

---

## Chapter 4. RACF Enhancements

This chapter details the enhancements made to the RACF element of the OS/390 Security Server, such as:

- Digital Certificate Support.

This include a SPE that is a replacement for the HTTP CA Servlet function, that was previously ship with the IBM HTTP Server for OS/390.

- Program Control Enhancements
- RVAR Y Enhancement

For more information regarding Digital Certificates, see the following red books:

*OS/390 Security Server 1999 Updates Technical Presentation Guide, SG24-5627.*

*Ready for e-business OS/390 Security Server Enhancements, SG24-5158*

---

### 4.1 Digital Certificate Introduction

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 OS/390 resources.

A crucial part of implementing certificates is managing the certificates used by server application, and ensuring is an uncompromising 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 provides functions to help managed 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 for security server. The functions are incorporated into two plug-ins: one for data library services and one for trust policy manager. OCEP functions are to be used by application complying with Common Data Security Architecture (CDSA) standard interfaces. This makes it easier for application developers and Independent Software Venders (ISVs) to develop

and port applications to the OS/390 platform. It also helps customers apply consistent security rules to e-business application that use digital certificates.

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. The system was rolled back to Release 2 Version 8 to ensure this function would work.

In OS/390 Security Server Version 2 Release 10 the `RACDCERT` command has some additional parameters of the keyword `GENCERT` and `EXPORT` along with an additional keyword `DEBUG`. The new components of the `RACDCERT` command are shown in Figure 61

```

RACDCERT

GENCERT

[KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN CERTSIGN)]

[ALTNAME[ (IP(numeric-ip-address)

           DOMAIN('internet-domain-name') EMAIL('email-address')

           UR('universal-resource-identifier')

EXPORT [FORMAT(CERTDER CERTB64 PKCS12DER PKCS12B64)]

DEBUG

```

Figure 61. New components of `RACDCERT` command.

`KEYUSAGE` specifies the appropriate values for the KeyUsage certificate extension, of which one or more of the values might be coded. For certificate authority certificates, the default is `CERTSIGN` and is always set. There is no default for certificates that are not certificate-authority certificates.

The subparameter of the keyword `KEYUSAGE` are displayed in Table 6

Table 6. Subparameters of keyword `KEYUSAGE`

SUBPARAMETER	DESCRIPTION
HANDSHAKE	Most commonly used Facilitates identification and key xchange during security handshakes, such as SSL, which set the digitalSignature and keyEncipherment indicators. Used with Server, Clients and Firewalls.

SUBPARAMETER	DESCRIPTION
DATAENCRYPT	Encrypts data, which sets the dataEncipherment indicator. Needed for Firewalls.
DOCSIGN	Specifies a legally binding signature, which sets the nonRepudiation indicator.
CERTSIGN	Specifies a signature for other digital certificates and CRLs, which sets the keyCertSign and cRLSign indicators,. Acts as or is the Certificate Authority for signing off certificates.

`ALTNAME` specifies the appropriate values for the `subjectAltName` extension, of which one or more of the values might be coded. If required for the extension, RACF converts the entire value ASCII.

Note: RACF assumes the terminal code page is IBM-1047 and translates to ASCII accordingly.

The subparameter of the keyword `ALTNAME` are displayed in Table 7

Table 7. Subparameter of keyword `ALTNAME`

SUBPARAMETER	DESCRIPTION
<code>IP('numeric-ip-address')</code>	Specifies a quoted string containing a fully qualified 'numeric-ip-address' in IPV4 dotted decimal form, which is four decimal numbers (each number must be a value from 0-255) separated by periods. i.e. 9.12.14.247
<code>DOMAIN('internet-domain-name')</code>	Specifies a quoted string containing a fully qualified 'internet-domain-name' such as 'www.widgits.com'. RACF does not check this value's validity.
<code>EMAIL('email-address')</code>	Specifies a quoted string containing a fully qualified 'e-mail-address' such as 'homer at moes bar.com'. RACF replaces the word "at" with the @ symbol (x'7C') to conform with RFC822. If RACF cannot locate the word "at" it assumes the address is already in RFC822 form and makes no attempt to alter it other than converting it to ASCII.

