEGRP      8192 Jul  8 14:36 .
drwxr-xr-x  27 2134     SYS1        8192 Jun 30 11:34 ..
-rw-r-----   1 GIANCA   DCEGRP   161034240 Jul  1 19:09 HOD40MVSCD.TAR
-rw-r-----   1 GIANCA   DCEGRP   12257280 Jul  1 19:00 HOD40SRV.TAR
-rw-r-----   1 GIANCA   DCEGRP   270233600 Jul  1 18:59 HOD40WWW.TAR
-rw-r-----   1 GIANCA   DCEGRP      2594 Jul  8 14:36 hod40mvs.sh
GIANCA @ RA03:/usr/lpp/HOD>hod40mvs.sh
GIANCA @ RA03:/usr/lpp/HOD>ls -al
drwx---rwx   3 KAKKY    DCEGRP      8192 Jul  8 14:36 .
drwxr-xr-x  27 2134     SYS1        8192 Jun 30 11:34 ..
-rw-r-----   1 GIANCA   DCEGRP   161034240 Jul  1 19:09 HOD40MVSCD.TAR
-rw-r-----   1 GIANCA   DCEGRP   12257280 Jul  1 19:00 HOD40SRV.TAR
-rw-r-----   1 GIANCA   DCEGRP   270233600 Jul  1 18:59 HOD40WWW.TAR
-rw-r-----   1 GIANCA   DCEGRP      2594 Jul  8 14:36 hod40mvs.sh
drwxr-xr-x   5 OMVSKERN DCEGRP      8192 Jun 29 18:16 hostondemand
GIANCA @ RA03:/usr/lpp/HOD>cd hostondemand
GIANCA @ RA03:/usr/lpp/HOD/hostondemand>ls -al
drwxr-xr-x   5 OMVSKERN DCEGRP      8192 Jun 29 18:16 .
drwx---rwx   3 KAKKY    DCEGRP      8192 Jul  8 14:36 ..
drwxr-xr-x  37 OMVSKERN DCEGRP    114688 Jun 29 18:51 HOD
drwxr-xr-x   5 OMVSKERN DCEGRP      8192 Jun 29 18:43 lib
drwxr-xr-x   2 OMVSKERN DCEGRP      8192 Jul  6 10:00 private
```

Figure 395. Installing HOD Version 4 using hod40mvs.sh

To install the Java code go to the Java directory and complete the following steps:

```
GIANCA @ RA03:/usr/lpp/HOD>cd /usr/lpp/java
GIANCA @ RA03:/usr/lpp/java>tar -xpozf HJVA11D.TAR.Z
GIANCA @ RA03:/usr/lpp/java>ls -al
total 99016
drwxr-xr-x   3 KAKKY    DCEGRP      8192 Jun 17 09:54 .
drwxr-xr-x  27 2134     SYS1        8192 Jun 30 11:34 ..
-rw-r-----   1 GIANCA   DCEGRP   50669271 Jun 17 09:52 HJVA11D_TAR.Z
drwxr-xr-x   7 GIANCA   DCEGRP      8192 May 27 14:55 J1.1
GIANCA @ RA03:/usr/lpp/java>cd J1.1
GIANCA @ RA03:/usr/lpp/java/J1.1>ls -al
total 3968
drwxr-xr-x   7 GIANCA   DCEGRP      8192 May 27 14:55 .
drwxr-xr-x   3 KAKKY    DCEGRP      8192 Jun 17 09:54 ..
-rw-r--r--   1 GIANCA   DCEGRP     96425 May 27 14:53 CHANGES.ADK
-rw-r--r--   1 GIANCA   DCEGRP       490 May 27 14:53 COPYRIGHT
-rw-r--r--   1 GIANCA   DCEGRP     28294 May 27 14:53 RAWT-readme.html
drwxr-xr-x   3 GIANCA   DCEGRP      8192 May 27 14:49 bin
drwxr-xr-x  25 GIANCA   DCEGRP      8192 May 27 14:54 demo
-rw-r--r--   1 GIANCA   DCEGRP      6714 May 27 14:53 fixes.lst
-rw-r--r--   1 GIANCA   DCEGRP       456 May 27 14:53 hw.class
drwxr-xr-x   3 GIANCA   DCEGRP      8192 May 27 14:53 include
drwxr-xr-x   4 GIANCA   DCEGRP      8192 May 27 14:55 jni_example
drwxr-xr-x   4 GIANCA   DCEGRP      8192 May 27 14:53 lib
-rw-r--r--   1 GIANCA   DCEGRP   1828051 May 27 14:52 src.tar.Z
```

Figure 396. Installing Java on OS/390

Next you have to create three environment variables to map the Java code to the applications:

```
$ export JAVA_HOME=/usr/lpp/java/J1.1
$ export PATH=/usr/lpp/java/J1.1:$PATH
$ export CLASSPATH=/usr/lpp/java/J1.1/lib/classes.zip:$CLASSPATH
```

Note: Put these three export commands in the /etc/profile file. They will be executed for all subsequent processes using a shell environment.

To check if Java is configured correctly and ready to be used, issue the following command:

```
GIANCA @ RA03:/u/gianca/>java -version
java version "1.1.6"
```

Figure 397. Checking Java installation and configuration

The attributes HFS data sets after Java and the HOD installation at ITSO Raleigh are shown in Figure 398 and Figure 399 on page 342.

```
Data Set Name . . . . : OMVS.RA03.HOD390

General Data                          Current Allocation
 Management class . . : STANDARD       Allocated cylinders : 2,100
 Storage class  . . . : STANDARD       Allocated extents . : 9
  Volume serial . . . : SPLEX7
  Device type . . . . : 3390
 Data class . . . . . : **None**      Current Utilization
  Organization  . . . : PO             Used pages  . . . . : 267,047
  Record format . . . : U              % Utilized  . . . . : 70
  Record length . . . : 0
  Block size  . . . . : 0
  1st extent cylinders: 600
  Secondary cylinders : 250
  Data set name type  : HFS

  Creation date . . . : 1999/06/16     Referenced date . . : 1999/06/30
  Expiration date . . : ***None***
```

Figure 398. HOD HFS data set attributes

```
Data Set Name . . . . : OMVS.RA03.JDK116

General Data                          Current Allocation
 Management class . . : STANDARD       Allocated cylinders : 250
 Storage class  . . . : STANDARD       Allocated extents . : 1
  Volume serial . . . : SPLEX7
  Device type . . . . : 3390
 Data class . . . . . : **None**      Current Utilization
  Organization  . . . : PO             Used pages  . . . . : 40,275
  Record format . . . : U              % Utilized  . . . . : 89
  Record length . . . : 0
  Block size  . . . . : 0
  1st extent cylinders: 250
  Secondary cylinders : 150
  Data set name type  : HFS

  Creation date . . . : 1999/06/16     Referenced date . . : 1999/06/30
  Expiration date . . : ***None***
```

Figure 399. Java HFS data set attributes

10.1.3.3 Working with the certificate for the Telnet server

CS for OS/390 V2R8 IP supports SSL for Telnet connections, and HOD can make use of this to provide secure sessions directory between a client and the OS/390 system. In order to implement this scenario, you must extract the server's certificate from its key database and then make it available to HOD clients. In other words, the HOD clients must have access to either the site certificate of CS for OS/390 or the Certificate Authority (CA) root that signed the site certificate of CS for OS/390.

If your Telnet server is using a site certificate from VeriSign or Thawte, no further action is needed, but if it is using a self-signed certificate, or a site certificate that was not signed by VeriSign or Thawte, you must follow the procedures described in this section.

Important

To make an SSL connection the HOD client must trust the server. When an SSL connection is started the server sends its certificate to the client. Then, HOD checks if this server can be trusted. Import the server certificate created in Figure 390 on page 336 into a file named CustomizedCAs.class using the `keyrng` Java utility provided by HOD.

Next add the /usr/lpp/HOD/hostondemand/lib/sm.zip file to your CLASSPATH in your .profile. Then complete the following steps.

Go to the directory where you created the certificate file in Figure 390 on page 336.

```
GRAAFF @ SC57:/u/graaff>java com.ibm.hodsslight.tools.keyrng CustomizedCAs add
--site telnet.crt    1
Password for CustomizedCAs.class:   2
Done.   3
GRAAFF @ SC57:/u/graaff>ls -all
drwxr-xr-x   5 LDAPPD   SYS1        8192 Apr 27 09:38 .
drwxr-xr-x  21 LDAPPD   SYS1        8192 Mar 22 13:17 ..
-rw-r--r--   1 LDAPPD   SYS1        1217 Apr 27 09:38 CustomizedCAs.class
-rw-r--r--   1 LDAPPD   SYS1         611 Apr 27 09:00 telnet.crt
-rw-r--r--   1 LDAPPD   SYS1       65080 Apr 27 09:00 telnet.kdb
-rw-------   1 LDAPPD   SYS1         129 Apr 27 09:06 telnet.sth
GRAAFF @ SC57:/u/graaff>cp CustomizedCAs.class /usr/lpp/HOD/hostondemand/HOD 4
```

Figure 400. HOD creating CustomizedCAs.class file

Figure 400 notes:

1 Add the certificate to the CustomizedCAs.class file, using the keyrng Java utility. Issue the `java` command as shown above. To facilitate you can create a script file. If you are adding a CA root certificate, rather than a site or self-signed certificate, you have to specify `ca` instead of `site` specified above. The name of the file that contains the certificate in binary DER format has to be specified.

Note: CustomizedCAs must be capitalized exactly as shown.

2 If no CustomizedCAs.class file exists, keyrng prompts you for a password with which to encrypt the new class file. However, CustomizedCAs.class files must *not* be encrypted, so just press Enter at the password prompt.

3This message indicates the file has been created.

4Copy this file to the HOD Web published directory.

10.1.3.4 Starting HTTP server and HOD Service Manager

First create the started procedure for the HOD Service Manager. An example of the JCL is located in directory /usr/lpp/HOD/hostondemand/lib member HOMSERVR:

```
//HODSRV PROC
//HODSRVG EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT,
//      PARM='sh /usr/lpp/HOD/hostondemand/lib/ServiceManager.sh'
//SYSPRINT DD SYSOUT=A
```

```
//SYSERR   DD SYSOUT=A
//STDOUT   DD SYSOUT=A
//STDERR   DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSIN    DD DUMMY
```

Associate the HOD started task with a RACF user ID that has an OMVS segment defined:

```
RDEFINE STARTED HODSRV STDATA(USER(HODSRV))
SETROPTS RACLIST(STARTED) REFRESH
```

To create a user ID to be used with HOD issue the following commands:

```
ADDUSER HODSRV OMVS(HOME('/') UID(777))
           DFLTGRP(OMVSGRP) NAME('HOD Server')
```

Note: The GROUP must have an OMVS segment defined, too.

If you do not have HTTP server running, see *WebSphere Application Server for OS/390 HTTP Server Planning, Installing, and Using*, SC31-8690.

The following is our started procedure for the HTTP server:

```
//IMWEBSRV  PROC  LEPARM='ENVAR("_BPXK_SETIBMOPT_TRANSPORT=ITCPIP")',
//    ICSPARM='-p 80 -vv -r /etc/httpd.conf'
//WEBSRV   EXEC PGM=IMWHTTPD,REGION=0K,TIME=NOLIMIT,
//         PARM=('&LEPARM/&ICSPARM')
//SYSIN    DD DUMMY
//OUTDSC   OUTPUT DEST=HOLD
//SYSPRINT DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSERR   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//STDOUT   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//STDERR   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSOUT   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//CEEDUMP  DD SYSOUT=*,OUTPUT=(*.OUTDSC)
```

Beginning with DGW 5.0, several modules in HFS directories must be defined as *program controlled*. Ensure that all dynamic libraries for the HTTP server have the program-controlled extended attribute set. The following screen shows you the file attributes in our system:

```
KAKKY @ RA03:/usr/lpp/internet/bin>ls -E *.dll
-rwxr-x--x  -ps  2 OMVSKERN 205      6909952 May 11 21:20 cms.dll
-rwxr-x--x  -ps  2 OMVSKERN 205        40960 May 11 21:22 cmskus.dll
-rwxr-x--x  -ps  2 OMVSKERN 205        53248 May 11 21:22 gsksys.dll
-rwxr-x--x  -ps  2 OMVSKERN 205      1306624 May 11 21:22 keyman.dll
-rwxr-x--x  -ps  2 OMVSKERN 205       204800 May 11 21:22 nspcommon.dll
-rwxr-x--x  -ps  2 OMVSKERN 205       765952 May 11 21:22 pfx.dll
-rwxr-x--x  -ps  2 OMVSKERN 205      1667072 May 11 21:22 skit.dll
-rwxr-x--x  -ps  2 OMVSKERN 205      2260992 May 11 21:22 x509cms.dll
KAKKY @ RA03:/usr/lpp/internet/bin>ls -E *.so
erwxrwxrwx       1 OMVSKERN OMVSGRP        6 Apr 20 17:20 IMWX00.so -> IMWX00
-rwxr-x--x  -ps  2 OMVSKERN 205       385024 May 11 21:21 Jav_dll.so
-rwxr-x--x  -ps  2 OMVSKERN 205        53248 May 11 21:21 gskipc.so
-rwxr-x--x  -ps  2 OMVSKERN 205       225280 May 11 21:21 htcounter.so
-rwxr-x--x  -ps  2 OMVSKERN 205       139264 May 11 21:21 libfcgi.so
-rwxr-x--x  -ps  2 OMVSKERN 205       221184 May 11 21:21 libhttpdapi.so
-rwxr-x--x  -ps  2 OMVSKERN 205        90112 May 11 21:21 mvsds.so
-rwxr-x--x  -ps  2 OMVSKERN 205        36864 May 11 21:22 wwwus.so
```

Figure 401. The file attributes in the /usr/lpp/internet/bin directory

If they do not have the program-controlled attribute, issue the `extattr` command to set the `p` attributes on files in the HFS.

For HOD and Java you have to include the following configuration in the /etc/httpd.conf file:

```
Pass /HOD/*.html   /usr/lpp/HOD/hostondemand/HOD/*.html.ascii
Pass /HOD/*.HTML   /usr/lpp/HOD/hostondemand/HOD/*.HTML.ascii
Pass /HOD/*        /usr/lpp/HOD/hostondemand/HOD/*
Pass /java/* /usr/lpp/java/J1.1/*
```

Now you can start the HOD server:

```
S HODSRV
$HASP100 HODSRV   ON STCINRDR
IEF695I START HODSRV   WITH JOBNAME HODSRV   IS ASSIGNED TO USER HODSRV
 , GROUP OMVSGRP
$HASP373 HODSRV   STARTED
IEF403I HODSRV - STARTED - TIME=10.00.44
```

Figure 402. Starting the HOD server

After starting the HOD started task, check the HOD server status using the Display MVS commands:

```
D A,L
IEE114I 14.03.43 2000.125 ACTIVITY 871
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
00005    00028    00001      00026    00013    00001/00025       00018
 LLA      LLA      LLA      NSW  S  JES2     JES2     IEFPROC  NSW  S
 VLF      VLF      VLF      NSW  S  NET      NET      NET      NSW  S
 RMF      RMF      IEFPROC  NSW  S  APPC     APPC     APPC     NSW  S
 RRS      RRS      RRS      NSW  S  OPTSO    OPTSO    OPTSO    OWT  S
 CSF      CSF               NSW  S  BWK      BWK      BWKSTEP  NSW  S
 RACF     RACF     RACF     NSW  S  ITCPIP   ITCPIP   TCPIP    NSW  SO
 TSO      TSO      STEP1    OWT  S  INETD1   STEP1    OMVSKERN OWT  AO
 DB2OMSTR DB2OMSTR IEFPROC  NSW  S  PORTMAP5 STEP1    STC      OWT  AO
 FTPD1    STEP1    OMVSKERN OWT  AO IRLOPROC IRLOPROC          NSW  S
 DB2UMSTR DB2UMSTR IEFPROC  NSW  S  IRLUPROC IRLUPROC          NSW  S
 DB2ODBM1 DB2ODBM1 IEFPROC  NSW  S  DB2UDBM1 DB2UDBM1 IEFPROC  NSW  S
 CICS57   CICS57   CICS530  NSW  S  ADSMCLI  ADSMCLI  *OMVSEX  IN   SO
 DB2UDIST DB2UDIST IEFPROC  NSW  SO DB2USPAS DB2USPAS IEFPROC  NSW  S
 DB2ODIST DB2ODIST IEFPROC  NSW  SO DB2OSPAS DB2OSPAS IEFPROC  NSW  S
 LDAPSRV  LDAPSRV  GO       OWT  SO HODSRV   HODSRV   *OMVSEX  OWT  SO
 HODSRV4  STEP1    HODSRV   OWT  AO HODSRV6  *OMVSEX  HODSRV   IN   AO
 IMWEBSRV IMWEBSRV WEBSRV   IN   SO
GRAAFF  IN   O
```

Figure 403. Checking HOD server: MVS console command

You must have three HOD processes running. If the HOD started task ends after it was started, you probably have some configuration problems. The best way to see what is happening is to execute the ServiceManager.sh script directly from a shell command. You will see all error messages in the shell itself.

The `Display TCPIP` console command is very useful to see if HOD is running properly by checking the TCP/IP sockets that are opened:

```
D TCPIP,ITCPIP,N,SOCKETS
EZZ2500I NETSTAT CS V2R8 ITCPIP 404
SOCKETS INTERFACE STATUS:
TYPE   BOUND TO              CONNECTED TO          STATE    CONN
NAME: BPXOINIT  SUBTASK: 007EFBF8
STREAM 0.0.0.0..10007        0.0.0.0..0            LISTEN   00000018
NAME: DB2ODIST  SUBTASK: 007D3520
STREAM 9.12.14.247..33325    0.0.0.0..0            LISTEN   0000003C
NAME: DB2ODIST  SUBTASK: 007D3B70
STREAM 0.0.0.0..33324        0.0.0.0..0            LISTEN   0000003B
NAME: DB2UDIST  SUBTASK: 007D3498
STREAM 9.12.14.247..33327    0.0.0.0..0            LISTEN   00000039
NAME: DB2UDIST  SUBTASK: 007D3B70
STREAM 0.0.0.0..33326        0.0.0.0..0            LISTEN   00000038
NAME: FTPD1     SUBTASK: 007EC2D0
STREAM 0.0.0.0..21           0.0.0.0..0            LISTEN   00000031
NAME: HODSRV6   SUBTASK: 007E3100
STREAM 9.12.14.247..1031     9.12.14.247..389      ESTABLSH 00000160
STREAM 9.12.14.247..1030     9.12.14.247..389      ESTABLSH 0000015E
STREAM 0.0.0.0..6612         0.0.0.0..0            LISTEN   00000162
NAME: HODSRV6   SUBTASK: 007E3E88
STREAM 0.0.0.0..8999         0.0.0.0..0            LISTEN   00000154
NAME: HODSRV6   SUBTASK: 007EC2D0
STREAM 0.0.0.0..8989         0.0.0.0..0            LISTEN   0000015D
NAME: INETD1    SUBTASK: 007EC400
STREAM 0.0.0.0..7            0.0.0.0..0            LISTEN   00000025
STREAM 0.0.0.0..19           0.0.0.0..0            LISTEN   00000027
STREAM 0.0.0.0..9            0.0.0.0..0            LISTEN   00000026
STREAM 0.0.0.0..37           0.0.0.0..0            LISTEN   00000029
STREAM 0.0.0.0..514          0.0.0.0..0            LISTEN   00000022
STREAM 0.0.0.0..513          0.0.0.0..0            LISTEN   00000023
STREAM 0.0.0.0..512          0.0.0.0..0            LISTEN   00000024
STREAM 0.0.0.0..13           0.0.0.0..0            LISTEN   00000028
NAME: ITCPIP    SUBTASK: 007E17A0
STREAM 0.0.0.0..23           0.0.0.0..0            LISTEN   0000001D
NAME: ITCPIP    SUBTASK: 007E19C0
STREAM 0.0.0.0..9923         0.0.0.0..0            LISTEN   0000001C
NAME: ITCPIP    SUBTASK: 007E1B58
STREAM 0.0.0.0..8823         0.0.0.0..0            LISTEN   0000001B
NAME: ITCPIP    SUBTASK: 007E1CF0
STREAM 0.0.0.0..7723         0.0.0.0..0            LISTEN   0000001A
NAME: ITCPIP    SUBTASK: 007E1E88
STREAM 0.0.0.0..6623         0.0.0.0..0            LISTEN   00000019
NAME: ITCPIP    SUBTASK: 007E6610
STREAM 127.0.0.1..1026       127.0.0.1..1025       ESTABLSH 00000016
NAME: ITCPIP    SUBTASK: 007F6088
STREAM 0.0.0.0..1025         0.0.0.0..0            LISTEN   0000000B
STREAM 127.0.0.1..1025       127.0.0.1..1026       ESTABLSH 00000017
NAME: LDAPSRV   SUBTASK: 007F69B0
STREAM 9.12.14.247..389      9.12.14.247..1031     ESTABLSH 00000161
STREAM 9.12.14.247..389      9.12.14.247..1030     ESTABLSH 0000015F
STREAM 0.0.0.0..389          0.0.0.0..0            LISTEN   00000152
STREAM 0.0.0.0..636          0.0.0.0..0            LISTEN   00000153
NAME: MVSNFSC   SUBTASK: 007E38A8
DGRAM  9.12.14.247..1023     *..*                  UDP      0000000C
NAME: PORTMAP5  SUBTASK: 007EC2A0
DGRAM  0.0.0.0..111          *..*                  UDP      0000002F
STREAM 0.0.0.0..111          0.0.0.0..0            LISTEN   00000030
39 OF 39 RECORDS DISPLAYED
```

Figure 404. HOD server checking: D TCPIP,TCPIPB,N,SOCKETS console command

10.1.3.5 Customizing HOD to use SSL with client authentication

You have to use a browser to customize HOD. In our environment we tested all functions using Internet Explorer and Netscape Navigator.

Go to a browser and start an HTTP session with the host that is running the HOD server, HTTP server and Telnet server:

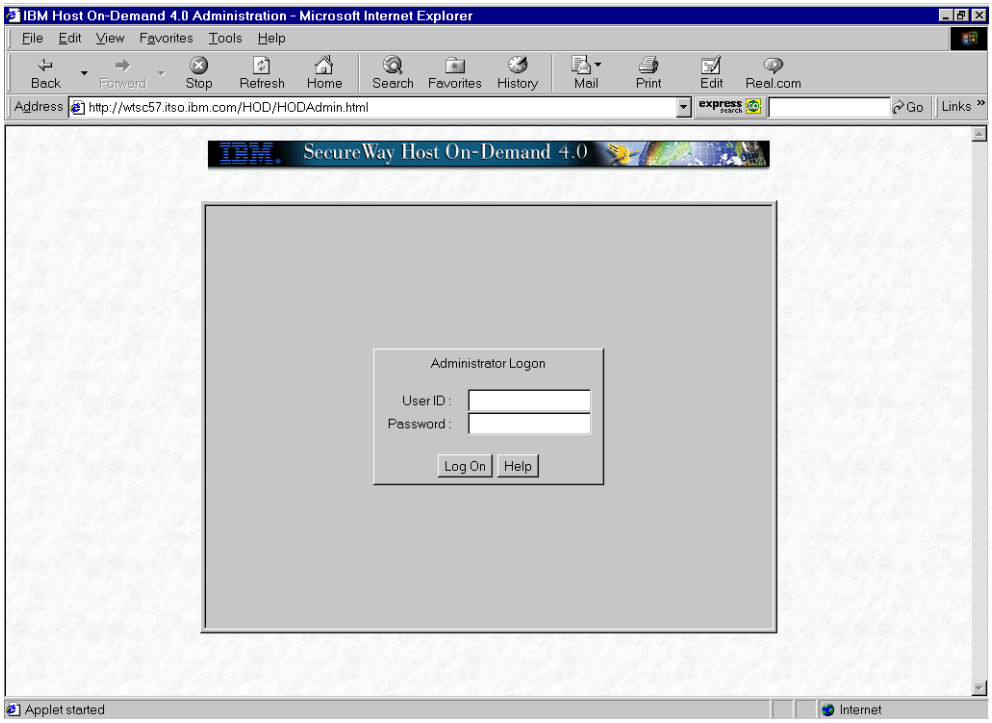


Figure 405. HOD Administrator logon screen

Fill in the fields by typing `admin` in the User ID field and `password` in the Password field, then click **Log On**.

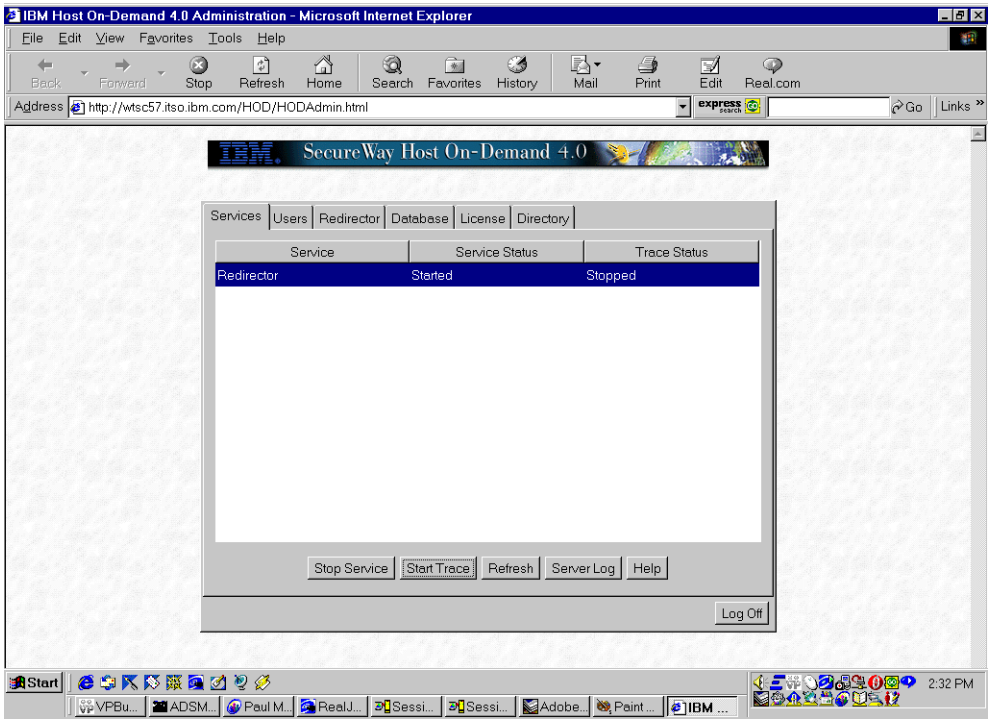


Figure 406. HOD Administrator panel

Now we will define a user to configure the sessions. Select the **Users** tab.

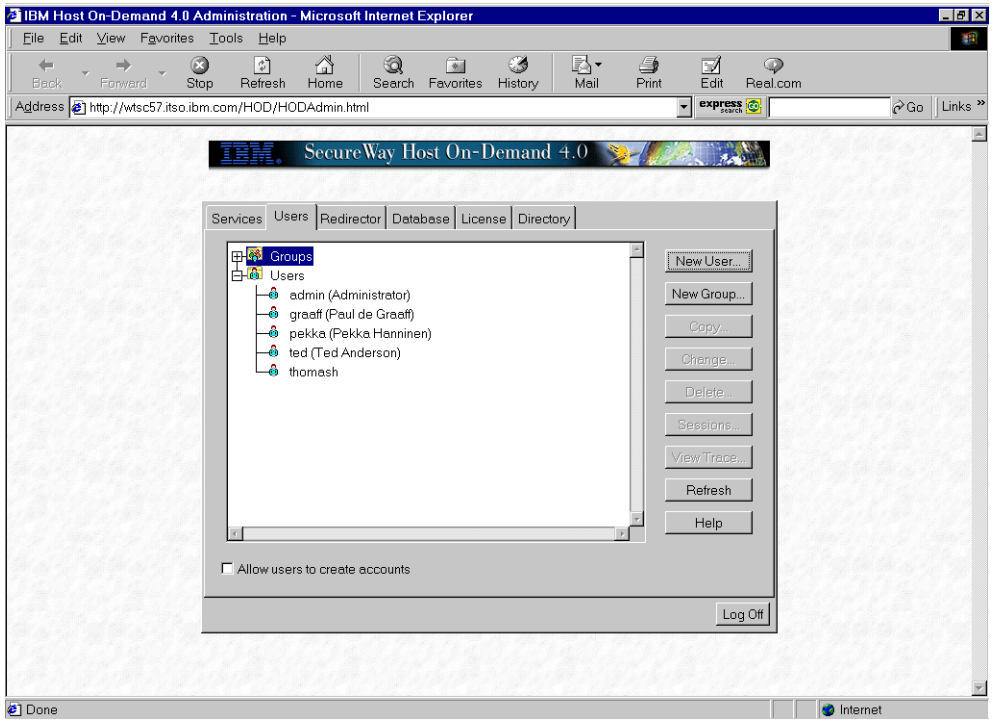


Figure 407. HOD users administration - defining users

Note: In the example shown we had already users defined.

In this screen, expand the **Users** tree and click **New User**.

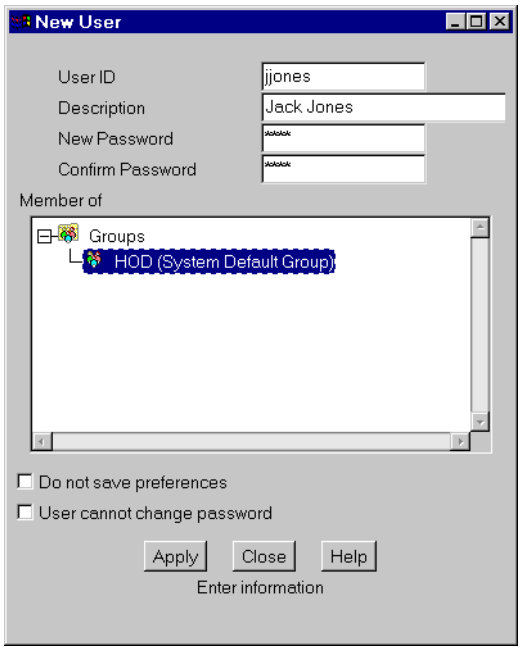


Figure 408. HOD Administration - user definition

Fill in the fields as shown in Figure 408. Select **HOD** (the default group), click **Apply**. A message will appear to indicate the account has been created, as shown in Figure 409.

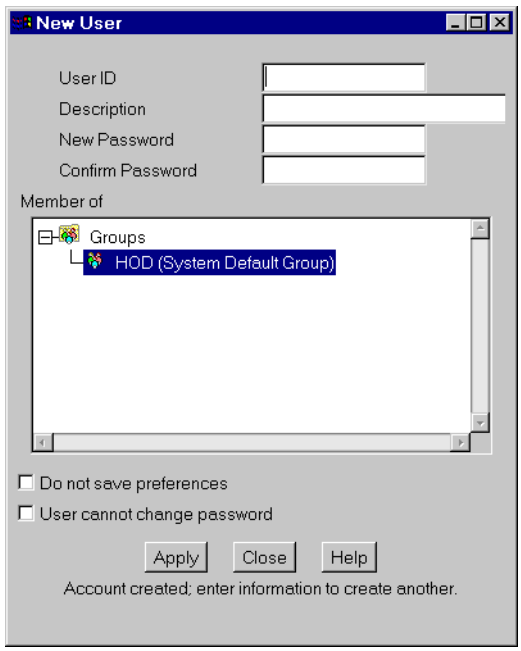


Figure 409. HOD Administration - account created message

Note: You can create more than one user without leaving this screen.

To finish click **Close**. You now return to the User window again, as shown in Figure 410 on page 351.

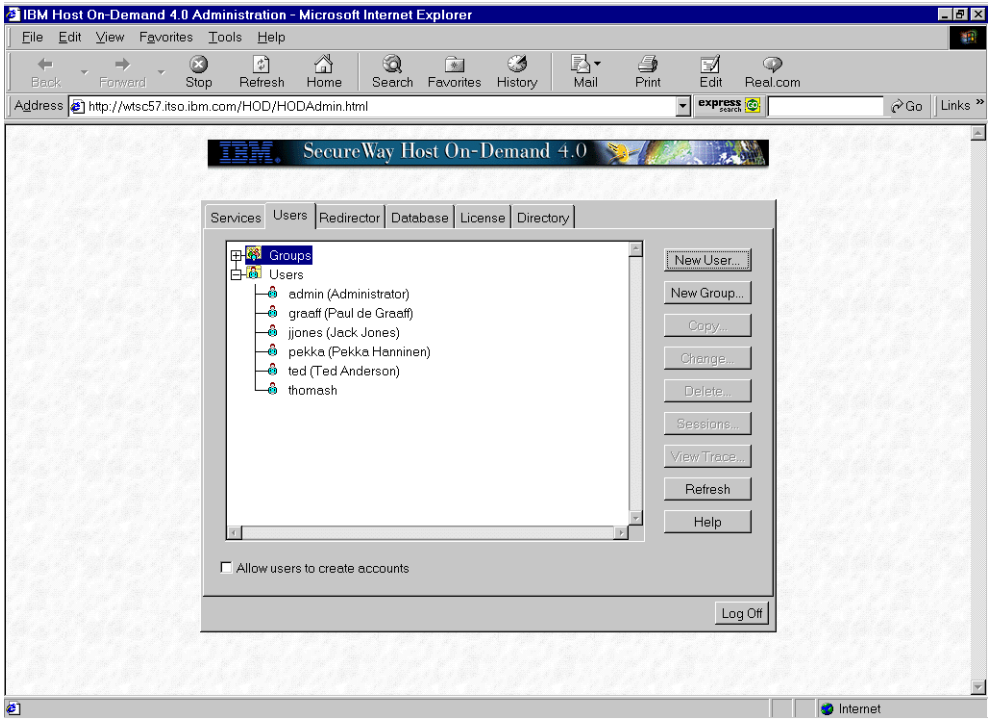


Figure 410. HOD administration - defining user sessions

Select the user you created and click **Sessions**.

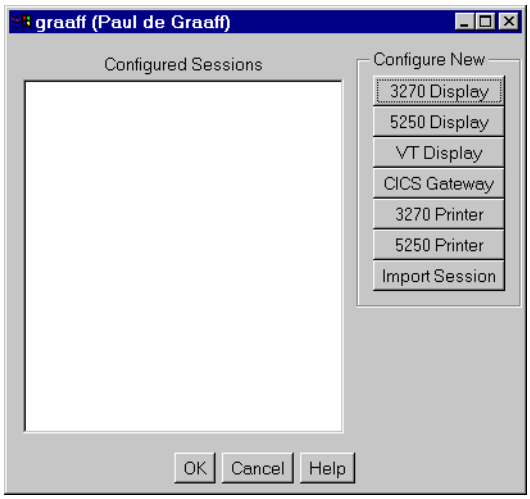


Figure 411. HOD administration - defining the first session

Now we need to define the 3270 sessions for the Telnet servers. The first session will be a Telnet 3270 display session. Click **3270 Display**.

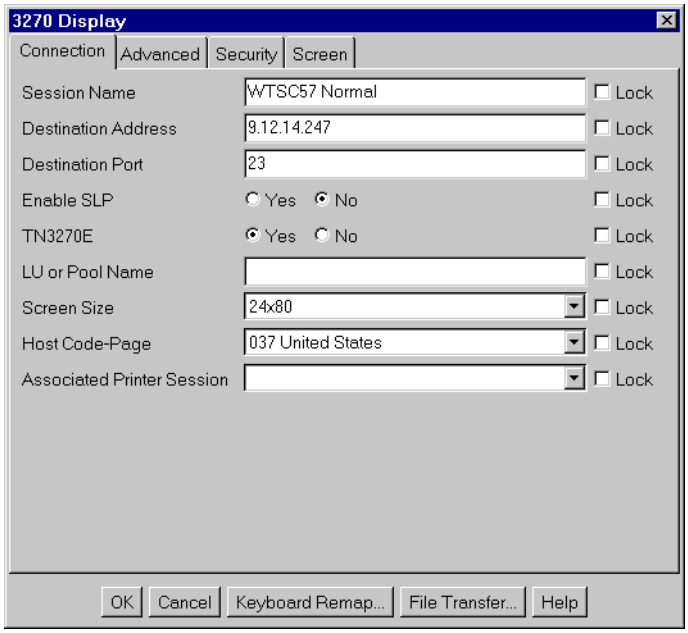


Figure 412. HOD administration - defining a Telnet session without SSL

Fill in the fields as shown in Figure 412 and click **OK**.

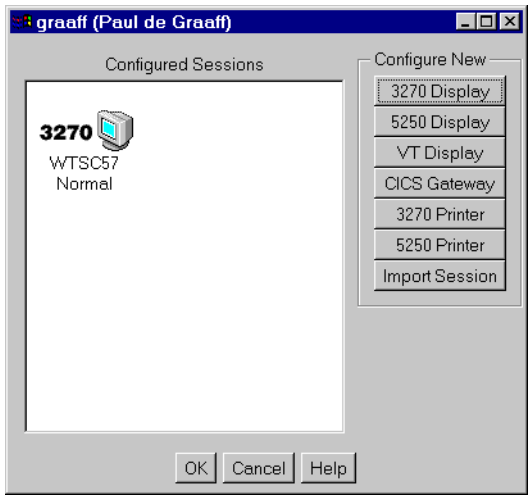


Figure 413. HOD administration - the first session defined

Now, instead of using the same process to create the other sessions, you can right-click the session icon and select **Copy**. It will duplicate the session. Select the new session you just created using the right button again and select **Properties**.

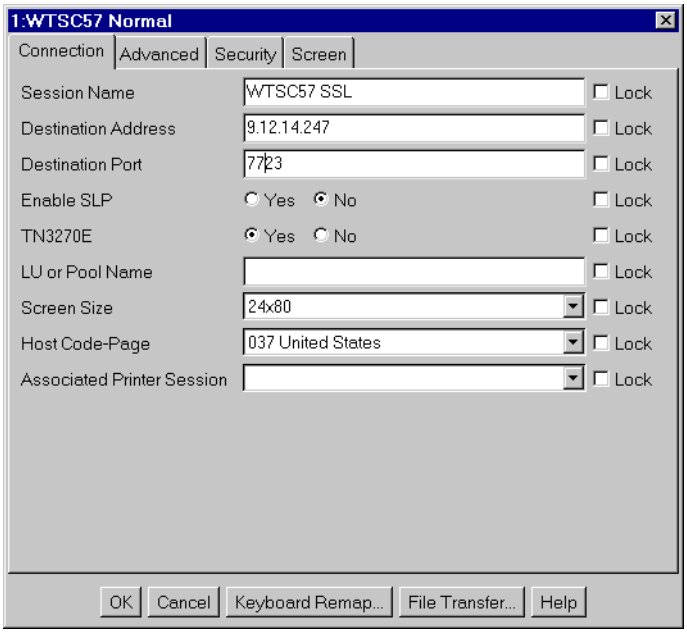


Figure 414. HOD administration - defining SSL session

Fill in the fields as shown in Figure 414 and click the **Security** tab.

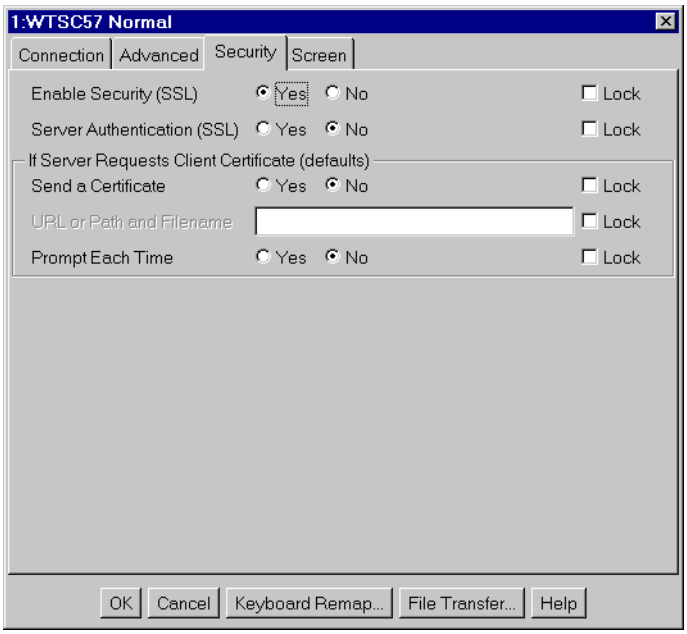


Figure 415. HOD administration - defining SSL session - security parameters

Check the boxes as shown in Figure 415. For this session we chose only **Enable Security (SSL)**. Click **OK** and copy this second session, too.

To create another secure session, select the new session using the right button and select **Properties**.

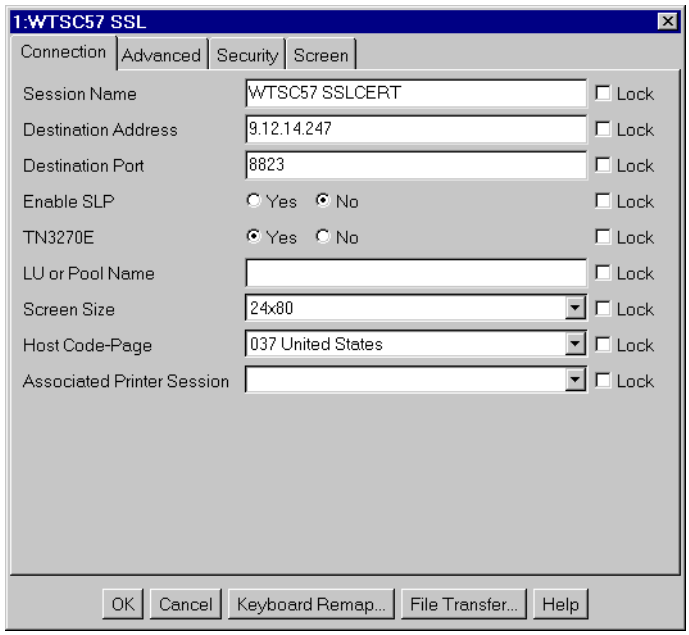


Figure 416. HOD administration - SSL client authentication

Fill in the fields as shown in Figure 416 and click **Security**.

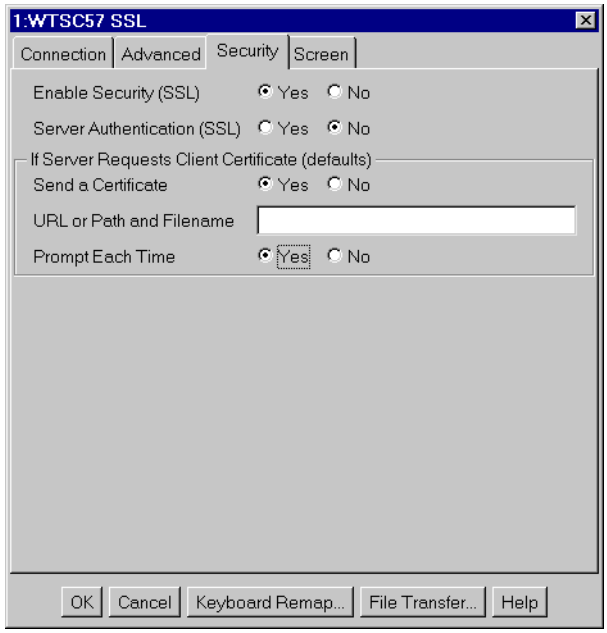


Figure 417. HOD administration - SSL client authentication - security parameters

Check the boxes as shown in Figure 417. For this session and the next two sessions, use the same security parameters. Click **OK** and create two more sessions using this session as a model. Note that from the client point of view it does not matter if you are using RACF to protect your system. The client only has to present its certificate. Using this session and the next two, the certificate of the CA that has signed the client's certificate has to be stored in the key database.

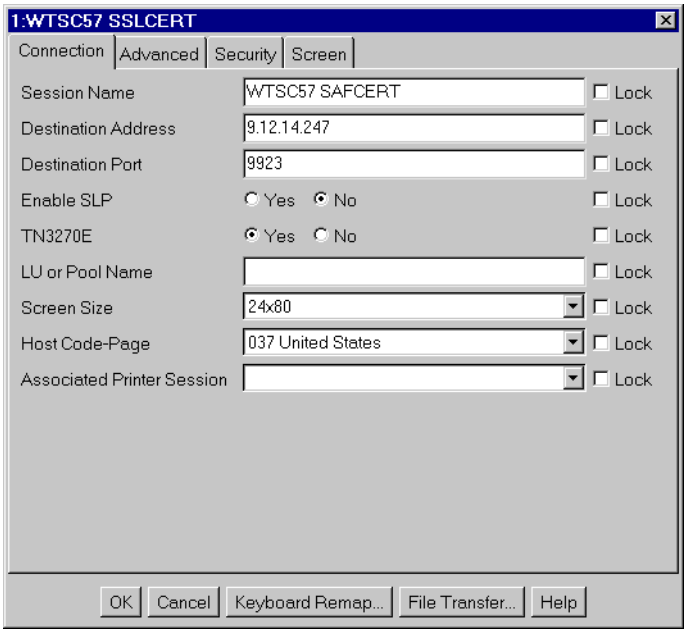


Figure 418. HOD administration - SSL client authentication with RACF

Create the connection shown in Figure 418 and click **OK**. For this session the client certificate must be stored in the RACF database.

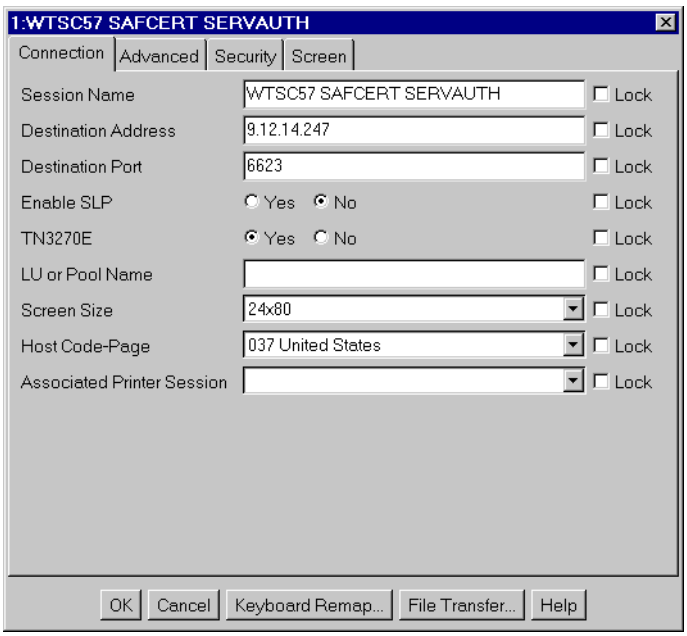


Figure 419. HOD administration - SSL client authentication with RACF (SERVAUTH)

Create the connection shown in Figure 419 and click **OK**. For this session the client certificate must be stored in the RACF database, and the user ID connected to this certificate must be authorized to use the server port.

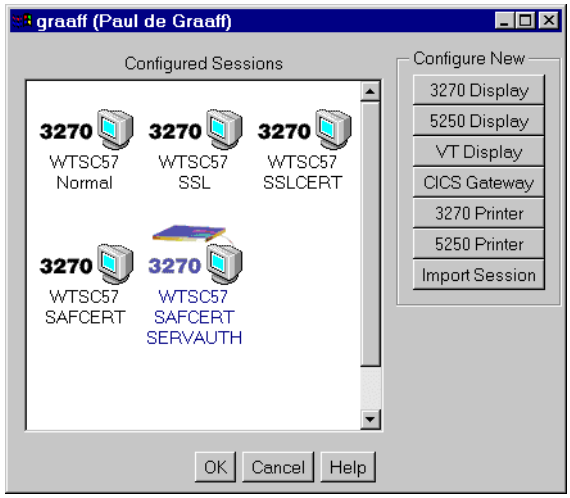


Figure 420. HOD administration - all sessions defined

Now you have five sessions defined, one for each Telnet server.

10.1.3.6 Working with the client certificate

Before starting the sessions you have to define a client certificate. We used two client certificates: a self-signed certificate and one obtained from VeriSign (a temporary certificate validated only for 60 days).

We created the client self-signed certificate using the HOD V4 Windows Client Certificate Management Utility (`IKEYMAN`). PCOMM (Personal Communications) Version 4.3.1 also has a Certificate Management Utility and can be used to handle the certificates.

Start the HOD V4 Certificate Management Utility by selecting a **Key Database File** and **New**. Choose a name for this database, click **OK**, then choose a password for this database and click **OK**. The process is very similar to the `GSKKYMAN` and the `IKEYMAN` utility used on OS/390.

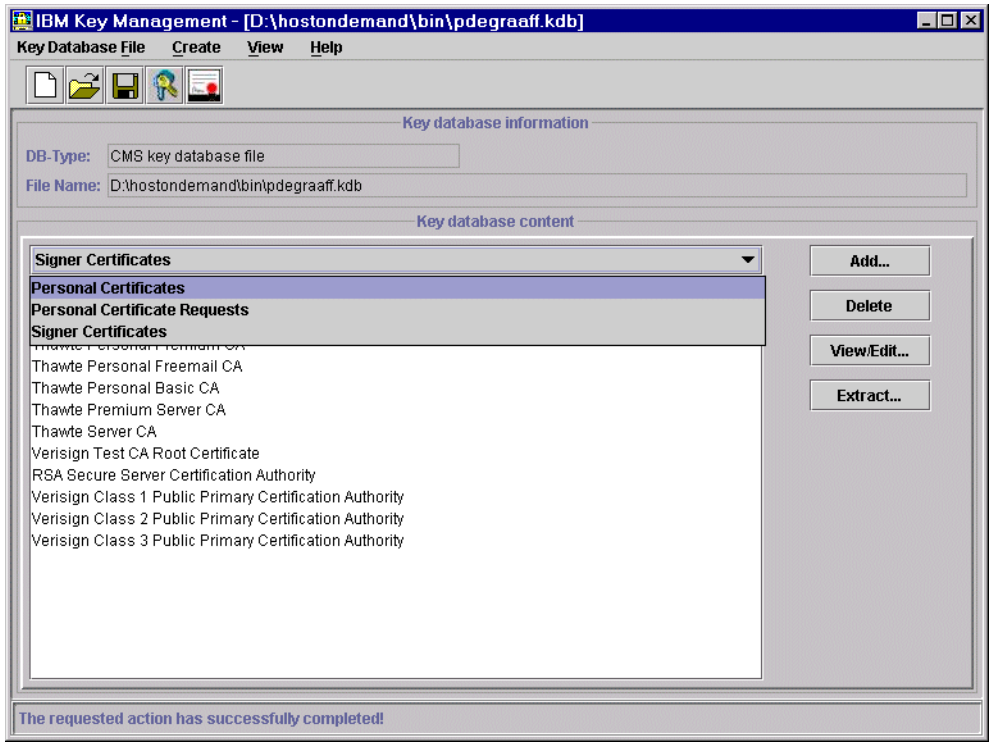


Figure 421. Selecting to work with personal certificates

Then select to work with **Personal Certificates** and click **New-Self Signed**.

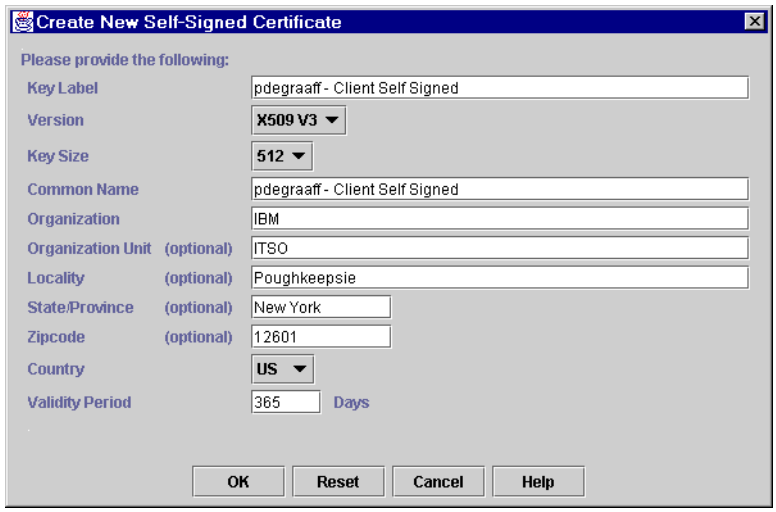


Figure 422. Defining a client self-signed certificate

Create the certificate as shown in Figure 422 and click **OK**.

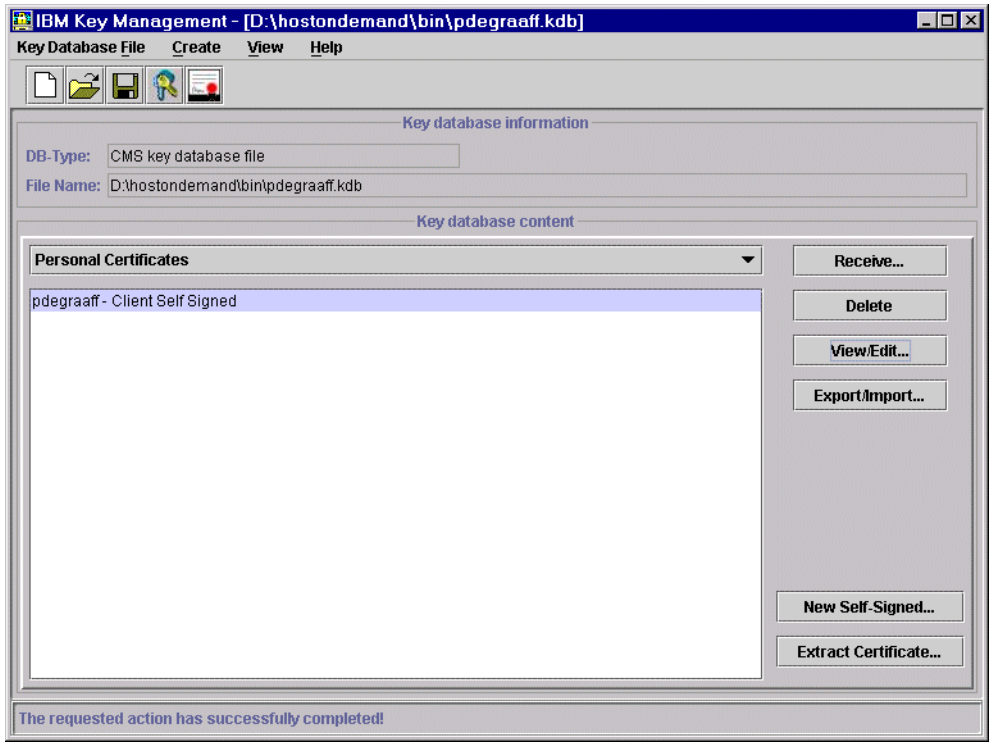


Figure 423. Exporting the client certificate

Now select the Certificate and click **Export/Import**. Choose the PKCS12 type, choose a file name and click **OK**. You will be asked to input a password in order to protect the target PKCS12 file.We used clientselfsigned.p12 as a file name here.

Next, on the same window, click **Extract Certificate ...**. Choose **Base-64 encoded ASCII data**, choose a file name and click **OK**. We used clientselfsigned.crt as a file name here.

Now you have three files in the directory specified in the previous procedures: a key database, a PKCS #12 formatted file, and a file that contains the certificate exported from the key database.

Next you have to transfer the exported certificate to OS/390 in ASCII mode (do not transfer in binary format). Copy this file to the same directory where you have the server key database file. Then use the following steps to store this certificate in the key database as a CA certificate (remember it is a self-signed certificate and it also acts as a CA certificate):