SUBPARAMETER	DESCRIPTION
URI('universal-resource-identifier')	Specifies the 'universal-resource-identifier' such as 'http://www.widgits.com'. RACF does not check the validity of this value.

EXPORT FORMAT(format-type) specifies the format of the exported certificate. Note that while the CERT keywords indicate to export only a certificate, the PKCS#12 keywords to export the certificate and the private key (which must exist and not be an ICSF key). The package produced by specifying one of the PKCS keywords is encrypted using the password specified according to the PKCS#12 standard.

The Values of the subparameter `FORMAT` are displayed in Table 8

Table 8. Values of subparameter `FORMAT`

VALUE	DESCRIPTION
CERTDER	Specifies a DER encoded X.509 certificate.
CERTB64	Specifies a DER encoded X.509 certificate that has been encoded using Base64
PKCS12B64	Specifies a DER encoded (then Base64 encoded) PKCS#12 package.
PKCS12DER	Specifies a DER encoded PKCS#12 package. This value is now the more acceptable format to be specified.

PKCS12B64 is the default if password is specified, otherwise CERTB64 is the default.

DEBUG displays additional diagnostic information pertaining to encryption calls and RACF invoked `ICHEINTY ALTER`, `RACROUTE REQUEST=EXTRACT`, and `RACROUTE REQUEST=DEFINE` failures. When a problem is encountered, customers can use this keyword to gather diagnostic information for the IBM support center.

We can now use RACF to create, register, store and administer digital certificates and their associated private keys, and to build certificates request that can be sent to certificate authority for signing. Digital certificates are managed in RACF using the `RACDCERT` command or by using an application that invokes `R_data1ib` callable service (IRRSDL00) or the `initACEE` callable service (IRRSIA00). The `R_data1ib` callable service provides an application programming interface to the Common Data Security Architecture (CDSA) data library functions, and is used by secure sockets layer (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.

#### 4.1.1 Generating a Digital Certificate to RACF

This section describes how to define a digital certificate using RACF.

1. We first use the `RACDCERT GENCERT` command to define the digital certificate. with one of the new parameters `KEYUSAGE` of the keyword `GENCERT`.

**Note:** That the label parameter is case sensitive:

```
RACDCERT GENCERT SUBJECTSDN(CN('JONATHAN BRIGGS') OU('ITSO')
O('IBM') L('POUGHKEEPSIE') SP('NEW YORK') C('US')) SIZE(1024)
WITHLABEL('JONBOY') SIGNWITH(CERTAUTH LABEL('ITSO CA'))
KEYUSAGE(HANDSHAKE)
```

The informational message `IRRD113I` as shown in Figure 62 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`.

```
IRRD113I The certificate that you are creating has an incorrect
date range. The certificate is added with NOTRUST status.
```

*Figure 62. Information message from the `RACDCERT GENCERT` command.*

2. To confirm the expiry date on the ITSO CA Certificate Authority `certauth` label we can display the label using the `RACDCERT CERTAUTH LIST` command:

```
RACDCERT CERTAUTH LIST(LABEL('ITSO CA'))
```

The start and end dates for the certificate are shown in Figure 63 for the Certificate Authority:

```
Digital certificate information for CERTAUTH:

Label: ITSO CA
Status: TRUST
Start Date: 2000/02/02 01:00:00
End Date: 2001/02/03 00:59:59
Serial Number:
  >00<
Issuer's Name:
  >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=NY.C=US<
Subject's Name:
  >CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=NY.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
  Ring Owner: GRAAFF
  Ring:
    >pauls-keyring<
  Ring Owner: TCPIPU
  Ring:
    >telnetserver<
```

Figure 63. Output from `RACDCERT CERTAUTH LIST` command

We can list of the digital certificate with the `RACDCERT LIST` command:

```
RACDCERT ID(BRIGGS) LIST(LABEL('JONBOY'))
```

From the output we can see the digital certificate we generated using the `RACDCERT GENCERT` command picked up today's date, along with the `NOTRUST STATUS` as shown in Figure 64:

```

Digital certificate information for user BRIGGS:

Label: JONBOY
Status: NOTRUST
Start Date: 2000/06/14 00:00:00
End Date: 2001/06/14 23:59:59
Serial Number:
>08<
Issuer's Name:
>CN=IBM-ITSO MAIN CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=NY.C=US<
Subject's Name:
>CN=Jonathan Briggs.OU=itso.O=ibm.L=poughkeepsie.SP=new york.
C=us<
Key Usage: HANDSHAKE
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***

```

Figure 64. Output from the RACDCERT LIST command.