```
GRAAFF @ SC57:/u/graaff>ls -all
drwxr-xr-x   6 STC      SYS1        8192 May  4 16:45 .
drwxr-xr-x  21 STC      SYS1        8192 Mar 22 13:17 ..
-rw-r--r--   1 STC      SYS1        1217 Apr 27 09:38 CustomizedCAs.class
-rw-r-----   1 STC      SYS1         778 May  4 16:45 clientselfsigned.crt
-rw-r--r--   1 STC      SYS1         611 Apr 27 09:00 telnet.crt
-rw-r--r--   1 STC      SYS1       65080 Apr 27 09:00 telnet.kdb
-rw-------   1 STC      SYS1         129 Apr 27 09:06 telnet.sth
GRAAFF @ SC57:/u/graaff>gskkyman


            IBM Key Management Utility

Choose one of the following options to proceed.

   1  - Create new key database
   2  - Open key database
   3  - Change database password

   0  - Exit program

Enter your option number: 2
Enter key database name or press ENTER for "key.kdb": telnet.kdb
Enter password for the key database.......>


            Key database menu

Current key database is /u/graaff/telnet.kdb

     1  - List/Manage keys and certificates
     2  - List/Manage request keys
     3  - Create new key pair and certificate request
     4  - Receive a certificate issued for your request
     5  - Create a self-signed certificate
     6  - Store a CA certificate
     7  - Show the default key
     8  - Import keys
     9  - Export keys
    10  - List all trusted CAs
    11  - Store encrypted database password

     0  - Exit program

Enter option number (or press ENTER to return to the parent menu): 6
Enter certificate file name or press ENTER for "cert.arm": clientselfsigned.crt
Enter a label for this key...............> pdegraaff - client self signed

Please wait while certificate is stored...

Your request has completed successfully, exit gskkyman? (1 = yes, 0 = no) Ý0¨: 1
GRAAFF @ SC57:/u/graaff>
```

Figure 424. Storing the client self-signed certificate in the server key database

Using the VeriSign certificate we will extract the issuer certificate, the certificate authority, and install it in the server key database file. To do that you have to export the certificate using the certificate management functions of your browser. In our environment, we used Internet Explorer.

When you ask for a temporary certificate from VeriSign you will have to follow an installation process. After you finish the certificate installation process the certificate will be installed in your browser.

To export the certificate from Internet Explorer go to **Internet Options -> Content -> Personal**, select the certificate and click **Extract**. This process may vary depending on the version of your browser. But, after you finish the export process, you will have a P12 format file on your disk. In our case we have a file named verisign.p12 on our disk. This file will be used to extract a Base-64 certificate format and install it in the Telnet server database file.

To extract the issuer certificate we will use the HOD client downloaded from OS/390. Go to your browser and do the following steps:

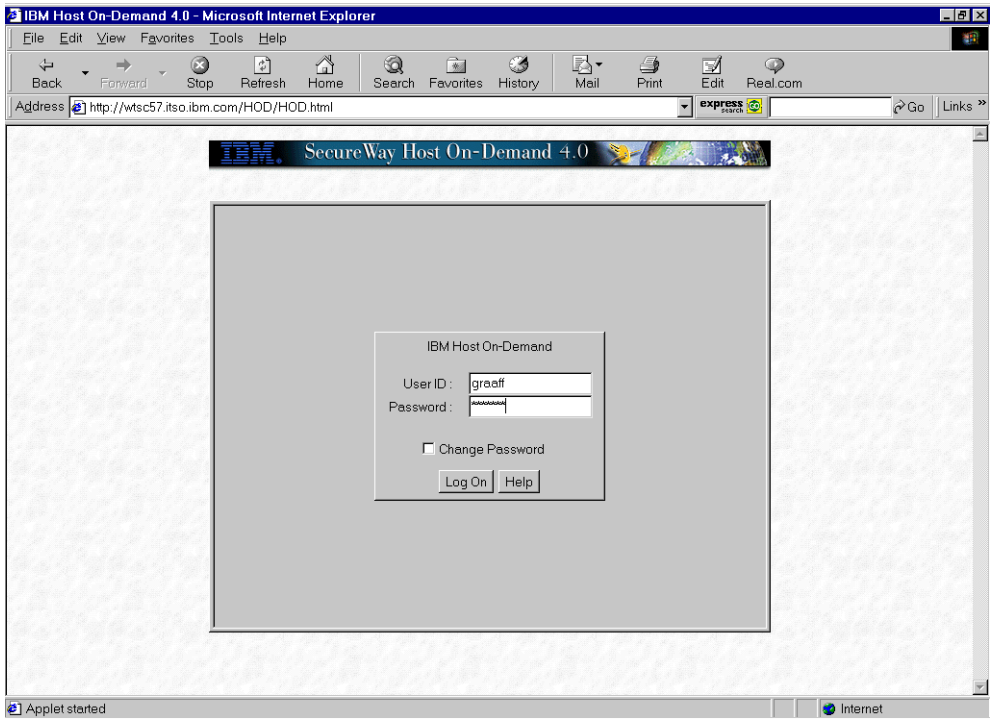


Figure 425. Extracting VeriSign issuer certificate - logging on

Fill in the fields using a user ID and password you created earlier and click **Log On**.

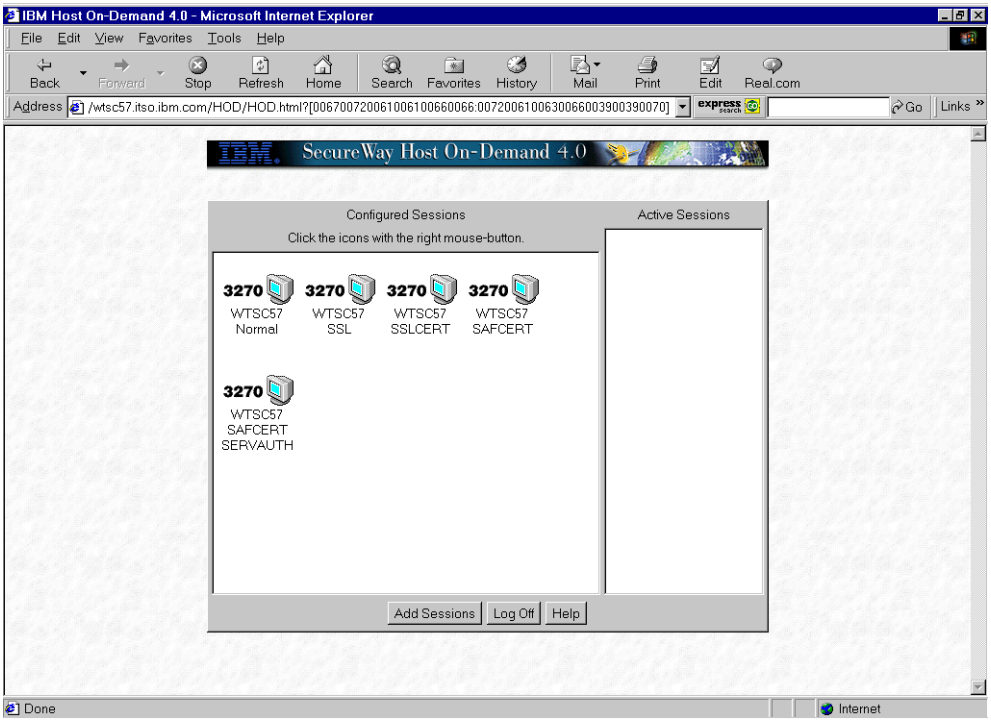


Figure 426. Extracting VeriSign issuer certificate - starting an SSL session

Now double-click any session that requires client authentication. We chose the SSLCERT session.

Figure 427. Extracting VeriSign issuer certificate - server requesting certificate

Fill in the fields as shown above. When you created the P12 format file you defined a password for this file. Use this password to open the client certificate. Click **View Certificate** to open the file.

Figure 428. Extracting VeriSign client certificate - looking at client certificate

Click **Extract** to extract this client certificate now in binary format.

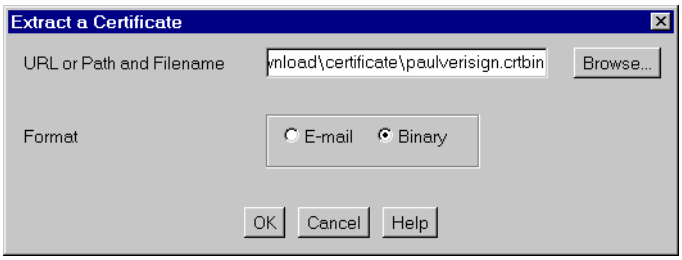


Figure 429. Extracting VeriSign client certificate - extracting to an ASCII format file

This certificate will be used later for RACF definition. Select the binary format and choose a file name for the file, then click **OK**. This returns you to the Security Information window, as shown in Figure 430.

Figure 430. Extracting VeriSign issuer certificate - looking at client certificate

Next, click **Show Issuer Certificate**.

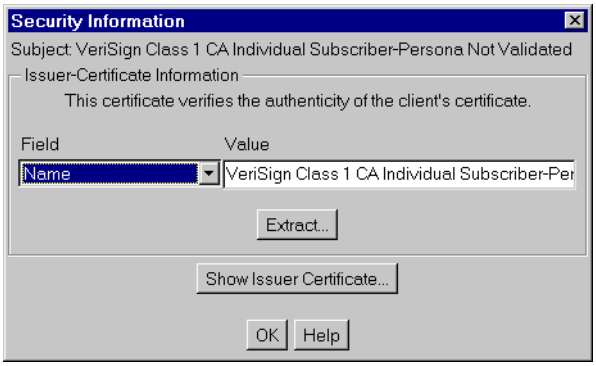


Figure 431. Extracting VeriSign issuer certificate - looking at issuer certificate

Click **Extract**.

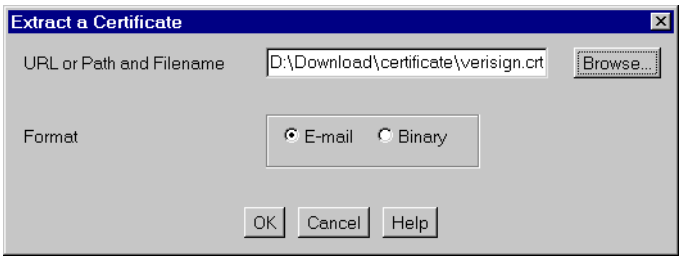


Figure 432. Extracting VeriSign issuer certificate - extracting the file

Extract the file using ASCII format (E-mail option in this case). Then click **OK**.

Transfer both files you have extracted to OS/390. The VeriSign issuer certificate in ASCII mode and the VeriSign client certificate in binary mode have to reside in the same directory as the server key database file. You have to repeat the same procedures as shown in Figure 424 on page 359 using the VeriSign issuer certificate file (verisign.crt) as input. Note that the certificate of the CA that has signed the client's certificate has to be stored in the key database.

Note: Each time you update the key database file you have to recycle the Telnet server.

10.1.3.7 Starting TN3270 sessions with SSL client authentication

Now you can start any of the first three sessions using both certificates. Log on with your user ID again as you did in Figure 425 on page 360 and start the first three sessions. The first two sessions will start automatically, but the third session (SSLCERT) will ask you for a certificate. You can choose either the self-signed or VeriSign certificate. Start two SSLCERT sessions using both certificates. When you receive the screen as shown in Figure 427 on page 361 just choose your own certificate, fill in the password for the p12 file and click **OK** to start the session. The HOD client menu (where you start the sessions) will look similar to Figure 433 on page 364.

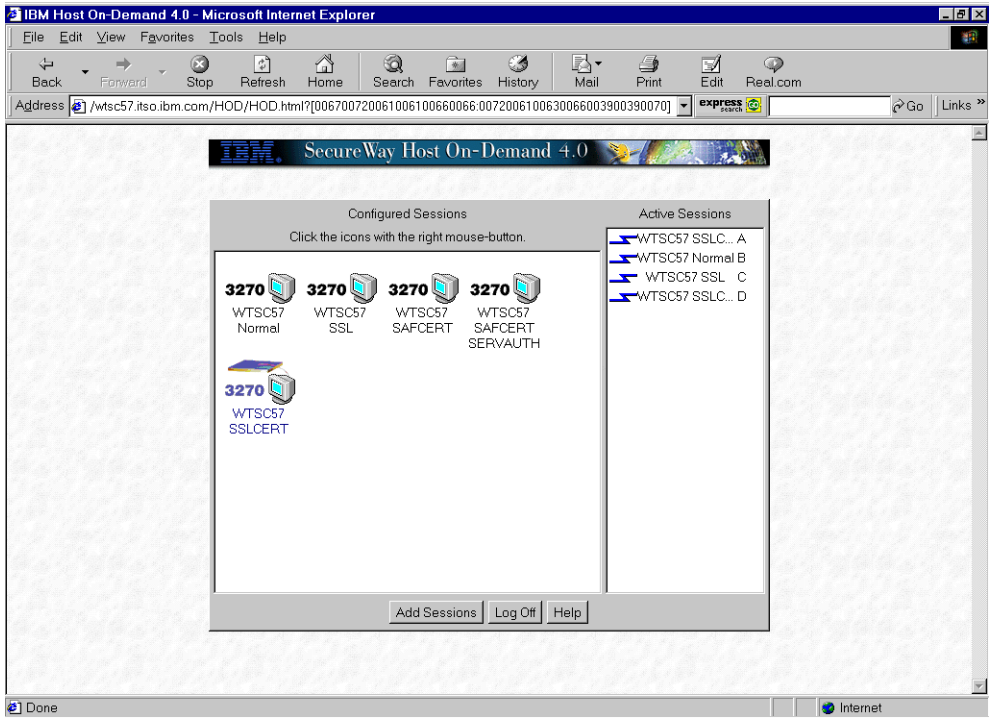


Figure 433. HOD client - SSL sessions started

Now you can check the Telnet servers at the OS/390 console, as shown in Figure 434 on page 365.

```
D TCPIP,ITCPIP,T,CONN
EZZ6064I TELNET CONNECTION DISPLAY 013
         EN                                          TSP
CONN     TY IPADDR..PORT           LUNAME   APPLID   PTR LOGMODE
-------- -- --------------------- -------- -------- --- --------
----- PORT:    23  ACTIVE   BASIC           PROF: CURR
00000028    9.12.2.122..1770       TCP57002 SC57TS03 TAE D4C32XX3
----- PORT:  7723  ACTIVE   SECURE          PROF: CURR
0000002B DS 9.12.2.122..1790       TCP57003 SC57TS04 TAE D4C32XX3
----- PORT:  8823  ACTIVE   SECURE          PROF: CURR
000000AC DS 9.12.2.122..1971       TCP57004 SC57TS05 TAE D4C32XX3
00000025 DS 9.12.2.122..1754       TCP57001 SC57TS02 TAE D4C32XX3
D TCPIP,ITCPIP,T,CONN,CONN=2B
EZZ6065I TELNET CONNECTION DISPLAY 017
  CONN: 0000002B    IPADDR..PORT: 9.12.2.122..1790
  HOSTNAME: NO HOSTNAME
  CONNECTED: 11:05:32  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  7723  ACTIVE   SECURE          ACCESS: SECURE  DS
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57003      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS04
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**
D TCPIP,ITCPIP,T,CONN,CONN=AC
EZZ6065I TELNET CONNECTION DISPLAY 019
  CONN: 000000AC    IPADDR..PORT: 9.12.2.122..1971
  HOSTNAME: NO HOSTNAME
  CONNECTED: 11:23:24  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  8823  ACTIVE   SECURE          ACCESS: SECURE  DS SSLCERT
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57004      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS05
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**
```

Figure 434. Checking Telnet connection (client certificate without RACF)

10.1.3.8 Defining RACF profiles for SAF certificate verification

If you want SAF certificate verification support (CLIENTAUTH SAFCERT), you must also do the following:

- Ensure that the installed SAF product supports client certificates.
- Define RACDCERT as an authorized TSO command in the IKJTSOxx member.
- Ensure that your client certificates have been defined to your security product and marked as trusted.
- Consider RACLISTing the DIGTCERT class (for best performance).
- If you are allowing users to self-register their certificates with RACF, users need (at the minimum) read authority to IRR.DIGTCERT.ADD in the facility class.

If you plan to provide port-specific profiles in your security product to specify the users that can access a particular TN3270 port, do the following:

- Ensure your security product has the SERVAUTH class defined (RACF provides this support in V2R8).
- Add the profile and access list information to your security product and ensure the profile name follows the following format:

  `EZB.TN3270.sysname.tcpname.PORTnnnnn`

  where `nnnnn` is the port number with leading zeros specified.

  Note: The minimum access level that users added to the access list must be granted is READ access.
- Ensure the SERVAUTH class is active.
- Refresh the SERVAUTH RACLIST after changes to the profiles.

We implemented those two levels of the access control for the next two HOD sessions as described in the following paragraphs.

First you have to transfer the certificate files to an MVS data set using the `OGET` TSO command:

```
oget '/u/graaff/verisign.crt' 'graaff.verisign.crt'
oget '/u/graaff/clientselfsigned.crt' 'graaff.selfsign.crt' binary
```

Issue the following RACF commands from the TSO command line, if you have not already defined these:

```
RDEFINE FACILITY IRR.DIGTCERT.ADD UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
PERMIT IRR.DIGTCERT.ADD CLASS(FACILITY) ID(graaff) ACC(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(graaff) ACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

The `RACDCERT` command manages the relationship (maps) between digital certificates and RACF User ID.

Authority to the `IRR.DIGTCERT.function` resource in the FACILITY class allows a user to issue the `RACDCERT` command that is used to install and maintain digital certificates and key rings in RACF. To issue the `RACDCERT` command, users must have one of the following authorities:

- The SPECIAL attribute
- Sufficient authority to resource `IRR.DIGTCERT.function` in the FACILITY class.
  - READ access to `IRR.DIGTCERT.function` to issue the `RACDCERT` command for themselves.
  - UPDATE access to `IRR.DIGTCERT.function` to issue the `RACDCERT` command for others.
  - CONTROL access to `IRR.DIGTCERT.function` to issue the `RACDCERT` command for SITE and CERTAUTH certificates. This authority also has other uses.

For more information about the `RACDCERT` command and the IRR.DIGTCERT.function RACF resources, refer to *OS/390 Security Server*

(RACF) Command Language Reference, SC28-1919, or see Chapter 3, “Digital certificate support enhancements” on page 25.

Now using the RACDCERT command you have to insert the two certificates into the RACF database:

```
RACDCERT ADD('gianca.selfsign.crt') TRUST WITHLABEL('Self Signed')
RACDCERT ADD('gianca.verisign.bincrt') TRUST WITHLABEL('VeriSign')
```

You will see the following message when the request has been completed successfully:

```
IRRD119I Certificate Authority not defined to RACF. Certificate added with
TRUST status.
```

The message indicates that you are adding a certificate to RACF and RACF does not know about the certificate authority that signed the certificate (remember it was self-signed). You can list the certificates in the RACF database using the RACDCERT LIST command:

```
Digital certificate information for user GRAAFF:

  Label: Self Signed
  Status: TRUST
  Start Date: 2000/05/03 16:33:44
  End Date:   2001/05/04 16:33:44
  Serial Number:
     >854C8D2F78E90285<
  Issuer's Name:
     >CN=pdegraaff - Client Self Signed.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New<
     > York.12601.C=US<
  Subject's Name:
     >CN=pdegraaff - Client Self Signed.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New<
     > York.12601.C=US<
  Private Key Type: None
  Ring Associations:
  *** No rings associated ***

  Label: VeriSign
  Status: TRUST
  Start Date: 1998/05/11 20:00:00
  End Date:   2008/05/12 19:59:59
  Serial Number:
     >00D2762E8D140C3D7DB2A8255DAFEE0D75<
  Issuer's Name:
     >OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=<
>US<
Subject's Name:
     >CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated.OU<
     >=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98.OU=Ver<
     >iSign Trust Network.O=VeriSign, Inc.<
Private Key Type: None
Ring Associations:
*** No rings associated ***
```

Figure 435. RACDCERT LIST command

Now you can start two more connections using the SAFCERT Telnet server on port 9923. Go to the HOD client menu and start the SAFCERT session twice.

If you try to start the SAFCERT SERVAUTH session now you will receive an error message like this:

```
ICH408I USER(GRAAFF  ) GROUP(SYS1    ) NAME(PAUL DE GRAAFF      )
  EZB.TN3270.SC57.ITCPIP.PORT06623 CL(SERVAUTH)
  INSUFFICIENT ACCESS AUTHORITY
  FROM ** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

As you can see, you have your certificate in the key database and in the RACF database, but you need another level of security to connect to this specific Telnet server. To activate this permission use the following steps.

First, activate the class SERVAUTH to RACF:

```
SETROPTS CLASSACT(SERVAUTH)
```

Now you have to give permission to access a specific server to the same user ID associated with the client certificate. When the Telnet server receives your certificate it will check if the user ID associated with this certificate has permission to read the SERVAUTH profile for this server. As you can see, this is another level of security you can implement in the Telnet server. Issue the command below to give a user ID permission to access a particular server(port):

```
RDEFINE SERVAUTH EZB.TN3270.SC57.ITCPIP.PORT06623 UACC(NONE)
PERMIT EZB.TN3270.SC57.ITCPIP.PORT06623 CLASS(SERVAUTH) ID(graaff)
    ACC(READ)
SETROPTS RACLIST(SERVAUTH) REFRESH
```

With this command you are giving a specific user permission to access the service 6623 in stack ITCPIP on system SC57. This permission is linked with the certificate by the user ID.

To display the list of users who have the permission to this profile, issue the TSO RACF command shown below:

```
rl servauth EZB.TN3270.SC57.ITCPIP.PORT06623 authuser

CLASS      NAME
-----      ----
SERVAUTH   EZB.TN3270.SC57.ITCPIP.PORT06623

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    GRAAFF          NONE              READ    NO
...
NOTIFY
------
NO USER TO BE NOTIFIED
....
USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
GRAAFF    READ        000000

   ID     ACCESS  ACCESS COUNT  CLASS                   ENTITY  NAME
-------- -------  ------------ -------- ----------------------------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

Figure 436. RLIST RACF command

Now you can start all sessions in the HOD clients configured. You will see the session status in the HOD panel:

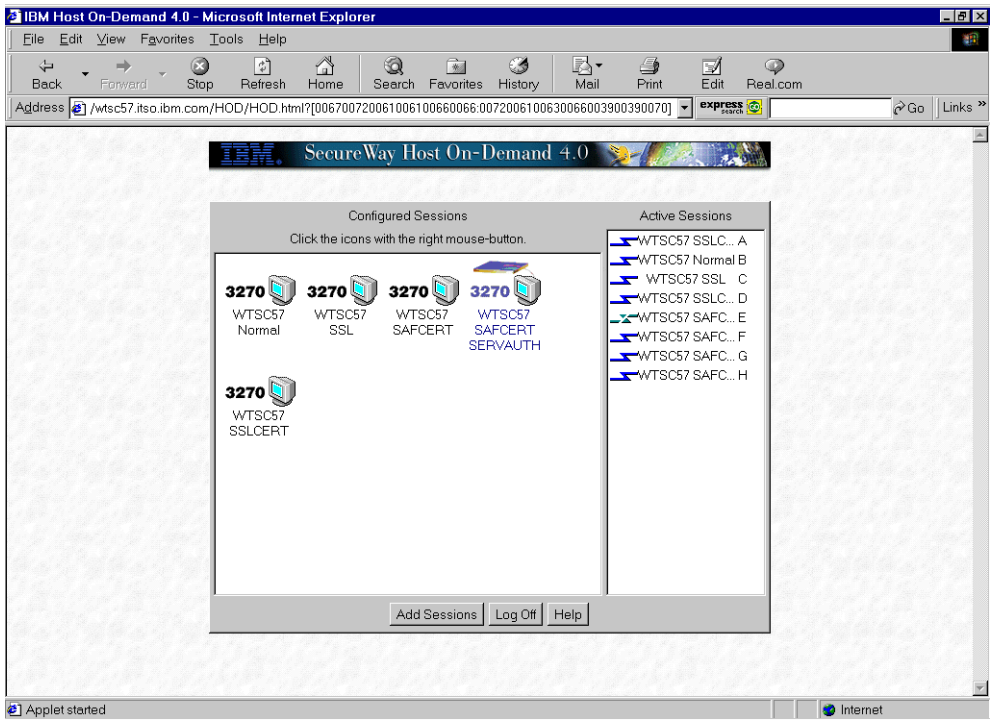


Figure 437. HOD - all sessions activated

Now you have a normal session, an SSL session without client authentication, and six sessions with client authentication using two different certificates and different security levels.

Using the `D TCPIP,,Telnet,CONnection` MVS console command, you will see the status of these connections:

```
D TCPIP,ITCPIP,T,CONN
EZZ6064I TELNET CONNECTION DISPLAY 094
         EN                                          TSP
CONN     TY IPADDR..PORT           LUNAME   APPLID   PTR LOGMODE
-------- -- --------------------- -------- -------- --- --------
----- PORT:    23  ACTIVE   BASIC           PROF: CURR
00000028    9.12.2.122..1770       TCP57002 SC57TS03 TAE D4C32XX3
----- PORT:  6623  ACTIVE   SECURE          PROF: CURR
0000022F DS 9.12.2.122..2615       TCP57008 SC57TS07 TAE D4C32XX3
000001A6 DS 9.12.2.122..2390       TCP57007 SC57TS08 TAE D4C32XX3
----- PORT:  7723  ACTIVE   SECURE          PROF: CURR
0000002B DS 9.12.2.122..1790       TCP57003 SC57TS04 TAE D4C32XX3
----- PORT:  8823  ACTIVE   SECURE          PROF: CURR
000000AC DS 9.12.2.122..1971       TCP57004 SC57TS05 TAE D4C32XX3
00000025 DS 9.12.2.122..1754       TCP57001 SC57TS02 TAE D4C32XX3
----- PORT:  9923  ACTIVE   SECURE          PROF: CURR
0000015F DS 9.12.2.122..2298       TCP57006 SC57TS07 TAE D4C32XX3
00000158 DS 9.12.2.122..2290       TCP57005 SC57TS06 TAE D4C32XX3
D TCPIP,ITCPIP,T,CONN,CONN=28
EZZ6065I TELNET CONNECTION DISPLAY 132
  CONN: 00000028    IPADDR..PORT: 9.12.2.122..1770
  HOSTNAME: NO HOSTNAME
  CONNECTED: 11:05:06  05/05/2000  STATUS: SESSION ACTIVE
  PORT:    23  ACTIVE   BASIC           ACCESS: NON-SECURE
  PROFILE ID: CURR   PROFILE OPTIONS: -------W-----
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57002      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS03
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**
D TCPIP,ITCPIP,T,CONN,CONN=22F
EZZ6065I TELNET CONNECTION DISPLAY 134
  CONN: 0000022F    IPADDR..PORT: 9.12.2.122..2615
  HOSTNAME: NO HOSTNAME
  CONNECTED: 13:01:02  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  6623  ACTIVE   SECURE          ACCESS: SECURE  DS SAFAUTH
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57008      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS07
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**       CERT.USERID: GRAAFF
```

Figure 438. Telnet - all connections active: the first display

```
D TCPIP,ITCPIP,T,CONN,CONN=2B
EZZ6065I TELNET CONNECTION DISPLAY 136
  CONN: 0000002B    IPADDR..PORT: 9.12.2.122..1790
  HOSTNAME: NO HOSTNAME
  CONNECTED: 11:05:32  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  7723  ACTIVE   SECURE          ACCESS: SECURE  DS
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57003      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS04
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**
D TCPIP,ITCPIP,T,CONN,CONN=AC
EZZ6065I TELNET CONNECTION DISPLAY 138
  CONN: 000000AC    IPADDR..PORT: 9.12.2.122..1971
  HOSTNAME: NO HOSTNAME
  CONNECTED: 11:23:24  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  8823  ACTIVE   SECURE          ACCESS: SECURE  DS SSLCERT
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57004      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS05
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**
D TCPIP,ITCPIP,T,CONN,CONN=204
EZZ6065I TELNET CONNECTION DISPLAY 140
  CONN: 00000204    IPADDR..PORT: 9.12.2.122..2540
  HOSTNAME: NO HOSTNAME
  CONNECTED: 12:48:34  05/05/2000  STATUS: SESSION ACTIVE
  PORT:  9923  ACTIVE   SECURE          ACCESS: SECURE  DS SAFAUTH
  PROFILE ID: CURR   PROFILE OPTIONS: -------------
  PROTOCOL: TN3270E  LOGMODE: D4C32XX3 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: TCP57006      TYPE: TERMINAL
    LU/PRTGROUP: *DEFLUS*    HN/IPGROUP: **N/A**
  APPLID: SC57TS09
    DEFAULTAPPL HN/IPGROUP: EXACT IP ADDR
    RESTRICTAPPL USERID: **N/A**       CERT.USERID: GRAAFF
```

Figure 439. Telnet - all connections active: the second display

10.1.3.9 TELNETPARMS encryption parameter

Each of the Telnet FMIDs supports a specific set of encryption algorithms. The ENCRYPTION/ENDENCRYPTION block within the TELNETPARMS allows the selection of a subset of the supported algorithms to use for this port. The ENCRYPTION/ENDENCRYPTION block applies only to a Telnet SECUREPORT that serves SSL V3 clients.

Note: If this keyword is not coded, the Telnet server supports all encryption methods available for the FMID.

To determine what encryption method is in effect when using HOD, use the **Communication** selection option on the main HOD window, and then select the Security option. A window as shown in Figure 440 indicates the encryption method used, in our case SSL_RSA_WITH_DES_CBC_SHA.

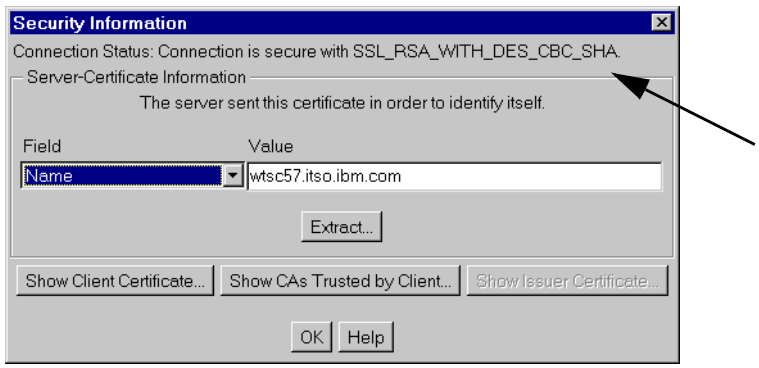


Figure 440. Security information window - indicating encryption method used

The encryption parameter enables us to limit the encryption methods used, for example only use the strongest encryption methods.

In our next example we update our telnetparms block to include the encypt parameter as follows:

```
TELNETPARMS
      SECUREPORT  9923 KEYRING  HFS /u/graaff/telnet.kdb
      ENCRYPT SSL_3DES_SHA ENDENCRYPT
      CLIENTAUTH SAFCERT
ENDTELNETPARMS
```

When we restart the telnet server and try to connect, we fail to set up a connection because the version of HOD we had installed was an international version and did not support 3DES. We then changed the encryption as follows:

```
TELNETPARMS
      SECUREPORT  9923 KEYRING  HFS /u/graaff/telnet.kdb
      ENCRYPT SSL_RC4_MD5_EX ENDENCRYPT
      CLIENTAUTH SAFCERT
ENDTELNETPARMS
```

When we again try to connect, we are now able to establish a “secure” connection, as shown in Figure 441 on page 373.

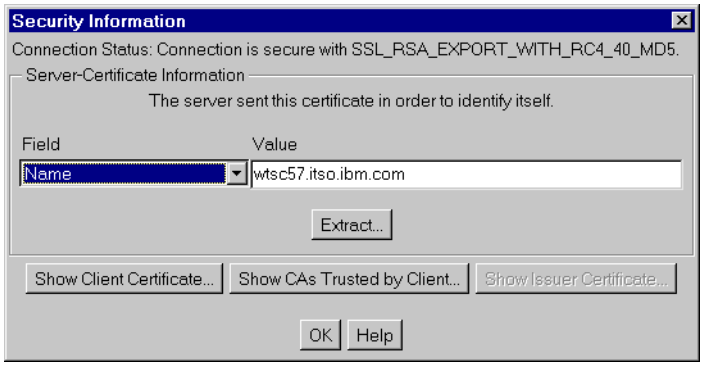


Figure 441. HOD Security Information window showing the secure connection

10.2 Personal Communications

The eNetwork Personal Communications product Version 4.3.1 has SSL support built in to work with the OS/390 TN3270 Server.

However, the SSL support here is server-side only. No client certificate support is currently available.

This section describes how to enable the use of SSL within eNetwork Personal Communications (PCOMM).

10.2.1 SSL setup

The SSL setup with Personal Communication is very straightforward ,and depends on the Certificate Authority in use.

You have to set up a connection, or re-use an existing connection definition, to your OS/390 system, using the **Communication** option on the main window when PCOMM starts up. The Customize Communication panel is then presented, as shown in Figure 442 on page 374.

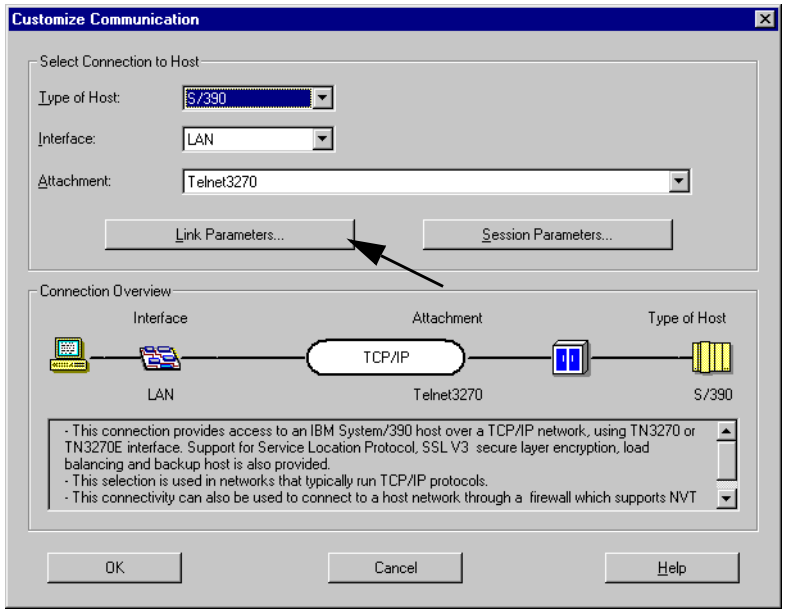


Figure 442. PCOMM - Customize Communication Window

Next click the **Link Parameters** button. The telnet3270 window appears as shown in Figure 443.

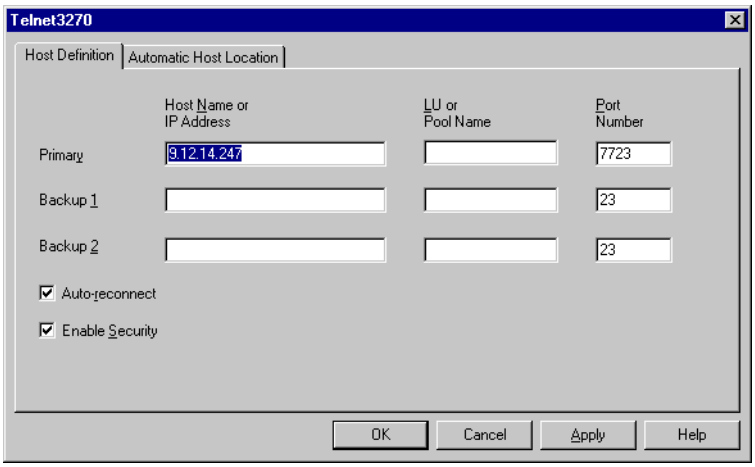


Figure 443. PCOMM - telnet3270 window

The port number we used, 7723 is the one that only supports SSL from a server-side (clientauth none). To enable SSL, check off the **Enable Security** box as shown in Figure 443. Press **OK** to finish the changes on this windows and the next window. Because you have changed the configuration, you receive a warning message indicating that communication will be terminated; click **OK** to continue.

If you have autoconnect enabled, the session will be re-established. In our case it failed because our TN3270 Server used a self-signed certificate that the client (PCOMM) did not know about. PCOMM ships as well with a Certificate Management Utility (IKEYMAN), which we need to use to create a key-ring that contains our self-signed certificate as a CA certificate, so the client (PCOMM) can trust the TN3270 server.

Start the Certificate Management utility, which is located in the IBM Personal Communications folder on your PC (Windows95, WIndows NT), and a window will come up as shown in Figure 444.

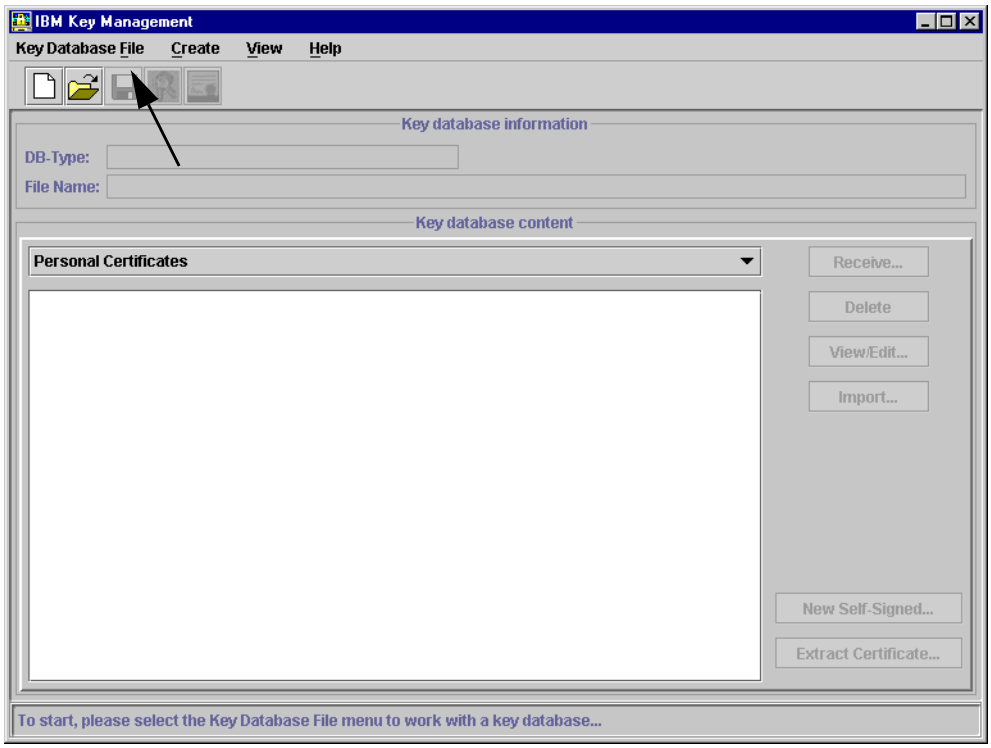


Figure 444. Certificate management utility window

Next we have to create a new key-ring file, where we store the self-signed certificate. From the **Key Database File** menu option, select **New** to create key-ring.