3. We recommend to amend the newly created digital certificate so it is in the **TRUST** status:

```
RACDCERT ID(BRIGGS) ALTER(LABEL('JONBOY')) TRUST
```

If we now use the RACDCERT LIST command again we would be able to see that the digital certificate is now in **TRUST** status, so it can be used through an OS/390 client server.

4. Now that we have generated the certificate for an OS/390 client server we need to export into an OS/390 MVS dataset, using the RACDCERT EXPORT command, we can export the digital certificate in any one of the four formats mention in Table 8 on page 54, in this example we recommend using PKCS12DER.

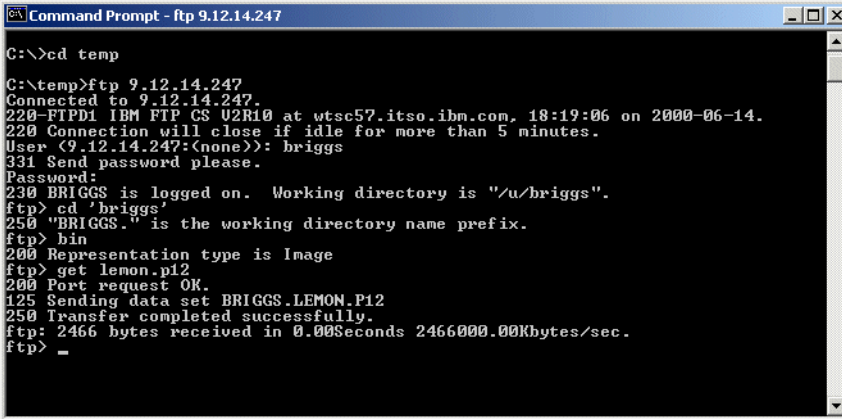
```
RACDCERT ID(BRIGGS) EXPORT(LABEL('JONBOY')) DSN(LEMON.P12)
FORMAT(PKCS12DER) PASSWORD('password')
```

The output dataset containing the digital certificate in a DER encoded X.509 certificate format, can now be defined to the Netscape Navigator and Internet Explorer. The procedure is completely different for both browsers.

### 4.1.2 Downloading the Digital Certificate

To download the digital certificate to the workstation, do the following:

- From Windows we opened up a Command Prompt session from the Start Menu. At the C: prompt in this example we changed directory to the TEMP directory for which our digital certificate is downloaded to. We now logon to our OS/390 host system using FTP. Once connected we enter our RACF User ID and password. After successfully connecting we then change directory to our the High Level Qualifier where the digital certificate is held next we typed in bin so we can download the digital certificate in to binary format. Now we issue the get command for our digital certificate. See Figure 65 for this sequence of events:



```
C:\>cd temp
C:\temp>ftp 9.12.14.247
Connected to 9.12.14.247.
220-FTPDI IBM FTP CS U2R10 at wtsc57.itso.ibm.com. 18:19:06 on 2000-06-14.
220 Connection will close if idle for more than 5 minutes.
User (9.12.14.247:(none)): briggs
331 Send password please.
Password:
230 BRIGGS is logged on. Working directory is "/u/briggs".
ftp> cd 'briggs'
250 "BRIGGS." is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp> get lemon.p12
200 Port request OK.
125 Sending data set BRIGGS.LEMON.P12
250 Transfer completed successfully.
ftp: 2466 bytes received in 0.00Seconds 2466000.00Kbytes/sec.
ftp> _
```

Figure 65. Command Prompt Screen.

### 4.1.3 Importing the Digital Certificate to Internet Explorer

The following steps will import the digital certificate into the internet explorer.

1. By using Windows Explorer we can find and select the file in the directory we saved the file to and by double clicking on the file we invoke the Certificate Import Wizard program as shown in Figure 66 on page 59:





Figure 66. The Welcome to the Certificate Import Wizard.