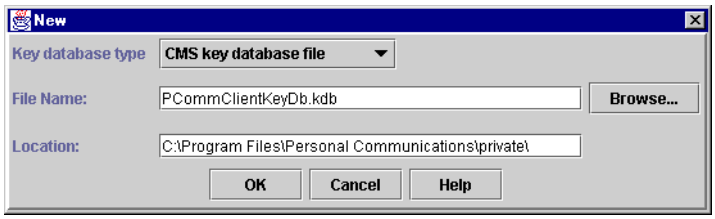


Figure 445. Create new key-ring window

Figure 445 shows the window that allows you create the new key-ring. Press **OK** to accept the default name and location. A window appears where you specify the password of the key-ring and other security options, as shown in Figure 446 on page 376.

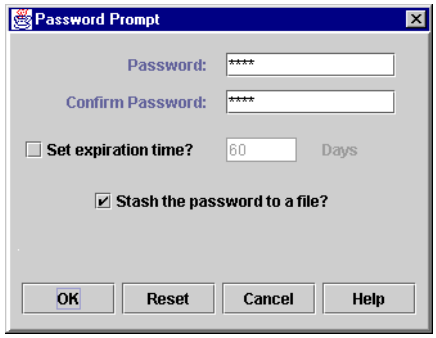


Figure 446. Password Prompt Window

After entering the password twice, you can set the password to expire (optional), and you need to stash the password to a file, for PCOMM to open the key-ring.

Press **OK** and you receive a confirmation window, as shown in Figure 447.

Figure 447. Password confirmation window

The next screen shows the main window, indicating all the trusted Certificate Authorities, as shown in Figure 448.

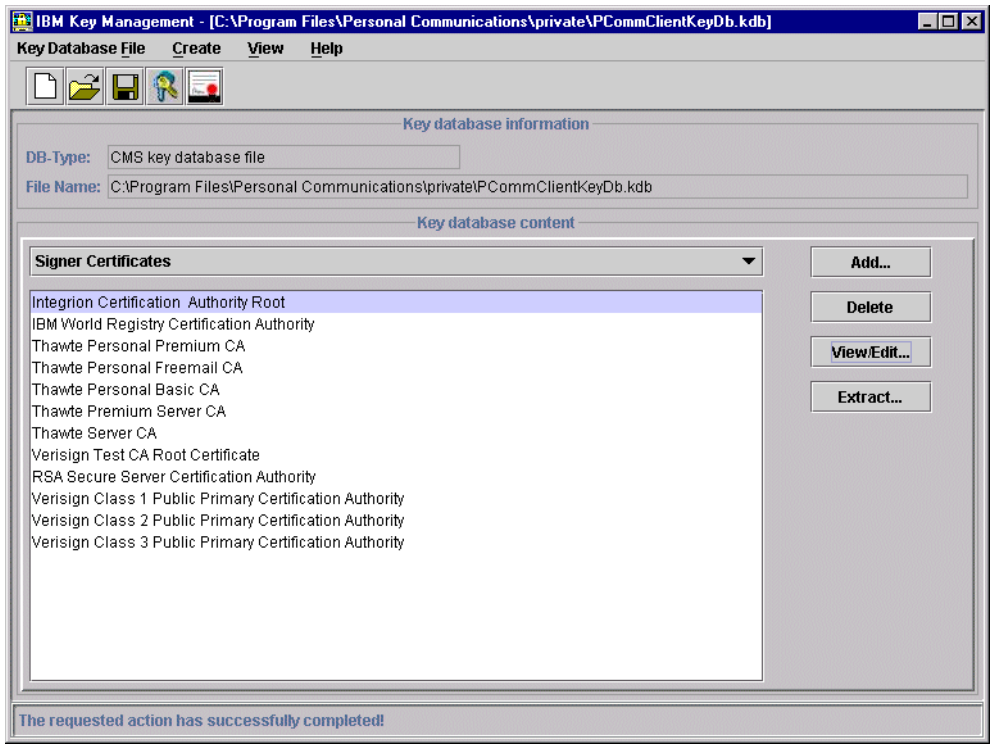


Figure 448. Key Management utility window

The next task is to import the self-signed certificate into the key-ring we created. We use the same file, `telnet.crt`, that we used for our HOD examples since it is the same TN3270 server we are going to (see Figure 390 on page 336). To add the self-signed certificate to the key-ring, click **Add..** on the main window (see Figure 448 on page 376). A window is displayed where you can enter the file type, name and location, as shown in Figure 449.

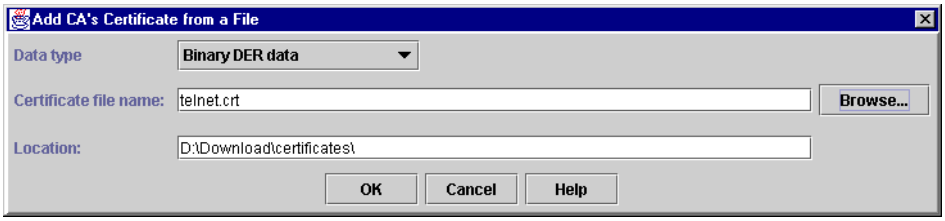


Figure 449. Add CA's certificate from a file window

Note: The file `telnet.crt` is in binary format.

To continue, click **OK**. You receive a window in which to enter a label for the certificate(key) you are adding. Choose a label name that represents the telnet server, as show in Figure 450.

Figure 450. Label Window

The certificate is then added as a CA (signer) certificate, as shown in Figure 451 on page 378.

Note: If you get error messages regarding asn encoding, you might have downloaded the file in text format rather than in binary.

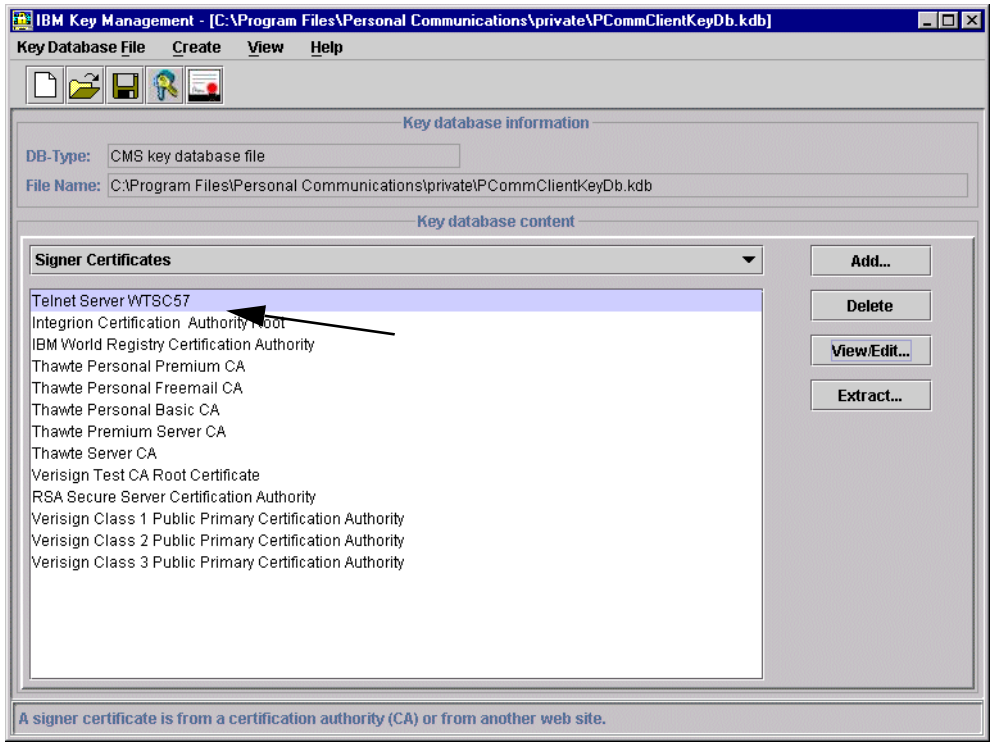


Figure 451. Certificate Management main window

You can close the database and exit out of the utility. Now when you connect to the telnet server, you receive a screen with a secure connection as shown in Figure 452.

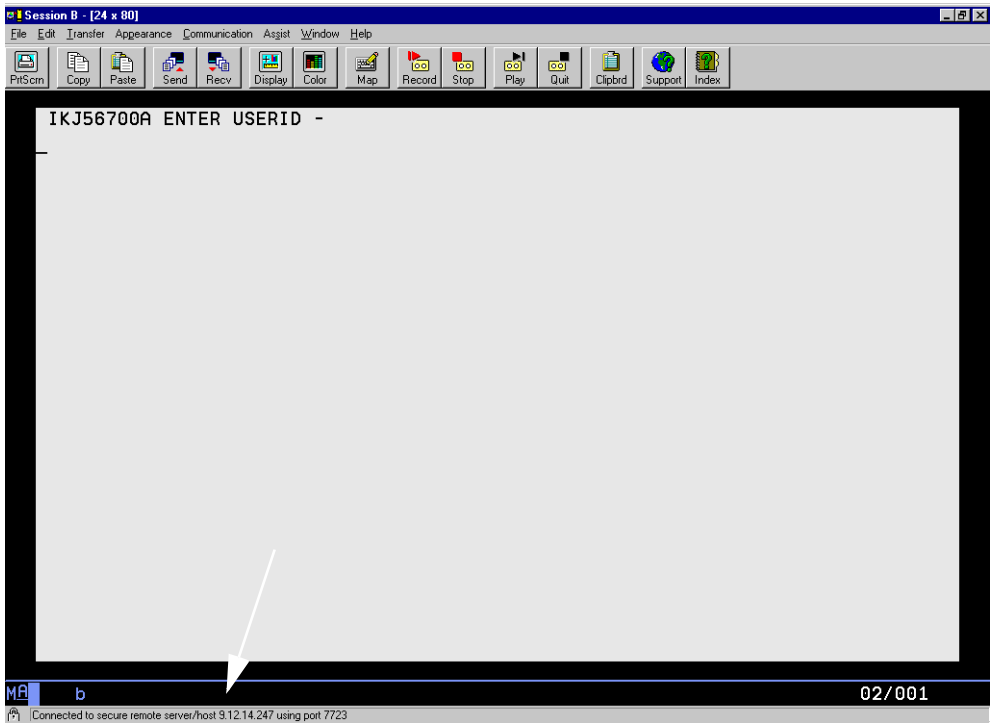


Figure 452. Secure connection with PCOMM

The lock in the lower left corner (similar to the browser's) indicates a secure connection. This is indicated as well by the message at the bottom of the window.

10.3 TCP/IP and ICSF

TCP/IP performs cryptographic operations for Telnet session with SSL. TCP/IP invokes the following ICSF Callable services:

CSFDSV Digital signature verify callable service

CSFPKD PKA decrypt callable service

CSFPKI PKA key import callable service

CSFCKI Clear key import callable service

If you are protecting ICSF Callable services in the RACF CSFSERV class, then the user ID associated with the TCP/IP address space needs to have read access to these services. This is achieved by entering the following RACF command:

```
RDEF CSFSERV CSFDSV UACC(NONE) OWNER(GRAAFF)
RDEF CSFSERV CSFCKI UACC(NONE) OWNER(GRAAFF)
RDEF CSFSERV CSFPKI UACC(NONE) OWNER(GRAAFF)
RDEF CSFSERV CSFPKD UACC(NONE) OWNER(GRAAFF)

PE CSFDSV CLASS(CSFSERV) ID(TCPIPU) ACC(READ)
PE CSFCKI CLASS(CSFSERV) ID(TCPIPU) ACC(READ)
PE CSFPKI CLASS(CSFSERV) ID(TCPIPU) ACC(READ)
PE CSFPKD CLASS(CSFSERV) ID(TCPIPU) ACC(READ)
```

Note: *TCPIPU* is the user ID associated with our TCP/IP address space.

You will need to perform a `RACF SETROPTS RACLIST(CSFSERV) REFRESH` to activate the changes made.

Appendix A. Configuration files used on MVS39

This appendix includes some of the configuration files that were used for the RA39 system in the ITSO Raleigh test installation of SecureWay Communications Server for OS/390 V2R8 IP.

A.1 PROFILE.TCPIP for the TCPIPA stack

```
;****************************** Top of Data ***********************
; Member TCPIP.TCPPARMS(PROF39A)
; ********************************************************************
; For more information about this file, see "Configuring the TCPIP
; Address Space" and "Configuring the Telnet Server" in the
; Configuration Guide.
; ********************************************************************
; You can specify DATASETPREFIX in the PROFILE.TCPIP and
; TCPIP.DATA data sets. If this statement is used in a profile or
; configuration data set that is allocated to a client or a server, then
; that client or server dynamically allocates additional required data
; sets using the value specified for DATASETPREFIX as the data set name
; prefix.  The DATASETPREFIX parameter can be up to 26 characters long,
; and the parameter must NOT end with a period.
;
; For more information please see "Understanding TCP/IP Data Set
; Names" in the Customization and Administration Guide.
;
DATASETPREFIX TCPIP
;
;

TCPCONFIG
;  INTerval  5             ; In minutes - Keep alive packet 0-35791
;  RESTRICTLowports
   UNRESTRICTLowports
   TCPSENDBfrsize  16384   ; Range is 256-256K - Default is 16K
   TCPRCVBufrsize  16384   ; Range is 256-256K - Default is 16K
   SENDGARBAGE FALSE       ; Packet contains no data
;  SENDGARBAGE TRUE        ; Packet contains 1 byte of random data

UDPCONFIG
;  RESTRICTLowports
   UNRESTRICTLowports
   UDPCHKsum               ; Do checksum
;  NOUDPCHKsum             ; Don't do checksum
   UDPSENDBfrsize 16384    ; Range is ???-???K (Default is 16K)
   UDPRCVBufrsize  16384   ; Range is ???-???K (Default is 16K)
;  UDPQueuelimit           ; Limit inbound UDP Queue ????
;  NOUDPQueuelimit         ; Do not Limit inbound UDP Queue ????
;
; ********************************************************
IPCONFig
 ARPTO  1200           ; In seconds
;CLAWUSEDoublenop      ; Applies only to first-level MVS systems
 DATAGRamfwd
;NODATAGRamfwd
```

```
;FIREWALL
;NOSOURCEVIPA
 SOURCEVIPA
;NOVARSUBNETTING       ; For RIPV1
 VARSUBNETTING          ; For RIPV2
;NOSYSPLEXRouting
 SYSPLEXRouting
 IGNORERedirect
 REASSEMBLytimeout   15 ; In seconds
 STOPONclawerror
 TTL    60              ; In seconds, but actually Hop count
 MULTIPATH
; NODYNAMICXCF
 DYNAMICXCF 172.16.233.39 255.255.255.0 1
; ---------------------------------------------------------------------
SACONFIG  COMMUNITY MVSsubagent
 ENABLED
 AGENT 161
 SETSENABLED
; OSASF 760
;ATMENABLED
; SETSDISABLED
; ---------------------------------------------------------------------
;
; AUTOLOG the following servers.
;

 AUTOLOG 1
;   FTPD    JOBNAME FTPD1      ; FTP Server
;   NAMED   JOBNAME NAMED1     ; Domain Name Server
;   FTPD JOBNAME FTPD1   ; FTP Server
    OMPROUTA             ; OSPF   Server
;   WEBSRV               ; DOMINO
    SYSLOGD JOBNAME SYSLOGD1 ; SYSLOG daemon
;   GOWLMRG1             ; sysplex test server
 ENDAUTOLOG

;
; ---------------------------------------------------------------------
;
; Reserve ports for the following servers.
;
; NOTES:
;
;    A port that is not reserved in this list can be used by any user.
;    If you have TCP/IP hosts in your network that reserve ports
;    in the range 1-1023 for privileged applications, you should
;    reserve them here to prevent users from using them.
;
;    The port values below are from RFC 1060, "Assigned Numbers."
;

PORT
     7 UDP MISCSERV           ; Miscellaneous Server
     7 TCP MISCSERV
     9 UDP MISCSERV
     9 TCP MISCSERV
    19 UDP MISCSERV
```

```
    19 TCP MISCSERV
    20 TCP OMVS      NOAUTOLOG ; FTP Server
    21 TCP FTPD1               ; FTP Server
    21 TCP FTPDA1              ; FTP Server
;   23 TCP INTCLIEN            ; Telnet Server
    25 TCP omvs                ; SMTP Server
    53 TCP omvs                ; Domain Name Server - Parent Process
    53 UDP omvs                ; Domain Name Server - Parent Process
    80 TCP WEBSRV              ; Domino webserver
   111 TCP OMVS                ; Portmap Server
   111 UDP OMVS                ; Portmap Server
   135 UDP LLBD                ; NCS Location Broker
   161 UDP OSNMPD              ; SNMP Agent
   162 UDP OMVS                ; osnmp command (SNMP Manager)
 16200 UDP OMVS                ; osnmp command (SNMP Manager)
   443 TCP OMVS                ; Domino webserver
   512 TCP RSHDA               ; Remote Execution Server
   514 TCP RSHDA               ; Remote Execution Server
   514 UDP OMVS                ; SYSLOG daemon
   515 TCP T03ALPD             ; LPD Server
;  520 UDP T39AROU             ; RouteD Server
;  520 UDP OMVS               ; RouteD Server
   580 UDP NCPROUT             ; NCPROUTE Server
   750 TCP MVSKERB             ; Kerberos
   750 UDP MVSKERB             ; Kerberos
   751 TCP ADM@SRV             ; Kerberos Admin Server
   751 UDP ADM@SRV             ; Kerberos Admin Server
   760 UDP IOASNMP             ; osa/sf
   760 TCP IOASNMP             ; osa/sf
; 20000 TCP RA03C               ; CICS Sockets
; 20001 TCP RA03C               ; CICS Sockets Server
; 3005 TCP T03AHWS            ; IMS Sockets (OTMA)
; 3012 TCP T03AIMSL           ; IMS Sockets (Listener)

;
; ---------------------------------------------------------------------
;
; Hardware definitions:
;
; NOTE: To use these device and link statements, update the statements
; to reflect your installation configuration and remove the semicolon
;
; ********************************************************
; VIPA Definition  (For V2R5)
; ********************************************************
;
   DEVICE VIPA39A  VIRTUAL     0
   LINK   VIPA39A  VIRTUAL     0     VIPA39A
;  DEVICE OSVIPA9A VIRTUAL     0
;  LINK   OSVIPA9A VIRTUAL     0     OSVIPA9A

;
;
; ********************************************************
; Netfinity VPPA Connection
; ********************************************************
;
;DEVICE DEVNF2    CTC  20C
```

```
;LINK   LINKNF2   CTC  0 DEVNF2
;
;DEVICE netfinity LCS  202 autorestart
;LINK  netfinity  IBMTR    0 netfinity
; ********************************************************
; LCS Definition
; osa ch D8                          Device # 2060-2061
; ********************************************************
 DEVICE TR1    LCS  2060 autorestart
 LINK  TR1    IBMTR    0 TR1
;
; ********************************************************
; LCS Definition
; osa ch D8                          Device # 2064-2065
; ********************************************************
  DEVICE EN1  LCS           2064
  LINK   EN1  ETHEROR802.3  1     EN1
; ********************************************************
; LCS Definition
; 3172 T/R attach - Rel Adapter 0 - Device # 320-321
; ********************************************************
;DEVICE ICP1 LCS  340 AUTORESTART
;LINK ICP1 IBMTR    0 ICP1
;
;DEVICE ICPV LCS  348 AUTORESTART
;LINK ICPV IBMTR    0 ICPV
; ********************************************************
; LCS Definition
; osa 00 fddi                      - Device # 2080-2081
; ********************************************************
;DEVICE FDDI1  LCS 2080 AUTORESTART
;LINK FDDI1   FDDI     0 FDDI1
;
; ********************************************************
; CLAW Definition
; 3172 T/R attach - Rel Adapter 0 - Device # 344-345
; Second level MVS under VM should be defined as V=R
; ********************************************************
;DEVICE iccp CLAW 350 HOST PSCA NONE 20 20 4096 4096 autorestart
;LINK iccp IP 0  iccp
;
; xcf definition to ra03
;DEVICE RA03M MPCPTP autorestart
;LINK RA03M  MPCPTP RA03M
;=========================================================
;DEVICE D2216a LCS  2c0 AUTORESTART
;LINK LD2216a IBMTR 2 D2216a
;=========================================================
; XCF DEFINITION TO RA28
; DEVICE RA28M MPCPTP AUTORESTART
; LINK RA28M  MPCPTP RA28M

; ********************************************************
; LCS Definition
; 2216 EN attach - Rel Adapter 0 - Device # 300-301
; ********************************************************
;DEVICE d2216 LCS  300 autorestart
;LINK ld2216 ETHERNET 0 d2216
```

```
;
; ************************************************************
; *          3172-3 2nd floor                                *
; *                                                          *
;DEVICE DEVEN1  LCS           302           **********************
;LINK   EN1     802.3         0     DEVEN1
;DEVICE DEVEN1  LCS           2026
;LINK   EN1     ETHEROR802.3  1     DEVEN1
;DEVICE EN1  LCS           2026
;LINK   EN1     ETHEROR802.3  1     EN1
;DEVICE EN2  LCS           2064
;LINK   EN2     ETHEROR802.3  1     EN2
 ;

; ********************************************************
; ATM OSA Definition
; ********************************************************
; DEVICE atm  ATM
; LINK atm ATM atm
;ATMLIS    LIS1     172.16.21.0 255.255.255.0
;; DFLTMTU  8000  INACTVTO  0  MINH  0
;; CEAGE  300  ARPRETR  6  ARPTO  4  PEAKCR  100
;DEVICE    OSARA03    ATM     PORTNAME   OSA58  ENABLEIN   NOAUTOR
;LINK      LATM1      ATM     OSARA03    LIS  LIS1
;LINK      LATM2      ATM     OSARA03
;;
;;
;ATMPVC  PVC300   LATM2
;;
;ATMARPSV  ARPSV1  LIS1  SVC  172.16.21.10 NSAP
;39999999999999000099990101400082100000000
;
; ********************************************************
; 31.8 by luca and cristina for 2216 connection via escon.
;4970341525543207999998010108005A99814100
; MPC Definition
; APAR PQ04890 for OS/390 R3
; VTAM requirement for TRLE Definition in VTAM VBUILD=TRL
; The DEVICE name must match the TRLE name in VTAM
; The TRLE entry must have MPCLEVEL=HPDT (default)
; Support is for ESCON and 2216 (future)
; ********************************************************
;  DEVICE MPC25 MPCPTP AUTORESTART
;  LINK   MPC25 MPCPTP MPC25
; ********************************************************
;DEVICE M392216A MPCPTP AUTORESTART
;LINK   M392216A MPCPTP M392216A
; ********************************************************
;  LINK   to 2216 mae
; ********************************************************
 DEVICE M392216B MPCPTP AUTORESTART
 LINK   M392216B MPCPTP M392216B
; ********************************************************
;  LINK   to 2216 400
; ********************************************************
 DEVICE M392216C MPCPTP AUTORESTART
 LINK   M392216C MPCPTP M392216C
;
```

```
; ********************************************************
; XCF Definition
; APAR PQ04890 for OS/390 R3
; APAR OW26845 for VTAM V4.4
; VTAM requirement for TRLE Definition in VTAM VBUILD=TRL
; The DEVICE name must match the host name of VTAM on other side of XCF
; The DEVICE name must match the TRLE name in VTAM
; The TRLE entry can be dynamically created if XCFINIT=YES in VTAM start
; Support is for Cross-system Coupling Facility
; ********************************************************
;  DEVICE RA28M MPCPTP autorestart
;  LINK RA28M  MPCPTP RA28M
;
;
; LCS1 is a 3172 Model 1 with a Token-Ring and Ethernet adapter.
;

;DEVICE LCS1   LCS           BA0
;LINK TR1   IBMTR    0 LCS1
;LINK ETH1  ETHERNET 1 LCS1

;
; LCS2 is a 3172 Model 2 with a FDDI adapter.
;

;DEVICE LCS2   LCS           BE0
;LINK FDDI1 FDDI     0 LCS2

;
; SNALU0 is an SNA Link.
;

;DEVICE DEVT25A SNAIUCV SNALINK RAPT25A  T03AT25A
;LINK LINKT25A SAMEHOST 0 DEVT25A

;
; DEV3746 is an 3746 IP over Channel attachment
;         To 3746-9x0 IP Router

;DEVICE A37463CDLC9901I15R15t4096 4096
;LINK A3746oCDLC609A3746 Router
;
;DEVICE B3746 CDLC 902 15 15 4096 4096
;LINK B3746 CDLC 0 B3746

;---------------------------------------------------------------------
;Enterprise extender definition
;---------------------------------------------------------------------
;DEVICE IUTSAMEH MPCPTP
;LINK   IUTSAMEH MPCPTP IUTSAMEH
;
; ---------------------------------------------------------------------
;
; HOME Internet (IP) addresses of each link in the host.
;
; NOTE: To use this home statement, update the ipaddress and linknames
; to reflect your installation configuration and remove the semicolon
;
```

```
; HOME Internet (IP) addresses of each link in the host.
HOME
    172.16.232.39 VIPA39A     ; VIPA
    9.24.104.149   TR1        ; osa TO public network
;   9.24.105.73    EN1        ; Ethernet
;   172.16.250.81 OSVIPA9A    ; VIPA
;   192.164.236.1  M392216A   ; MPC TO 2216 A
    172.16.102.39 M392216B    ; MPC TO 2216 mae
;   172.16.105.39 M392216C    ; MPC TO 2216 400
;   172.16.234.39 RA28M       ; XCF TO 28
;   172.16.233.39 RA03M       ; XCF TO 03
;   172.16.221.39 ICP1        ; 3172
;   172.16.221.41 ICPV        ; DUMMY TO GET VIPA UP
;   172.16.220.50 loopback    ; ndr cluster
;   172.16.220.51 loopback    ; ndr cluster
;   172.16.220.52 loopback    ; ndr cluster
;    172.16.228.39    MPC25     ; MPC TO MVS25
;   172.16.223.40 IUTSAMEH    ; EE
;   172.16.116.5   LINKNF2    ; NETFINITY
; commented gateway statement ...morris 13/2/99
;; did two comments instead of one
;;GATEWAY
;;  9         =           TR1         4000      0.255.255.0  0.24.104.0
;;  9         9.24.104.1  TR1         4000      0.255.255.0  0.24.105.0
;;  9         9.24.104.1  TR1         4000      0.255.255.0  0.24.106.0
;;  172.16.118 =          netfinity   4000    255.255.255.0  0.168.118.0
; 172.16.116 =          LINKNF2     32768     0
; DEFAULTNET  9.24.104.1  TR1         4000      0

;GATEWAY
; 9         =           TR1         4000      0.255.255.0  0.24.104.0
; 9         9.24.104.1  TR1         4000      0.255.255.0  0.24.106.0
; 9         9.24.104.1  TR1         4000      0
;172.16.228.25   =      MPC25       32678     HOST

; ---------------------------------------------------------------------
;
; The PRIMARYINTERFACE statement is used to specify which interface
; is the primary interface.
;
; If PRIMARYINTERFACE is not specified, then the first link in the HOME
; statement is the primary interface, as usual.
;
; NOTE: To use this primary statement, update the and linkname
; to reflect your installation configuration and remove the semicolon
;

;  PRIMARYINTERFACE tr1
;  PRIMARYINTERFACE ICP1

;
; ********************************************************************-
;
; IP routing information for the host.  All static IP routes should
; be added here.
;
; NOTE: To use this GATEWAY statement, update the addresses and links
```

```
; to reflect your installation configuration and remove the semicolon
;
; ************************************************************************-
;INCLUDE TCP.TCPPARMS(PR39AGW)
; ------------------------------------------------------------------------
;
; orouted Routing Information
;
; if you are using orouted, comment out the GATEWAY statement and
; update the BSDROUTINGPARMS statement to reflect your installation
; configuration and remove the semicolon
;  if you are useng omproute, none is needed
;
; ************************************************************************-
;INCLUDE TCP.TCPPARMS(PR39ABSD)
; ************************************************************************-
;
; Use TRANSLATE to specify the hardware address of a specific IP
; address.  See the Customization and Administration Guide for more
; information.
;

;TRANSLATE
; A null translate statement issues the warning message EZZ0323I

;
; ------------------------------------------------------------------------
;
; Turn off all tracing.  If tracing is to be used, change the following
; line.  To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1
;

ITRACE OFF

;
; ------------------------------------------------------------------------
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks.  If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; ------------------------------------------------------------------------
;
;  Dynamic VIPA Definitions
;
VIPADynamic
;  VIPABackup   10   172.16.240.3
   VIPABackup   80   172.16.240.28
;  VIPABackup   30   172.16.240.40
   VIPADEFine   255.255.255.192 172.16.240.39
   VIPARange    DEFINE  255.255.255.192  172.16.240.192
;  VIPARange    DEFINE  255.255.255.192  172.16.240.200
ENDVIPADYNAMIC
;
GLOBALCONFIG
   TCPIPSTATISTICS
; ------------------------------------------------------------------------
;
```

```
;  the VTAM parameters are in telnet3a
;
 INCLUDE TCPIP.TCPPARMS(TELN&SYSCLONE.A)
;;;TRUNC 72
; ---------------------------------------------------------------------
;
; Start all the defined devices.
;
; NOTE: To use these START statements, update the device name
; to reflect your installation configuration and remove the semicolon
;

; Start all the defined devices.
;START RA03M               ; XCF TO RA03
;START RA28M               ; XCF TO RA28
 START TR1                 ; OSA
;START EN1                 ; Ethernet
;START ICP1                ; 3172
;START ICPV                ; 3172
;START IUTSAMEH            ; EExtender
 START M392216b            ; mpc to 2216 mae
;START M392216c            ; mpc to 2216 mae
;START d2216a              ; 2216 Virtual Token Ring
;START DEVNF2              ; VPPA to netfinity
;START netfinity           ; VPPA to netfinity
; START MPC25
```

A.2 TELN39A (included member for PROFILE.TCPIP)

```
;  TRUNC 72
;******************************** Top of Data ***********************
; Member TCPIP.TCPPARMS(teln39a) telnet
; ********************************************************************
TELNETPARMS
;TESTMODE
    PORT 23
    INACTIVE 0
    TIMEMARK 7200
    SCANINTERVAL 300
    SMFINIT STD
    SMFTERM STD
    WLMCLUSTERNAME TNRAL ENDWLMCLUSTERNAME
ENDTELNETPARMS
;
BEGINVTAM
PORT 23 223
;   ; Define logon mode tables to be the defaults shipped with the
;   ; latest level of VTAM
  TELNETDEVICE 3277     DSILGMOD          ; 24 x 80   old model 2
  TELNETDEVICE 3278-2   D4B32782,SNX32702 ; 24 x 80
  TELNETDEVICE 3278-2-e NSX32702,SNX32702 ; 24 x 80
  TELNETDEVICE 3278-3   D4B32783,SNX32703 ; 32 x 80,  primary 24 x 80
  TELNETDEVICE 3278-3-e NSX32703,SNX32703 ; 32 x 80,  primary 24 x 80
  TELNETDEVICE 3278-4   D4B32784,SNX32704 ; 43 x 80,  primary 24 x 80
  TELNETDEVICE 3278-4-e NSX32704,SNX32704 ; 43 x 80,  primary 24 x 80
  TELNETDEVICE 3278-5   D4B32785,SNX32705 ; 27 x 132, primary 24 x 80
  TELNETDEVICE 3278-5-e NSX32705,SNX32705 ; 27 x 132, primary 24 x 80
```

```
  TELNETDEVICE 3279-2   D4B32782          ; 24 x 80
  TELNETDEVICE 3279-2-e NSX32702          ; 24 x 80
  TELNETDEVICE 3279-3   D4B32783          ; 32 x 80,  primary 24 x 80
  TELNETDEVICE 3279-3-e NSX32703          ; 32 x 80,  primary 24 x 80
  TELNETDEVICE 3279-4   D4B32784          ; 43 x 80,  primary 24 x 80
  TELNETDEVICE 3279-4-e NSX32704          ; 43 x 80,  primary 24 x 80
  TELNETDEVICE 3279-5   D4B32785          ; 27 x 132, primary 24 x 80
  TELNETDEVICE 3279-5-e NSX32705          ; 27 x 132, primary 24 x 80
  TELNETDEVICE LINEMODE INTERACT          ; linemode terminals
  TELNETDEVICE DYNAMIC          ,D4C32XX3 ; tbd by application (QUERY)
; TELNETDEVICE 3287-1           ,DSC2K    ; printer 2 kbyte bfr LU3
  TELNETDEVICE 3287-1           ,SCS      ; printer             LU1
; Define the LUs to be used for general users.
;
  LUGROUP LU1
      RA39TN01..RA39TN05
  ENDLUGROUP
;
  IPGROUP IP1
      255.255.255.0:9.24.104.0
  ENDIPGROUP
;
  PRTGROUP PRT1
      RA39TP01..RA39TP05
  ENDPRTGROUP
;
  LUMAP LU1 IP1 GENERIC PRT1
;
  LUGROUP LU2
      RA39TN06..RA39TN10
  ENDLUGROUP
;
  IPGROUP IP2
      255.255.255.0:9.24.105.0
  ENDIPGROUP
;
  PRTGROUP PRT2
      RA39TP06..RA39TP10
  ENDPRTGROUP
;
  LUMAP LU2 IP2 GENERIC PRT2
;
  LUGROUP LU3
      RA39TN11..RA39TN15
  ENDLUGROUP
;
  IPGROUP IP3
      255.255.255.0:9.24.106.0
  ENDIPGROUP
;
  PRTGROUP PRT3
      RA39TP11..RA39TP15
  ENDPRTGROUP
;
  LUMAP LU3 IP3 GENERIC PRT3
;
  LUGROUP LU4
      RA39TN16..RA39TN20
```

```
  ENDLUGROUP
;
  IPGROUP IP4
      255.0.0.0:9.0.0.0
  ENDIPGROUP
;
  PRTGROUP PRT4
      RA39TP16..RA39TP20
  ENDPRTGROUP
;
  LUMAP LU4 IP4 GENERIC PRT4
;
  LUGROUP LU5
      RA39TN26..RA39TN40
  ENDLUGROUP
;
  IPGROUP IP5
      255.255.0.0:172.16.0.0
  ENDIPGROUP
;
  PRTGROUP PRT5
      RA39TP26..RA39TP40
  ENDPRTGROUP
;
  LUMAP LU5 IP5 GENERIC PRT5
;
  MSG07            ; Error messages will be issued
  LUSESSIONPEND    ; On termination of a Telnet server connection,
;                  ; the user will revert to the DEFAULTAPPL
; USSTCP TELNUST 9.24.104.201
; USSTCP TELNUSU 9.24.104.202
; USSTCP TELNUSV 9.24.104.47
  USSTCP TELNUST
; DEFAULTAPPL RA03T  ; Set the default application for all Telnet => TSO
; DEFAULTAPPL RAKAA ; Set the default application for all Telnet
;                    ; sessions to allow CLSDST Pass
;                    ; APPLID=RAKAA ===> NVAS20
  LINEMODEAPPL RA03T ; Send all line-mode terminals directly to TSO.
; ALLOWAPPL SAMON QSESSION  ; SAMON appl does CLSDST Pass to next appl
; ALLOWAPPL RA03T * DISCONNECTABLE ; Allow all users access to TSO
;             ; applications.
;             ; TSO is multiple applications all beginning with RA3AT,
;             ; so use the * to get them all.  If a session is closed,
;             ; disconnect the user rather than log off the user.
; RESTRICTAPPL IMS ; Only 3 users can use IMS.
;   USER USER1     ; Allow user1 access.
;     LU TCPIMS01  ; Assign USER1 LU TCPIMS01.
;   USER USER2     ; Allow user2 access from the default LU pool.
;   USER USER3     ; Allow user3 access from 3 Telnet sessions,
;                  ; each with a different reserved LU.
;     LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
; ALLOWAPPL RA03T * DISCONNECTABLE ; Allow all users access to TSO
;             ; applications.
;             ; TSO is multiple applications all beginning with RA3AT,
;             ; so use the * to get them all.  If a session is closed,
;             ; disconnect the user rather than log off the user.
  ALLOWAPPL RA*  ;* DISCONNECTABLE  Allow all users access to TSO
  ALLOWAPPL AD*
```

```
  ALLOWAPPL A2*
  ALLOWAPPL FD*
  ALLOWAPPL X6*
  ALLOWAPPL X7*
; ALLOWAPPL *      ; Allow all applications that have not been
;                  ; previously specified to be accessed.
;
;   Map Telnet sessions from this node to display USSAPC screen.
;   USSTAB USSAPC 130.50.10.1
;
;   Map Telnet sessions from this link to display USSCBA screen.
;   USSTAB USSCBA SNA1
;
ENDVTAM
;TELNETPARMS
;   SECUREPORT 223 KEYRING hfs /etc/telnetssl/keyfile.kyr
;    INACTIVE 7200
;    TIMEMARK 7200
;    SCANINTERVAL 300
;    SMFINIT STD
;    SMFTERM STD
;    WLMCLUSTERNAME TNRALSSL2 ENDWLMCLUSTERNAME
;ENDTELNETPARMS
;TELNETPARMS
;    PORT 323 KEYRING hfs /etc/telnetssl/keyfile.kyr
;    INACTIVE 7200
;    TIMEMARK 7200
;    SCANINTERVAL 300
;    SMFINIT STD
;    SMFTERM STD
;    WLMCLUSTERNAME TNRALSSL3 ENDWLMCLUSTERNAME
;ENDTELNETPARMS
;TELNETPARMS
;    PORT 423 SSLONLY KEYRING hfs /etc/telnetssl/keyfile.kyr
;    INACTIVE 7200
;    TIMEMARK 7200
;    SCANINTERVAL 300
;    SMFINIT STD
;    SMFTERM STD
;    WLMCLUSTERNAME TNRALSSL4 ENDWLMCLUSTERNAME
;ENDTELNETPARMS
```

A.3 TCPIP.DATA

```
;*********************************************************************
;                                                                    *
;   Name of Data Set:     TCPIP.TCPPARMS(TCPD39A)                    *
;                                                                    *
;*********************************************************************
TCPIPJOBNAME TCPIPA
HOSTNAME  MVS39A
DOMAINORIGIN  buddha.ral.ibm.com
NSINTERADDR  9.24.104.125
NSPORTADDR 53
;TRACE RESOLVER
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
```

```
RESOLVERUDPRETRIES 1
DATASETPREFIX TCPIP
MESSAGECASE MIXED
; MESSAGECASE UPPER
; LOADDBCSTABLES SJISKANJI EUCKANJI
```

A.4 OSPF configuration file

```
Area    Area_Number=0.0.0.0
              Stub_Area=NO
              Authentication_type=None;
OSPF_Interface IP_Address=172.16.102.39
              Name=m392216b
              Cost0=5
              Subnet_mask=255.255.255.0
              MTU=32768;
OSPF_Interface IP_Address=172.16.105.39
              Name=m392216c
              Cost0=5
              Subnet_mask=255.255.255.0
              MTU=32768;
OSPF_Interface IP_Address=172.16.233.*
              Cost0=10
              Subnet_mask=255.255.255.0
              MTU=32768;
OSPF_Interface IP_Address=9.24.104.149
              Name=tr1
              Cost0=6
              Subnet_mask=255.255.255.0
              MTU=4082;
OSPF_Interface IP_Address=9.24.105.73
              Name=en1
              Cost0=8
              Subnet_mask=255.255.255.0
              MTU=1492;
OSPF_Interface IP_Address=172.16.240.*
              name=DVIP39A
              Subnet_mask=255.255.255.192
              Cost0=10;
Interface     IP_Address=172.16.232.39
              name=VIPA39A
              Subnet_mask=255.255.255.0
              MTU=32768;
AS_Boundary_routing
              Import_Direct_Routes=YES;
RouterID=172.16.232.39;
ROUTESA_CONFIG ENABLED=YES
             COMMUNITY="MVSsubagent";
```

A.5 FTP server FTP.DATA

```
;*********************************************************************
;                                                                    *
;   Name of File:      TCP.TCPPARMS(FDATA03)                         *
;                                                                    *
;   FTP.DATA for the CS/390 V2R5 FTP server on Ports 20/21           *
```

```
;                                                                    *
;*********************************************************************
;
;ASATRANS      FALSE           ; do NOT translate control characters
                               ; in ASA text
AUTOMOUNT     TRUE            ; automatic mount of unmounted volume
AUTORECALL    TRUE            ; automatic recall of migrated data sets
;AUTOTAPEMOUNT FALSE           ; do NOT automatically mount tape volumes
BLOCKSIZE     6233            ; new data set allocation blocksize
BUFNO         5               ; number of access method buffers
CHKPTINT      0               ; checkpoint interval
CONDDISP      CATLG           ; data sets catalogued if transfer fails
; CTRLCONN      7BIT            ; ascii code set for control connection
CTRLCONN      IBM-850         ; ascii code set for control connection
;DATACLASS     SMSDATA         ; sms data class name
;DB2           D31             ; db2 subsystem name
;DB2PLAN       PLANNAME        ; db2 plan name for OE-FTP
;DCBDSN        MODEL.DCB       ; new data set allocation model dcb name
;DEST          USER14@MVSL     ; files destination for store
DIRECTORY     27              ; new data set allocation directory blocks
DIRECTORYMODE FALSE           ; directorymode vs. data set mode
FILETYPE      SEQ             ; file transfer mode
;INACTIVE      300             ; inactive time out
INACTIVE      3600            ; inactive time out
JESLRECL      80              ; lrecl of jes jobs
JESPUTGETTO   600             ; timeout for remote job submission put/ge
JESRECFM      F               ; recfm of jes jobs
LRECL         256             ; new data set allocation lrecl
;MGMTCLASS     SMSMGMT         ; sms mgmtclass name
;MIGRATEVOL    MIGRAT          ; migration volume volser
PRIMARY       20              ; new data set allocation primary space
;QUOTESOVERRIDE FALSE         ; single quote(s) are treated as part of
                               ; hfs filename, i.e. single quotes do
                               ; NOT indicate working directory override
RDW           false           ; if RDWs are treated as a part of record
RECFM         VB              ; new data set allocation record format
;RETPD         30              ; new data set retention period: 30 days
SBDATACONN   (IBM-1047,IBM-850) ; ebcdic/ascii code sets for data conn.
;SBDATACONN    TCP.FTPKANA.TCPXLBIN
SECONDARY     10              ; new data set allocation secondary space
;SMF           STD             ; SMF subtype use standard subtypes
;SMFAPPL       70              ; SMF subtype for APPE(APPEND) subcommand
;SMFDEL        71              ; SMF subtype for DELE(DELETE) subcommand
;SMFEXIT                       ; load SMF user exit FTPSMFEX
;SMFJES                        ; SMF recording when filetype=jes
;SMFLOGN       72              ; SMF subtype for login failure
;SMFREN        73              ; SMF subtype for RNFT/RNTO(RENAME)
;SMFRETR       74              ; SMF subtype for RETR(RETRIEVE)
;SMFSQL                        ; SMF recording when filetype=sql
;SMFSTOR       75              ; SMF subtype for STOR(STORE) and STOU
SPACETYPE     CYL             ; new data set allocation space type
SPREAD        FALSE           ; sql output format
SQLCOL        NAMES           ; sql output uses column names as headings
;STARTDIR      HFS             ; use MVS directory at connect time
STARTDIR      MVS             ; use MVS directory at connect time
;STORCLASS     SMSSTOR         ; sms storclass name
;TRACE                         ; trace active
;TRAILINGBLANKS TRUE           ; include trailing blanks when fixed
```

```
                                ; format data sets are retrieved
UCSHOSTCS      IBM-939          ; the EDCDIC code page from/to UCS-2
UCSSUB         FALSE            ; whether substitution is permitted.
UCSTRUNC       FALSE            ; whether truncation is permitted.
;UMASK          027             ; octal UMASK to restrict setting
                                ; of permission bits when creating
                                ; new hfs files
;UNITNAME       3380            ; new data set allocation unit
;VOLUME         WRKLB2          ; new data set allocation volume serial
WLMCLUSTERNAME FTPRAL          ; group name registered in DNS/WLM sysplex
WRAPRECORD     FALSE           ; data is NOT wrapped to next record
```

Appendix B. $$CNTL$$ member

This appendix shows the contents of the $$CNTL$$ member as it ships within SYS1.SAMPLIB.

```
//USER01I  JOB MSGLEVEL=(0,0),CLASS=5,NOTIFY=&SYSUID,MSGCLASS=H
/*JOBPARM  S=ANY,LINES=99
//*--------------------------------------------------------------------
//*
//* RACFICE -  Create reports from RACF database unload utility
//*            (IRRDBU00) and RACF SMF unload utility (IRRADU00)
//* RACF Database Reports
//*
//*    Name   Description
//*    ----   ----------------------------------------------------------
//*    ALDS   Discrete data set profiles which have IDs on the
//*           standard access list with ALTER authority
//*    ASOC   Users who have explicit associations defined
//*    BGGR   Discrete general respource profiles with generic
//*           characters in their name
//*    CCON   Count of user connections, flagging those with more
//*           than "x" connections
//*    CGEN   Count of general resource profiles
//*    CPRO   Count of profiles
//*    CONN   User IDs with group privileges above use
//*    IDSC   Data set conditional access lists with ID(*) of other
//*           than NONE
//*    IDSS   Data set standard access lists with ID(*) of other
//*           than NONE
//*    IGRC   General resource conditional access lists with ID(*)
//*           of other than NONE
//*    IGRS   General resource standard access lists with ID(*)
//*           of other than NONE
//*    OMVS   User IDs which have UNIX System Services (OMVS) segments
//*    SUPU   UNIX System Services super users (UID of Zero)
//*    UGLB   User IDs With extraordinary system-level authorities
//*    UGRP   User IDs with extraordinary RACF group authorities
//*    UIDS   UNIX System Services UIDs used more than once
//*    URVK   User IDs which are currently revoked
//*    UADS   Data set profiles with UACCs of other than NONE
//*    UAGR   General resource profiles with UACCs of other than NONE
//*    WNDS   Data set profiles in WARNING mode
//*    WNGR   General resource profiles in WARNING mode
//*
//*
//* RACF Audit Reports
//*
//*    Name   Description
//*    ----  ----------------------------------------------------------
//*    ACD$   Users who are using automatic command direction
//*    CADU   Count of IRRADU00 events
//*    CCMD   Count of commands issued (by user)
//*    ECD$   Users who are directing commands explicitly
//*    LOGB   Users who log on with LOGON BY
//*    LOGF   All users with excessive incorrect passwords
//*    OPER   Accesses allowed because the user has OPERATIONS
//*           authority
```

```
//*    PWD$   Users who are using password synchronization
//*    RACL   RACLINK audit records
//*    RINC   RACF class initialization records
//*    SELU   All audit records for a specific user
//*    SPEC   Events that succeeded because the user has SPECIAL
//*           authority
//*    TRMF   Excessive incorrect passwords from terminals
//*    VIOL   Access violations
//*    WARN   Accesses allowed due to WARNING mode profiles
//*--------------------------------------------------------------------
//         JCLLIB ORDER=USER01.RACFICE.CNTL
//         SET  ADUDATA=USER01.RACFICE.IRRADU00           IRRADU00 data
//         SET  DBUDATA=USER01.RACFICE.IRRDBU00           IRRDBU00 data
//         SET  ICECNTL=USER01.RACFICE.CNTL               ICETOOL  data
//*--------------------------------------------------------------------
//*------- IRRDBU00-Based Reports ---------------------------------------
//*--------------------------------------------------------------------
//ALDS     EXEC RACFICE,REPORT=ALDS
//ASOC     EXEC RACFICE,REPORT=ASOC
//BGGR     EXEC RACFICE,REPORT=BGGR
//CCON     EXEC RACFICE,REPORT=CCON
//CGEN     EXEC RACFICE,REPORT=CGEN
//CPRO     EXEC RACFICE,REPORT=CPRO
//CONN     EXEC RACFICE,REPORT=CONN
//IDSC     EXEC RACFICE,REPORT=IDSC
//IDSS     EXEC RACFICE,REPORT=IDSS
//IGRC     EXEC RACFICE,REPORT=IGRC
//IGRS     EXEC RACFICE,REPORT=IGRS
//OMVS     EXEC RACFICE,REPORT=OMVS
//SUPU     EXEC RACFICE,REPORT=SUPU
//UADS     EXEC RACFICE,REPORT=UADS
//UAGR     EXEC RACFICE,REPORT=UAGR
//UGLB     EXEC RACFICE,REPORT=UGLB
//UGRP     EXEC RACFICE,REPORT=UGRP
//UIDS     EXEC RACFICE,REPORT=UIDS
//URVK     EXEC RACFICE,REPORT=URVK
//WNDS     EXEC RACFICE,REPORT=WNDS
//WNGR     EXEC RACFICE,REPORT=WNGR
//*--------------------------------------------------------------------
//*------- IRRADU00-Based Reports ---------------------------------------
//*--------------------------------------------------------------------
//ACD$     EXEC RACFICE,REPORT=ACD$
//CADU     EXEC RACFICE,REPORT=CADU
//CCMD     EXEC RACFICE,REPORT=CCMD
//ECD$     EXEC RACFICE,REPORT=ECD$
//LOGB     EXEC RACFICE,REPORT=LOGB
//LOGF     EXEC RACFICE,REPORT=LOGF
//OPER     EXEC RACFICE,REPORT=OPER
//PWD$     EXEC RACFICE,REPORT=PWD$
//RACL     EXEC RACFICE,REPORT=RACL
//RINC     EXEC RACFICE,REPORT=RINC
//SELU     EXEC RACFICE,REPORT=SELU
//SPEC     EXEC RACFICE,REPORT=SPEC
//TRMF     EXEC RACFICE,REPORT=TRMF
//VIOL     EXEC RACFICE,REPORT=VIOL
//WARN     EXEC RACFICE,REPORT=WARN
```

Appendix C. Sample reports

This appendix shows the sample reports as they appear in the order that $$CNTL$$ runs them if unchanged.

```
- 1 -        ALDS: IDs with ALTER Auth to Discrete DS Profs        99/06/23        09:08:47 am

Profile Name                                        Volume   ID         Access Count
------------------------------------------------    ------   --------   ------------
CATALOG.ICFICFM.VICFCAT                             ICFCAT   HAIMO      00000
SYS1.UADS                                           MXARS1   IBMUSER    00000
```

Figure 453. Sample ALDS report

```
- 1 -       ASOC: Users Who Have Explicit Associations Defined       99/06/17       04:52:18 pm

User ID   Tgt Node   Tgt User   Peer   Remote Pend   Local Pend   PWSYNC   Accepted Date   Accepted Time   Creator
--------  --------   --------   ----   -----------   ----------   ------   -------------   -------------   --------
MEUDT     SC47TS     SARDELL    YES    YES           NO           NO                                       MEUDT
SLACKER   ICF        TEEDEE     YES    NO            NO           YES      1999-06-17      20:07:38        TEEDEE
TEEDEE    ICF        SLACKER    YES    NO            NO           YES      1999-06-17      20:07:38        TEEDEE
```

Figure 454. Sample ASOC report

```
- 1 -        BGGR: Discrete GR Profiles with Generic Names        99/06/17        01:26:35 pm

Profile                                     Generic   Class      Created      By
----------------------------------------    -------   --------   ----------   --------
*.SUBMIT                                    NO        SURROGAT   1999-06-17   TSO
```

Figure 455. Sample BGGR report

```
- 1 -   CCON: Users Who Have More Than 20 Group Connections        99/06/17        10:42:56 am

User ID    Number of Group Connections
--------   ---------------------------
SLACKER                            22
```

Figure 456. Sample CCON report

As shipped, CCON has a value of HIGHER(100) - modified for testing purposes to HIGHER(20).

```
- 1 -       CGEN: Number of General Resource Profiles       99/06/17       10:42:57 am

Class                Count
--------   ---------------
ACCTNUM                  2
APPCLU                   3
APPCPORT                 1
APPCSERV                 2
APPCTP                   2
APPL                     2
CSFKEYS                  2
CSFSERV                  6
DASDVOL                 23
DIGTCERT                 7
DSNR                     1
FACILITY                51
FIELD                    2
GCICSTRN                 4
OPERCMDS                 4
PROGRAM                  5
```

Figure 457. Sample CGEN report

```
- 1 -     CPRO: Number of Profiles in the RACF Data Base       99/06/17       12:13:05 pm

Type                                Count
--------------------------   ---------------
Dataset Profiles                      305
General Resource Profiles             209
Group Profiles                        217
User Profiles                          86
```

Figure 458. Sample CPRO report

```
- 1 -       CONN: User IDs With Group Privileges Above Use       99/06/17       12:13:06 pm

User ID    Group ID   Authority
--------   --------   ---------
WEBSRV     IMWEB      JOIN
WEBADM     IMWEB      JOIN
IBMUSER    SYSCTLG    JOIN
IBMUSER    SYS1       JOIN
HAIMO      SYS1       JOIN
KARRAS     SYS1       JOIN
DAV        SYS1       JOIN
SRCDAWS    SYS1       JOIN
RCONWAY    SYS1       JOIN
ALFREDC    SYS1       JOIN
BOCHE      SYS1       JOIN
CARL       SYS1       JOIN
DB         SYS1       JOIN
ROGERS     SYS1       JOIN
DRAGON     SYS1       JOIN
```

Figure 459. Sample CONN report

```
- 1 -        IDSC: DS Conditional ACLs With ID(*) Other Than None        99/06/17        01:26:39 pm

Profile Name                                 Volume   Access     Access Type   Access Element
-------------------------------------------  ------   --------   -----------   --------------
SLACKER.S.MOST.VALUABLE.DSN.**                        UPDATE     PROGRAM       TELNET
```

Figure 460. Sample IDSC report

```
- 1 -    IDSS: DS Standard ACLs With ID(*) Other Than None        99/06/17        01:26:39 pm

Profile Name                                 Volume   Access
-------------------------------------------  ------   --------
JJONES.**                                             READ
```

Figure 461. Sample IDSS report

```
- 1 -   IGRC: GR Conditional ACLs with ID(*) Other Than None        99/06/17        01:49:46 pm

Class      Profile Name                                                  Access     Access Type   Access Element
--------   ------------------------------------------------------------  --------   -----------   --------------
TCICSTRN   HELP                                                          READ       TERMINAL      HACKRTRM
```

Figure 462. Sample IGRC report

```
- 1 -   IGRS: GR Standard ACLs with ID(*) Other Than None        99/06/17        01:49:47 pm

Class      Profile Name                                                                  Access
--------   ----------------------------------------------------------------------------  -------
FACILITY   IRR.DIGTCERT.ADD                                                              READ
FACILITY   IRR.DIGTCERT.DELETE                                                           READ
GCICSTRN   DKMS.CSGM                                                                     UPDATE
```

Figure 463. Sample IGRS report

```
- 1 -    OMVS: User IDs With an OMVS Segment        99/06/16        02:55:27 pm

User ID    OMVS UID     OMVS Home Path                                  Default Program
--------   ----------   ---------------------------------------------   ---------------
ALFREDC    0000000000   /u/alfredc                                      /bin/sh
BOCHE      0000000271   /u/boche                                        /bin/sh
BPX        0000000101   /
BPXROOT    0000000000   /                                               /bin/sh
CARL       0000000270   /u/carl                                         /bin/sh
CARLK      0000000275   /u/carlk                                        /bin/sh
DAND       0000000000   /
DAV        0000000328   /u/dav                                          /bin/sh
DB         0000000000   /u/db                                           /bin/sh
FSARDEL    0000002222   /u/fsardel                                      /bin/sh
FWKERN     0000000000   /u/fwkern
GRAAFF     0000000000   /u/graaff                                       /bin/sh
HAIMO      0000000000   /                                               /bin/sh
HILDING    0000000000   /u/hilding                                      /bin/sh
HNOMO      0000055556   /u/hnomo                                        /bin/sh
INTERNAL   0000000537   /usr/lpp/internet                               /bin/sh
IOPER01    0000000157   /u/ioper01                                      /bin/sh
```

Figure 464. Sample OMVS Report

```
- 1 -   SUPU: OpenEdition Super Users (UID of Zero)        99/06/17        01:49:48 pm

User ID    Path Name                             Home Name
--------   -----------------------------------   -----------------------------------
ALFREDC    /u/alfredc                            /bin/sh
BPXROOT    /                                     /bin/sh
DAND       /
DB         /u/db                                 /bin/sh
FWKERN     /u/fwkern
GRAAFF     /u/graaff                             /bin/sh
HAIMO      /                                     /bin/sh
HILDING    /u/hilding                            /bin/sh
KAPPELE    /u/kappele                            /bin/sh
LDAPSRV    /                                     /bin/sh
LOGD       /
MARTI      /u/marti                              /bin/sh
OMVSKERN   /                                     /bin/echo
RCONWAY    /u/rconway                            /bin/sh
ROGERS     /                                     /BIN/SH
STC        /
```

Figure 465. Sample SUPU report

```
- 1 -       UADS: DS Profiles with a UACC Other Than None        99/06/23        09:08:58 a

Data Set Name Profile Name                    Volume   Generic   Owner      UACC
-------------------------------------------   ------   -------   --------   --------
@PL.*.**                                               YES       @PL        ALTER
ACFNCP.*.**                                            YES       ACFNCP     ALTER
ALFREDC.**                                             YES       ALFREDC    ALTER
AMS.*.**                                               YES       AMS        ALTER
ANF.**                                                 YES       ANF        READ
ANO.*.**                                               YES       ANO        READ
APL2.*.**                                              YES       APL2       ALTER
AP2V1R03.*.**                                          YES       AP2V1R03   ALTER
BASIC.*.**                                             YES       BASIC      ALTER
BFS.**                                                 YES       BFS        READ
CATALOG.*.**                                           YES       CATALOG    ALTER
CATALOG.ESAICFM.VESACAT                       ESACAT   NO        CATALOG    READ
CATALOG.ICFICFM.VICFCAT                       ICFCAT   NO        CATALOG    READ
CBC.**                                                 YES       CBC        READ
CBIPO.*.**                                             YES       CBIPO      READ
CBPDO.*.**                                             YES       CBPDO      READ
```

Figure 466. Sample UADS report

```
- 1 -       UAGR: GR Profiles with a UACC Other Than None        99/06/23        09:08:59 a

Class      General Resource Profile Name                 Generic   Owner      UACC
--------   -------------------------------------------   -------   --------   --------
ACCTNUM    ACCNT#                                        NO    0   GRAAFF     READ
ACCTNUM    ACCT#                                         NO    0   GRAAFF     READ
APPCPORT   *                                             YES   0   SYS1       ALTER
APPCTP     *                                             YES   0   SYS1       READ
CSFKEYS    **                                            YES   1   DKMS       READ
CSFKEYS    XDM9005*                                      YES   1   FINNTC     READ
DASDVOL    MXXE83                                        NO    0   SYS1       READ
DIGTCERT   03256E20E1097BCAC58A2B8E155ADA13.OU=VeriSign  NO    0   BOCHE      ALTER
DIGTCERT   037C.CN=DEMO¢ZERO¢VALUE¢CA.OU=DEMONSTRATION¢  NO    0   SWEENY     ALTER
DIGTCERT   113C17D8D3A2371D0A76B63A6C5424E8.OU=BT¢Trust  NO    0   GRAAFF     ALTER
DIGTCERT   45F2747676D3EA719EF4A920F9200724.CN=VeriSign  NO    0   JJONES     ALTER
DIGTCERT   576AAF086990B4A7E8DA83718C28FCBB.OU=VeriSign  NO    0   GRAAF2     ALTER
DIGTCERT   59D344A1A457827589C9741A5A7C0958.CN=VeriSign  NO    0   GRAAFF     ALTER
DIGTCERT   7FF17E7A3DBE7365A8D419820BA0E0E5.CN=VeriSign  NO    0   TSTBUDS    ALTER
FACILITY   QCBTRACE.AUTHORIZATION                        NO    0   HAIMO      READ
FACILITY   STGADMIN.IDC.BINDDATA                         NO    0   HAIMO      READ
```

Figure 467. Sample UAGR report

```
- 1 -    User IDs With Extraordinary Global Authorities        99/06/17        01:49:49 pm

User ID    User Name              Special   Operations   Auditor
--------   --------------------   -------   ----------   -------
ALFREDC    ALFRED CHRISTENSEN     YES       YES          NO
BE24286    YVES DEPOORTER         YES       NO           NO
BOCHE      ULRICH BOCHE           YES       NO           YES
CARL       CARL KLITSCHER         YES       NO           NO
DAV        DENNIS VIRGINIA        YES       YES          NO
DB         DAVE BENNIN            YES       YES          NO
DRAGON     DRAGON                 YES       YES          NO
FINNTC     FINN THUNBO            YES       YES          YES
FSARDEL    FSARDEL                YES       YES          NO
GRAAFF     PAUL DE GRAAFF         YES       NO           YES
GRAAF2     GRAAF2                 YES       NO           NO
HAIMO      BOB HAIMOWITZ          YES       YES          NO
HILDING    HILDING LANDEN         YES       YES          NO
IBMUSER                           YES       YES          NO
JJONES     JACK JONES             YES       NO           YES
KAPPELE    KAPPELE                NO        NO           YES
```

Figure 468. Sample UGLB report

```
- 1 -       UGRP: User IDs With Extraordinary RACF Group Authorities       99/06/17       01:49:50 pm

User ID    Group ID   Group Special   Group Operations   Group Auditor
--------   --------   -------------   ----------------   -------------
SLACKER    BLUEGRP    NO              YES                NO
SLACKER    BROGRP     NO              YES                NO
SLACKER    DADSGRP    NO              YES                NO
SLACKER    GREENGRP   NO              YES                NO
SLACKER    HAPPYGRP   NO              YES                NO
SLACKER    JOHNL      NO              YES                NO
SLACKER    JULIAR     NO              YES                NO
SLACKER    KIDSGRP    NO              YES                NO
SLACKER    MAUVEGRP   NO              YES                NO
SLACKER    MIKAH      NO              YES                NO
SLACKER    MOMSGRP    NO              YES                NO
SLACKER    REDGRP     NO              YES                NO
SLACKER    RICHIII    NO              YES                NO
SLACKER    ROBERTR    NO              YES                NO
SLACKER    ROCKGRP    NO              YES                NO
SLACKER    SADGRP     NO              YES                NO
```

Figure 469. Sample UGRP report

```
- 1 -        UIDS: OS/390 UNIX UIDs Used More Than Once        99/06/17        02:21:41 pm

OpenEdition UID   Number of Times Used
---------------   --------------------
0000000000                         22
0000099999                          2
```

Figure 470. Sample UIDS report

```
- 1 -       URVK: User IDs Which are Currently Revoked       99/06/23       09:09:02 am

User ID    User Name
--------   --------------------
irrcerta   CERTAUTH Anchor
irrsitec   SITE Anchor
IBMUSER
TEDEE      TEDS TEST ID
```

Figure 471. Sample URVK report

```
- 1 -       WNDS: DS Profiles with in WARNING Mode       99/06/23       09:09:03 am

Data Set Name                                Volume   Owner      UACC
-------------------------------------------  ------   --------   --------
PEKKAH.**                                             PEKKAH     NONE
SYS1.APPCSI                                  ICFRS1   SYS1       NONE
TSTBUDS.**                                            TSTBUDS    NONE
```

Figure 472. Sample WNDS report

```
- 1 -       WNGR: GR Profiles with in WARNING Mode       99/06/23       09:09:04 am

General Resource Profile Name                Class      Owner      UACC
-------------------------------------------  --------   --------   --------
*                                            APPCPORT   SYS1       ALTER
*                                            APPCSERV   SYS1       NONE
*                                            APPCTP     SYS1       READ
DB3A.*                                       DSNR       FINNTC     NONE
APPCMVS.DBTOKEN                              FACILITY   MEUDT      NONE
DFH*                                         FACILITY   FINNTC     NONE
**                                           OPERCMDS   BOCHE      NONE
ATBSDFMU                                     PROGRAM    MEUDT      NONE
*                                            VTAMAPPL   SYS1       NONE
```

Figure 473. Sample WNGR report

```
- 1 -       Users Who Are Using Automatic Command Direction       99/06/17       02:46:34 pm

User       Date         Time       Result     Profile              Command (From LOGSTR)
--------   ----------   --------   --------   ------------------   ------------------------------------------------
```

Figure 474. Sample ACD$ report

```
- 1 -       Number of Events in the IRRADU00 Data Set        99/06/17        02:46:34 pm

Type                 Count
--------   ---------------
ACCESS                  77
ADDGROUP                25
ADDSD                    5
ADDUSER                  8
ALTDSD                   1
ALTUSER                 10
CHDIR                   90
CHKPRIV                  2
CONNECT                 21
DEFINE                  66
DELGROUP                 3
DELRES                  43
DELUSER                  2
FACCESS                  4
INITOEDP                 3
JOBINIT                151
```

Figure 475. Sample CADU report

```
- 1 -       CCMD: Count of Commands Issued by User ID        99/06/23        09:09:22 am

User ID    Command              Count
--------   --------   ---------------
GRAAFF     RALTER                   1
GRAAFF     SETROPTS                 1
HAIMO      ADDGROUP                 5
HAIMO      ADDSD                    5
HAIMO      PERMIT                   6
HAIMO      RVARY                    5
JJONES     ALTUSER                  2
JJONES     PERMIT                   3
JJONES     RACDCERT                 2
JJONES     SETROPTS                 5
PEKKAH     ADDGROUP                 5
PEKKAH     ADDUSER                  8
PEKKAH     ALTGROUP                 2
PEKKAH     ALTUSER                 28
PEKKAH     CONNECT                  4
PEKKAH     DELGROUP                 3
```

Figure 476. Sample CCMD report

```
- 1 -       ECD$: Users Who Are Directing Commands Explicitly        99/06/17        02:46:36 pm

User      Date         Time      Result    Profile              Command (From LOGSTR)
--------  ----------   --------  --------  -------------------  ------------------------------------------------------
```

Figure 477. Sample ECD$ report

```
- 1 -    LOGB: Users Who Log on with LOGON BY        99/06/17        02:46:36 pm

Date         Time       System   Terminal   User       LOGSTR     By User
----------   --------   ------   --------   --------   --------   --------
```

Figure 478. Sample LOGB report

```
- 1 -     LOGF: User IDs With Excessive Incorrect Passwords        99/06/17        02:46:37 pm

User ID    Number of Incorrect Passwords
--------   -----------------------------
SLACKER                               5
```

Figure 479. Sample LOGF report

```
- 1 -    OPER: Accesses Allowed Because the User has OPERATIONS       99/06/17       02:46:38 pm

Time       Date         User ID    Resource Name                        Volume   Profile
--------   ----------   --------   ----------------------------------   ------   ------------------
14:23:53   1999-06-16   SLACKER    CATALOG.ICFICFM.VICFCAT              ICFCAT
14:23:55   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.D0615               PDGSY1   TSTBUDS.**
14:23:55   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.D0615               PDGSY1   TSTBUDS.**
14:23:58   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.NONUM               PDGSY1   TSTBUDS.**
14:23:58   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.NONUM               PDGSY1   TSTBUDS.**
14:24:00   1999-06-16   SLACKER    TSTBUDS.ISPF.ISPPROF                 PDGSY1   TSTBUDS.**
14:24:00   1999-06-16   SLACKER    TSTBUDS.ISPF.ISPPROF                 PDGSY1   TSTBUDS.**
14:24:02   1999-06-16   SLACKER    TSTBUDS.JCL.CNTL                     PDGSY1   TSTBUDS.**
14:24:02   1999-06-16   SLACKER    TSTBUDS.JCL.CNTL                     PDGSY1   TSTBUDS.**
14:24:07   1999-06-16   SLACKER    TSTBUDS.RECJCL                       PDGSY1   TSTBUDS.**
14:24:10   1999-06-16   SLACKER    TSTBUDS.SMF.FLATFILE                 PDGSY1   TSTBUDS.**
14:24:14   1999-06-16   SLACKER    TSTBUDS.SMFDBA.FLATFILE              PDGSY1   TSTBUDS.**
14:24:25   1999-06-16   SLACKER    CATALOG.ICFICFM.VICFCAT              ICFCAT
14:24:36   1999-06-16   SLACKER    GRAAFF.PENTEST.CLIST                 PDGSY1   GRAAFF.PENTEST.**
14:24:36   1999-06-16   SLACKER    GRAAFF.PENTEST.CLIST                 PDGSY1   GRAAFF.PENTEST.**
14:24:39   1999-06-16   SLACKER    GRAAFF.PENTEST.CNTL                  PDGSY1   GRAAFF.PENTEST.**
```

Figure 480. Sample OPER report

```
- 1 -        PWD$: Users Who Are Using Password Synchronization        99/06/23 09:09:42 am

User       Date         Time       Result     Profile                LOGSTR Data
--------   ----------   --------   --------   --------------------   ------------------------
SLACKER    1999-06-18   14:01:30   SUCCESS
SLACKER    1999-06-18   14:01:52   SUCCESS
TEEDEE     1999-06-18   14:02:25   SUCCESS
SLACKER    1999-06-18   14:02:51   SUCCESS
TEEDEE     1999-06-18   14:03:25   SUCCESS
```

Figure 481. Sample PWD$ report.

The USERs must have UAUDIT turned on in order for anything to appear in this report.

```
- 1 -    RACL: RACLINK Audit Records        99/06/17        04:52:36 pm

User       Date         Time       Result     Phase              Password   Assoc     RACLINK Operands
--------   ----------   --------   --------   ----------------   --------   -------   --------------------------------
TEEDEE     1999-06-17   16:07:37   SUCCESS    LOCAL ISSUANCE     SUPPLIED   PENDING   ID(TEEDEE) DEFINE(ICF.SLACKER)
SLACKER    1999-06-17   16:07:38   SUCCESS    TARGET PROCESSING  VALID      ESTAB     ID(TEEDEE) DEFINE(ICF.SLACKER)
TEEDEE     1999-06-17   16:07:38   SUCCESS    TARGET RESPONSE    VALID      ESTAB     ID(TEEDEE) DEFINE(ICF.SLACKER)
```

Figure 482. Sample RACL report

```
- 1 -       RINC: Class Statistics at IPL        99/06/23        09:09:50 am

Time/Date/System:  11:09:38 1999-06-22 3090

Date         Time       Sys    Class      Stats   Audit   Active   Generic   GENCMD   Global   RACLIST   GENLIST   LOGOPTIONS
----------   --------   ----   --------   -----   -----   ------   -------   ------   ------   -------   -------   ----------
1999-06-22   11:09:38   3090   ACCTNUM    NO      YES     YES      YES       YES      NO       YES       NO        DEFAULT
1999-06-22   11:09:38   3090   ACCTNUM    NO      YES     YES      YES       YES      NO       YES       NO        DEFAULT
1999-06-22   11:09:38   3090   ACCTNUM    NO      YES     YES      YES       YES      NO       YES       NO        DEFAULT
1999-06-22   11:09:38   3090   ACICSPCT   NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   ACICSPCT   NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   ACICSPCT   NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   AIMS       NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   AIMS       NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   AIMS       NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   ALCSAUTH   NO      NO      NO       NO        NO       NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   ALCSAUTH   NO      NO      NO       NO        NO       NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   ALCSAUTH   NO      NO      NO       NO        NO       NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   APPCLU     NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
1999-06-22   11:09:38   3090   APPCLU     NO      YES     YES      YES       YES      NO       NO        NO        DEFAULT
```

Figure 483. Sample RINC report

```
- 1 -        Events Associated with SLACKER, TEEDEE and PEKKAH        99/06/18        10:32:34 a

User ID    Event      Qualifier   Time       Date         System   Terminal   Jobname
--------   --------   ---------   --------   ----------   ------   --------   --------
PEKKAH     JOBINIT    SUCCESS     11:02:17   1999-06-16   9672     SCGPVM04   PEKKAH
PEKKAH     DEFINE     SUCCESS     11:02:45   1999-06-16   9672     SCGPVM04   PEKKAH
PEKKAH     DELRES     SUCCESS     11:05:59   1999-06-16   9672     SCGPVM04   PEKKAH
PEKKAH     JOBINIT    TERM        11:06:01   1999-06-16   9672     SCGPVM04   PEKKAH
TEEDEE     JOBINIT    SUCCESS     14:19:56   1999-06-16   9672     SCGPVM08   TEEDEE
TEEDEE     DEFINE     SUCCESS     14:19:57   1999-06-16   9672     SCGPVM08   TEEDEE
SLACKER    JOBINIT    REVKAUTO    14:20:33   1999-06-16   9672     SCGPVM10
SLACKER    JOBINIT    SUCCESS     14:22:12   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    DEFINE     SUCCESS     14:22:13   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    ACCESS     INSAUTH     14:22:43   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    ACCESS     INSAUTH     14:22:49   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    ACCESS     INSAUTH     14:22:53   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    ACCESS     INSAUTH     14:22:56   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    DEFINE     SUCCESS     14:22:59   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    ACCESS     INSAUTH     14:23:02   1999-06-16   9672     SCGPVM11   SLACKER
SLACKER    DELRES     SUCCESS     14:23:38   1999-06-16   9672     SCGPVM11   SLACKER
```

Figure 484. Sample SELU report

```
- 1 -        SPEC: Events Allowed Because of SPECIAL        99/06/23        09:09:58 am

Date         Time       User ID    Group ID   Event      Result     Terminal   Job Name
----------   --------   --------   --------   --------   --------   --------   --------
1999-06-21   15:30:43   GRAAFF     SYS1       RALTER     SUCCESS    SCGPVM03   GRAAFF
1999-06-21   15:31:00   GRAAFF     SYS1       SETROPTS   SUCCESS    SCGPVM03   GRAAFF
1999-06-22   11:43:06   HAIMO      SYS1       ADDGROUP   SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:06   HAIMO      SYS1       ADDGROUP   SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:06   HAIMO      SYS1       ADDGROUP   SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:14   HAIMO      SYS1       ADDSD      SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:14   HAIMO      SYS1       ADDSD      SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:14   HAIMO      SYS1       ADDSD      SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:26   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:26   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:26   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:36   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:36   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   11:43:36   HAIMO      SYS1       PERMIT     SUCCESS    SC38T00C   HAIMO
1999-06-22   15:38:37   HAIMO      SYS1       ADDGROUP   SUCCESS    SC38T00C   HAIMO
1999-06-22   15:38:37   HAIMO      SYS1       ADDGROUP   SUCCESS    SC38T00C   HAIMO
```

Figure 485. Sample SPEC report

```
- 1 -        Terminals with Excessive Incorrect Passwords        99/06/18        04:35:48 pm

Terminal ID   Number of Incorrect Passwords
-----------   -----------------------------
HACKRTRM                                10
```

Figure 486. Sample TRMF report

```
- 1 -        VIOL: Access Violations        99/06/23        03:08:10 pm

Date         Time       Result     User ID    Resource Name                  Class      Volume   Profile
----------   --------   --------   --------   ----------------------------   --------   ------   -----------------
1999-06-15   13:47:56   INSAUTH    TSTBUDS    GRAAFF.IRRDBU00.CNTL           DATASET    ICFUS4   GRAAFF.**
1999-06-15   13:48:17   INSAUTH    TSTBUDS    GRAAFF.RACFTOOL.CNTL           DATASET    ICFUS4   GRAAFF.**
1999-06-15   14:18:44   INSAUTH    TSTBUDS    GRAAFF.JCL.CNTL                DATASET    ICFUS4   GRAAFF.**
1999-06-16   13:08:22   INSAUTH    TEDEE      CATALOG.ICFICFM.VICFCAT        DATASET    ICFCAT
1999-06-16   13:15:02   INSAUTH    TSTBUDS    GRAAFF.PENTEST.CNTL            DATASET    PDGSY1   GRAAFF.PENTEST.**
1999-06-16   13:21:35   INSAUTH    TSTBUDS    RACFADM.OS390FW.REXX           DATASET    PDGSY1   RACFADM.**
1999-06-16   13:30:26   INSAUTH    TEDEE      CATALOG.ICFICFM.VICFCAT        DATASET    ICFCAT
1999-06-16   13:50:47   INSAUTH    TSTBUDS    CATALOG.ICFICFM.VICFCAT        DATASET    ICFCAT
1999-06-16   14:22:43   INSAUTH    SLACKER    TSTBUDS.ICE.TOOL.D0615         DATASET    PDGSY1   TSTBUDS.**
1999-06-16   14:22:49   INSAUTH    SLACKER    TSTBUDS.ICE.TOOL.NONUM         DATASET    PDGSY1   TSTBUDS.**
1999-06-16   14:22:53   INSAUTH    SLACKER    TSTBUDS.ISPF.ISPPROF           DATASET    PDGSY1   TSTBUDS.**
1999-06-16   14:22:56   INSAUTH    SLACKER    TSTBUDS.JCL.CNTL               DATASET    PDGSY1   TSTBUDS.**
1999-06-16   14:23:02   INSAUTH    SLACKER    TSTBUDS.RACFDB1.FLATFILE       DATASET    PDGSY1   TSTBUDS.**
1999-06-16   14:27:33   INSAUTH    SLACKER    GRAAFF.PENTEST.CLIST           DATASET    PDGSY1   GRAAFF.PENTEST.**
1999-06-16   14:27:39   INSAUTH    SLACKER    GRAAFF.PENTEST.CNTL            DATASET    PDGSY1   GRAAFF.PENTEST.**
1999-06-16   14:27:41   INSAUTH    SLACKER    GRAAFF.PENTEST.EXEC            DATASET    PDGSY1   GRAAFF.PENTEST.**
```

Figure 487. Sample VIOL report

```
- 1 -       Accesses Allowed Due to WARNING Mode Profiles       99/06/18       04:35:50 pm

Time       Date         User ID    Resource Name                          Volume   Profile
--------   ----------   --------   ------------------------------------   ------   -----------
17:22:51   1999-06-17   PEKKAH     TSTBUDS.RACFDB1.FLATFILE               PDGSY1   TSTBUDS.**
17:29:39   1999-06-17   PEKKAH     TSTBUDS.RACFDB1.FLATFILE               PDGSY1   TSTBUDS.**
08:52:13   1999-06-18   RCONWAY    SYS0.IPLPARM                           IODFPK
08:52:13   1999-06-18   RCONWAY    SYS0.IPLPARM                           IODFPK
08:52:20   1999-06-18   RCONWAY    SYS0.IPLPARM                           IODFPK
14:28:21   1999-06-16   SLACKER    TSTBUDS.COMMNDS.CLIST                  PDGSY1   TSTBUDS.**
14:28:29   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.D0615                 PDGSY1   TSTBUDS.**
14:28:29   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.D0615                 PDGSY1   TSTBUDS.**
14:28:33   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.NONUM                 PDGSY1   TSTBUDS.**
14:28:33   1999-06-16   SLACKER    TSTBUDS.ICE.TOOL.NONUM                 PDGSY1   TSTBUDS.**
14:28:38   1999-06-16   SLACKER    TSTBUDS.ISPF.ISPPROF                   PDGSY1   TSTBUDS.**
14:28:38   1999-06-16   SLACKER    TSTBUDS.ISPF.ISPPROF                   PDGSY1   TSTBUDS.**
14:28:42   1999-06-16   SLACKER    TSTBUDS.JCL.CNTL                       PDGSY1   TSTBUDS.**
14:28:42   1999-06-16   SLACKER    TSTBUDS.JCL.CNTL                       PDGSY1   TSTBUDS.**
14:28:46   1999-06-16   SLACKER    TSTBUDS.RACFDB1.FLATFILE               PDGSY1   TSTBUDS.**
14:28:54   1999-06-16   SLACKER    TSTBUDS.RECJCL                         PDGSY1   TSTBUDS.**
```

Figure 488. Sample WARN report

```
- 1 -       HLQs With Excessive Fully Qual. Generic Profiles       99/06/18       10:54:40 am

HLQ                 Count
--------   ---------------
PASSWORD                1
SHARE                   1
SYS1                    1
```

Figure 489. Sample $CFQG Stand-Alone report

```
- 1 -       HLQs With Excessive Generic Profiles       99/06/18       11:00:36 am

HLQ                 Count
--------   ---------------
ALFREDC                 3
SYS1                    3
```

Figure 490. Sample $CHLQ report

Note: The sample report is shipped with a threshold of 200; for demonstration purposes we reset the threshold to 2.

```
- 1 -       User Defined Within the past 90 days        99/06/18       10:54:51 am

Date         User ID    Owner      Special   Operations   Auditor   Last Date    Last Time
----------   --------   --------   -------   ----------   -------   ----------   ---------
1999-06-17   ALICE      USERS      NO        NO           NO
1999-06-17   BRUCE      USERS      NO        NO           NO
1999-04-06   MCNAMEE    TSO        YES       YES          NO        1999-04-22   11:11:28
1999-06-17   MSHELL     TSO        NO        NO           NO
1999-04-26   NEWUSER    SYS1       NO        NO           NO
1999-06-01   PEKKAH     GRAAFF     YES       NO           NO        1999-06-17   10:39:26
1999-06-16   SLACKER    TSO        NO        NO           NO        1999-06-17   16:47:25
1999-06-16   TEDEE      TSO        NO        NO           NO        1999-06-16   13:54:39
1999-06-16   TEEDEE     TSO        NO        NO           NO        1999-06-17   16:47:24
1999-05-12   TONYNIX    GRAAFF     NO        NO           NO        1999-06-07   09:55:13
1999-06-15   TSTBUDS    GRAAFF     YES       NO           NO        1999-06-17   16:49:14
```

Figure 491. Sample $ULAST90 report.

Appendix D. RACF list of certificate authority certificates

This appendix contains a list of all certificate authorities certificates that are installed in the RACF database and that can be used in a RACF key ring.

The list of all certificate authority certificates can be retrieved by executing the following RACF command:

```
      RACDCERT LIST CERTAUTH

 Digital certificate information for user irrcerta:

   Label: Verisign Class 3 Primary CA
   Status: NOTRUST
   Start Date: 1996/01/28 19:00:00
   End Date:   2004/01/07 18:59:59
   Serial Number:
        >00E49EFDF33AE80ECFA5113E19A4240232<
   Issuer's Name:
        >OU=Class 3 Public Primary Certification Authority.O=VeriSign, Inc..C=<
        >US<
   Subject's Name:
        >OU=Class 3 Public Primary Certification Authority.O=VeriSign, Inc..C=<
        >US<
   Private Key Type: None
   Ring Associations:
   *** No rings associated ***

   Label: Verisign Class 2 Primary CA
   Status: NOTRUST
   Start Date: 1996/01/28 19:00:00
   End Date:   2004/01/07 18:59:59
   Serial Number:
        >00BA5AC94C053B92D6A7B6DF4ED053920D<
   Issuer's Name:
        >OU=Class 2 Public Primary Certification Authority.O=VeriSign, Inc..C=<
        >US<
   Subject's Name:
        >OU=Class 2 Public Primary Certification Authority.O=VeriSign, Inc..C=<
        >US<
   Private Key Type: None
   Ring Associations:
   *** No rings associated ***

   Label: Verisign Class 1 Primary CA
   Status: NOTRUST
   Start Date: 1996/01/28 19:00:00
   End Date:   2020/01/07 18:59:59
   Serial Number:
        >325033CF50D156F35C81AD655C4FC825<
   Issuer's Name:
        >OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=<

        >US<
   Subject's Name:
        >OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=<

        >US<
   Private Key Type: None
1  Ring Associations:
   *** No rings associated ***
```

```
Label: RSA Secure Server CA
Status: NOTRUST
Start Date: 1994/11/08 19:00:00
End Date:   2010/01/07 18:59:59
Serial Number:
     >02AD667E4E45FE5E576F3C98195EDDC0<
Issuer's Name:
     >OU=Secure Server Certification Authority.O=RSA Data Security, Inc..C=<

     >US<
Subject's Name:
     >OU=Secure Server Certification Authority.O=RSA Data Security, Inc..C=<

     >US<
Private Key Type: None
Ring Associations:
*** No rings associated ***

Label: Thawte Server CA
Status: NOTRUST
Start Date: 1996/07/31 19:00:00
End Date:   2020/12/31 18:59:59
Serial Number:
     >01<
Issuer's Name:
     >server-certs@thawte.com.CN=Thawte Server CA.OU=Certification Services<

     > Division.O=Thawte Consulting cc.L=Cape Town.SP=Western Cape.C=ZA<

Subject's Name:
     >server-certs@thawte.com.CN=Thawte Server CA.OU=Certification Services<

     > Division.O=Thawte Consulting cc.L=Cape Town.SP=Western Cape.C=ZA<

Private Key Type: None
Ring Associations:
*** No rings associated ***

Label: Thawte Premium Server CA
Status: NOTRUST
Start Date: 1996/07/31 19:00:00
End Date:   2020/12/31 18:59:59
Serial Number:
     >01<
Issuer's Name:
     >premium-server@thawte.com.CN=Thawte Premium Server CA.OU=Certificatio<

     >n Services Division.O=Thawte Consulting cc.L=Cape Town.SP=Western Cap<

     >e.C=ZA<
Subject's Name:
     >premium-server@thawte.com.CN=Thawte Premium Server CA.OU=Certificatio<

1    >n Services Division.O=Thawte Consulting cc.L=Cape Town.SP=Western Cap<

     >e.C=ZA<
Private Key Type: None
Ring Associations:
*** No rings associated ***
```

```
   Label: Thawte Personal Basic CA
   Status: NOTRUST
   Start Date: 1995/12/31 19:00:00
   End Date:   2020/12/31 18:59:59
   Serial Number:
        >00<
   Issuer's Name:
        >personal-basic@thawte.com.CN=Thawte Personal Basic CA.OU=Certificatio<

        >n Services Division.O=Thawte Consulting.L=Cape Town.SP=Western Cape.C<

        >=ZA<
   Subject's Name:
        >personal-basic@thawte.com.CN=Thawte Personal Basic CA.OU=Certificatio<

        >n Services Division.O=Thawte Consulting.L=Cape Town.SP=Western Cape.C<

        >=ZA<
   Private Key Type: None
   Ring Associations:
   *** No rings associated ***

   Label: Thawte Personal Freemail CA
   Status: NOTRUST
   Start Date: 1995/12/31 19:00:00
   End Date:   2020/12/31 18:59:59
   Serial Number:
        >00<
   Issuer's Name:
        >personal-freemail@thawte.com.CN=Thawte Personal Freemail CA.OU=Certif<

        >ication Services Division.O=Thawte Consulting.L=Cape Town.SP=Western <

        >Cape.C=ZA<
   Subject's Name:
        >personal-freemail@thawte.com.CN=Thawte Personal Freemail CA.OU=Certif<

        >ication Services Division.O=Thawte Consulting.L=Cape Town.SP=Western <

        >Cape.C=ZA<
   Private Key Type: None
   Ring Associations:
   *** No rings associated ***

   Label: Thawte Personal Premium CA
   Status: NOTRUST
   Start Date: 1995/12/31 19:00:00
   End Date:   2020/12/31 18:59:59
   Serial Number:
1       >00<
   Issuer's Name:
        >personal-premium@thawte.com.CN=Thawte Personal Premium CA.OU=Certific<

        >ation Services Division.O=Thawte Consulting.L=Cape Town.SP=Western Ca<

        >pe.C=ZA<
   Subject's Name:
        >personal-premium@thawte.com.CN=Thawte Personal Premium CA.OU=Certific<

        >ation Services Division.O=Thawte Consulting.L=Cape Town.SP=Western Ca<

        >pe.C=ZA<
   Private Key Type: None
   Ring Associations:
   *** No rings associated ***
```

```
Label: IBM World Registry CA
Status: NOTRUST
Start Date: 1997/05/20 06:34:25
End Date:   2017/05/20 06:34:25
Serial Number:
     >33820AD2<
Issuer's Name:
     >CN=IBM World Registry Certification Authority.O=IBM World Registry.C=<

     >US<
Subject's Name:
     >CN=IBM World Registry Certification Authority.O=IBM World Registry.C=<

     >US<
Private Key Type: None
Ring Associations:
*** No rings associated ***

Label: Integrion CA
Status: NOTRUST
Start Date: 1997/05/20 05:03:48
End Date:   2017/05/20 05:03:48
Serial Number:
     >3381F595<
Issuer's Name:
     >CN=Integrion Certification Authority Root.O=Integrion Financial Netwo<

     >rk.C=US<
Subject's Name:
     >CN=Integrion Certification Authority Root.O=Integrion Financial Netwo<

     >rk.C=US<
Private Key Type: None
Ring Associations:
*** No rings associated ***
```

Appendix E. VPN planning worksheet

Using the following template, specify the information as part of the planning for your dynamic VPN configuration. Create a worksheet for each TCP/IP stack you plan to configure with a dynamic tunnel.

Table 15. VPN planning worksheet

VPN Parameter	Value
Key Policy, Proposal, Transform	
Initiator Negotiation	
Responder Negotiation	
Authentication Method	
Hash Algorithm	
Encryption Algorithm	
Diffie-Hellmann Group	
Maximum Key Lifetime	
Maximum Size Limit	
Key Lifetime Range	
Size Limit Range	
Data Policy, Proposal, AH and ESP Transform	
PFS (Perfect Forward Secrecy)	
AH Encapsulation Mode	
AH Authentication Algorithm	
AH Maximum Data Lifetime	
AH Maximum Size Limit	
AH Data Lifetime Range	
AH Size Limit Range	
ESP Encapsulation Mode	
ESP Authentication Algorithm	
ESP Encryption Algorithm	
ESP Maximum Data Lifetime	
ESP Maximum Size Limit	
ESP Data Lifetime Range	
ESP Size Limit Range	
Dynamic Tunnel Policy	
Initiation	
Connection Lifetime	

VPN Parameter	Value
Authentication Information	
Remote Key Server	
Authentication Method	
Shared Key	
Certificate Authority	
Racdcert Label	
Key Ring	
User ID	
Key Ring Name	
Dynamic Connection	
Source	
Destination	
Source Port	
Destination Port	
Automatic Activation	
Protocol	
Remote Key Server	
Key Servers	
Local Key Server ID Type	
Local Key Server ID	
Remote Key Server ID Type	
Remote Key Server ID	

Appendix F. Special notices

This publication is intended to help system programmers, system integrators and Webmasters to configure the OS/390 Security Server for e-business. The information in this publication is not intended as the specification of any programming interfaces that are provided by SecureWay Security Server for OS/390. See the PUBLICATIONS section of the IBM Programming Announcement for the SecureWay Security Server for OS/390 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®
AIX
AIX/6000
APL2
AS/400
AT
CICS
CT
DB2
DFSMS
DFSORT
eNetwork
ESCON
FAA
IMS
MQ
Netfinity
NetView
OpenEdition
OS/2
OS/390
OS/400
Parallel Sysplex
QMF
RACF
RMF
RS/6000
S/370
S/390
SecureWay
SP
SP1
System/390
VTAM
WebSphere
World Registry
XT
3090
400

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix G. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

G.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 423.

- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149 (version 01)
- *OS/390 Security Server 1999 Updates: Technical Presentation Guide*, SG24-5627
- *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046
- *A Comprehensive Guide to Virtual Private Networks, Volume III: IBM Cross-Platform and Key Management Solutions*, SG24-5309
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *Understanding LDAP,* SG24-4986
- *Ready for e-business: OS/390 Security Server Enhancements,* SG24-5158
- *LDAP Implementation Cookbook,* SG24-5110
- *OS/390 V2R6 UNIX System Services Implementation and Customization,* SG24-5178
- *OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing,* SG24-5227
- *OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 2: UNIX Applications,* SG24-5228
- *OS/390 eNetwork Communications Server Volume 3: MVS Applications,* SG24-5229
- *OS/390 Security Server Audit Tool and Report Application*, SG24-4820

G.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

G.3 Other resources

These publications are also relevant as further information sources:

- *OS/390 Firewall Technologies Guide and Reference*, SC24-5835
- *Open Cryptographic Services Facility Application Developer's Guide and Reference*, SC24-5875
- *OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference*, SC24-5877
- *OS/390 Initialization and Tuning Reference*, SC28-1752
- *OS/390 UNIX System Services Planning*, SC28-1890
- *OS/390 UNIX System Services User's Guide*, SC28-1891
- *OS/390 UNIX System Services Command Reference*, SC28-1892
- *OS/390 Security Server (RACF) Command Language Reference*, SC28-1919
- *OS/390 SecureWay Communications Server IP Migration*, SC31-8512
- *OS/390 SecureWay Communications Server IP Configuration*, SC31-8513
- *OS/390 Security Server Open Cryptographic Enhanced Plug-ins Guide and Reference*, SA22-7429
- *OS/390 Security Server LDAP Server Administration and Usage Guide*, SC24-5861
- *OS/390 Security Server (RACF) Security Auditor's Guide,* SC28-1916
- *OS/390 Security Server (RACF) Macros and Interfaces*, SC28-1914
- *OS/390 Security Server (RACF) Systems Administrators Guide*, SC28-1915
- *DFSORT Application Programming Guide*, SC33-4035
- *WebSphere Application Server for OS/390 HTTP Server Planning, Installing, and Using*, SC31-8690

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number

☐ Credit card number

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

X

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5629-00 OS/390 Security Server 1999 Updates: Installation Guide
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

OS/390 Security Server 1999 Updates: Installation Guide

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

OS/390 Security Server 1999 Updates: Installation Guide

Redbooks

Exploit RACF's Public Key Infrastructure enhancements

Configure and install Directory Services (LDAP)

Tunnel through to OS/390 using IKE

This redbook will help you to install, configure and exploit the latest security enhancements to the SecureWay Security Server for OS/390 and the IBM Communication Server.

Using examples, we show you how to maximize the new Public Key Infrastructure enhancements that have been made to RACF, as well as exert more control over UNIX System Services superusers by using the new RACF security facilities.

With Directory Services becoming ever more important in an e-business security infrastructure, we illustrate how to set up LDAP on OS/390.

And as Virtual Private Network (VPN) becomes more accepted, OS/390 has a role in this area too. The new Internet Key Exchange (IKE) support will enable interoperability with other platforms, firewall, and VPN clients, and we describe how you can set up VPN support on OS/390.

This redbook is the companion to OS/390 Security Server 1999 Updates: Technical Presentation Guide, SG24-5627.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5629-00 ISBN 0738416347