The Certificate Import Wizard guides us through the correct process to properly install the digital certificate into right certificate store of the Internet Explorer browser.

2. Press the **Next** button, to continue the installation of the digital certificate. See Figure 67:

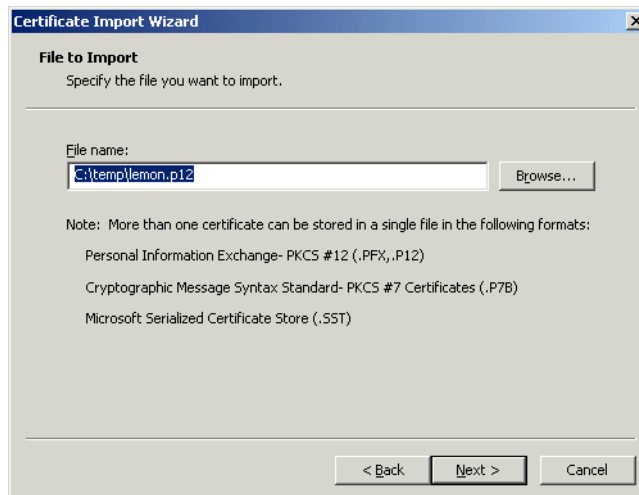


Figure 67. File to Import screen

Verify that the path is correct and click on the **Next** button. See Figure 68 on page 60 for the Password prompt screen. This is the password you specified

when you exported the file using the `RACDCERT EXPORT` command. Select the box option opposite Mark the private key as exportable, then click on the **Next** button.

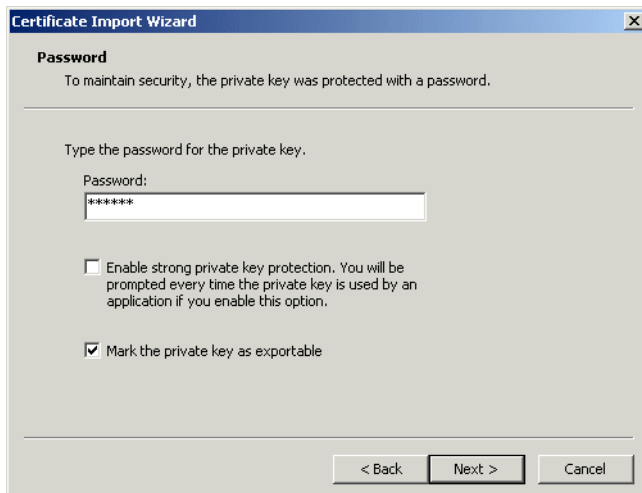


Figure 68. Password prompt screen

3. Figure 69 allows us to specify where the key is to be stored, we recommend that you click on the box opposite Automatically select the certificate store based on the type of certificate. this lets the Certificate Import Wizard automatically decide where to store our digital certificate, rather than a manual intervention:

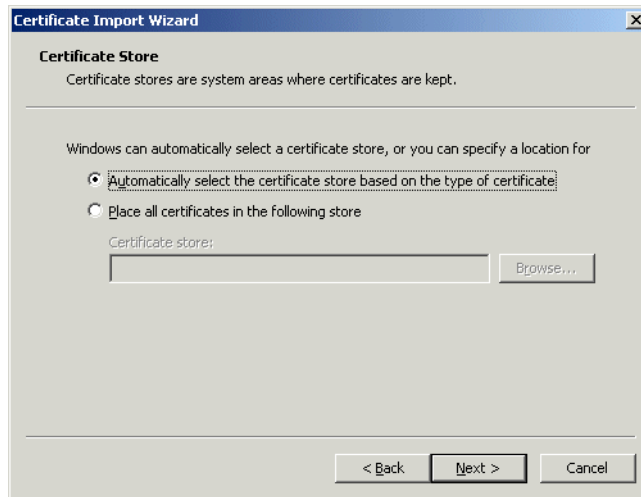


Figure 69. Certificate Store screen

**Note:** By using this method we can export the digital certificate out of the browser.

4. Press the **Next** button to continue the installation of the digital certificate:

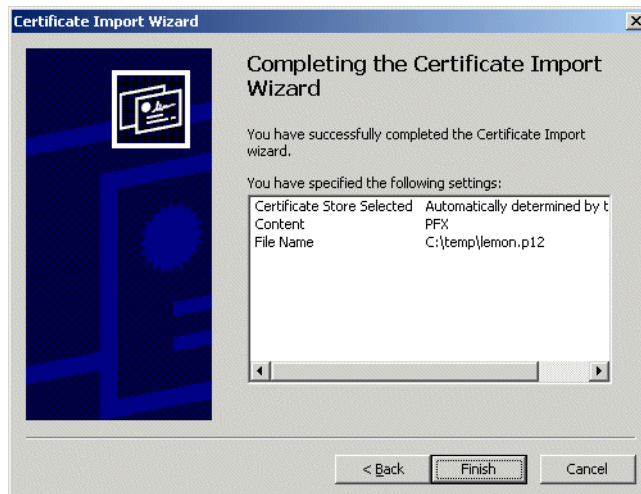


Figure 70. the Complete the Certificate Import Wizard screen.

The Certificate Import Wizard informs us that the installation is now complete for the install of the digital certificate as shown in Figure 70. Press the **Finish** button to continue.

A successful import prompt is displayed by the Certificate Import Wizard as shown in Figure 71 when the import of the digital certificate is successfully completed.



Figure 71. Certificate Import Wizard OK screen.

5. We need to verify that the digital certificate indeed allows for client authentication. Click on the **Tools** option on the task bar then select the **Internet Options**. This brings up the Internet Options panel see Figure 72, click on the **Content** tab:

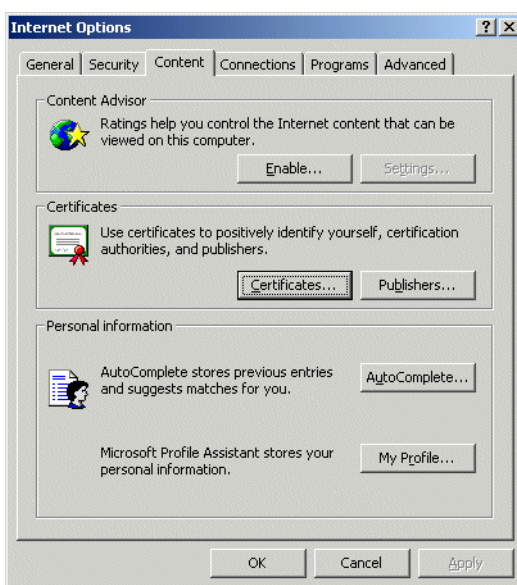


Figure 72. Internet Options panel.

Click on the **certificates** button, to bring up a list of Certificates. Highlight the digital certificate and click on the **Advanced** button, see Figure 73 on page 63:

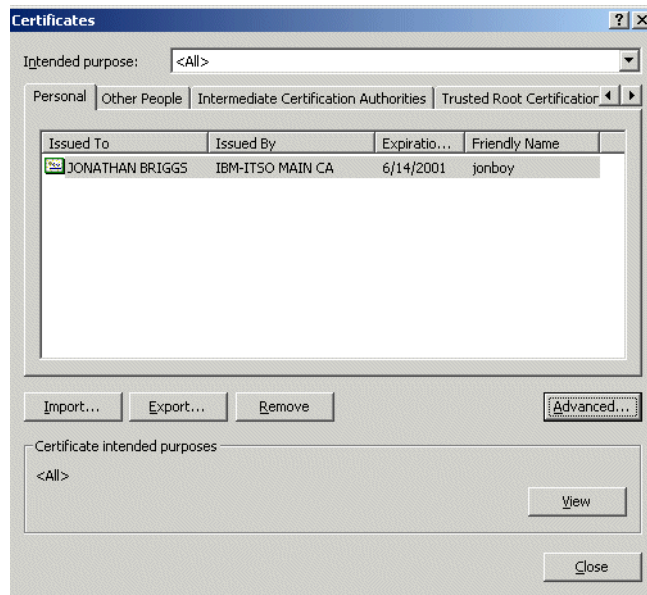


Figure 73. Certificates screen

The Advanced screen appears with various certificate purposes ticked. For Client Authentication by default is not selected. Tick the **Client Authentication** box and then click **OK** to save. As shown in Figure 74:

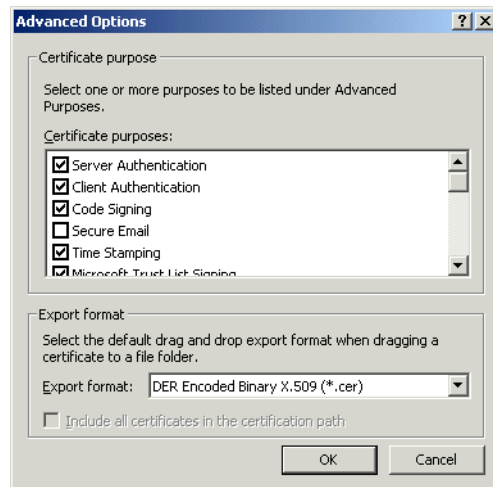


Figure 74. Advanced Options screen

#### 4.1.4 Importing the Digital Certificate to Netscape Navigator

1. We now have to import the digital certificate into the Netscape Navigator browser. Open up the Netscape Navigator by clicking on the **Netscape Navigators** icon. Once the Netscape Navigator is open click on the **Security** icon on the task bar, this opens up the Security Info panel, as shown in Figure 75:

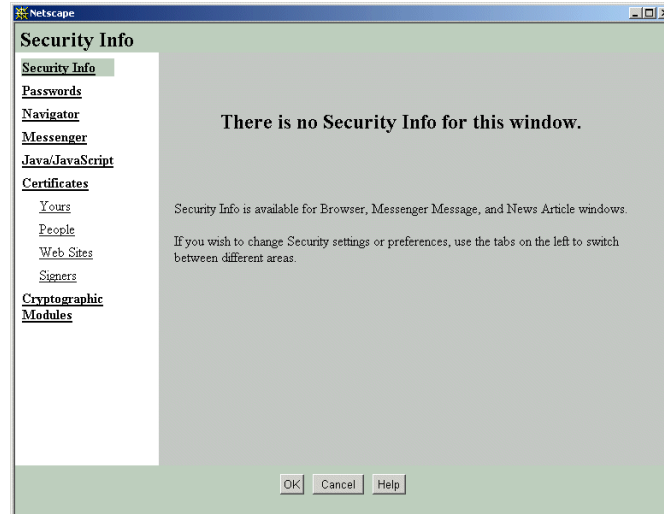


Figure 75. Netscape Navigators Security Info panel.

2. The Security window shows a Certificate option, where you select **Yours**, to perform the operation to import the digital certificate. The Security window changes, and shows your digital certificates stored in the certificate database as shown in Figure 76 on page 65:

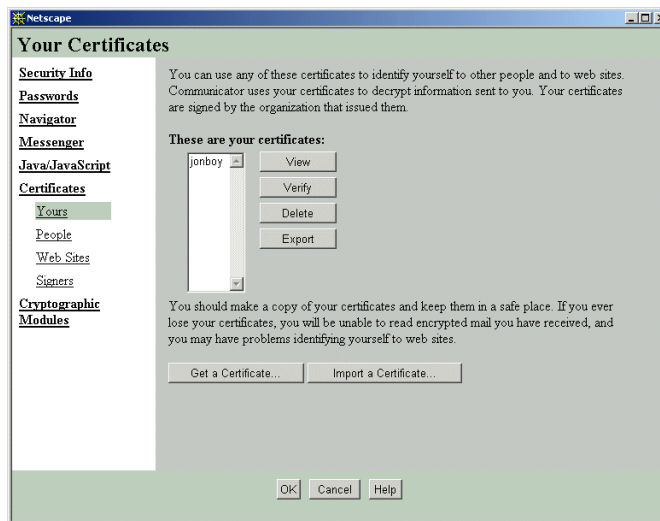


Figure 76. Your Certificates panel.

3. We now click on the **Import a Certificate** button, which will take us into the Windows Explorer. Locate the digital certificate to be imported and open the file, as shown in Figure 77:

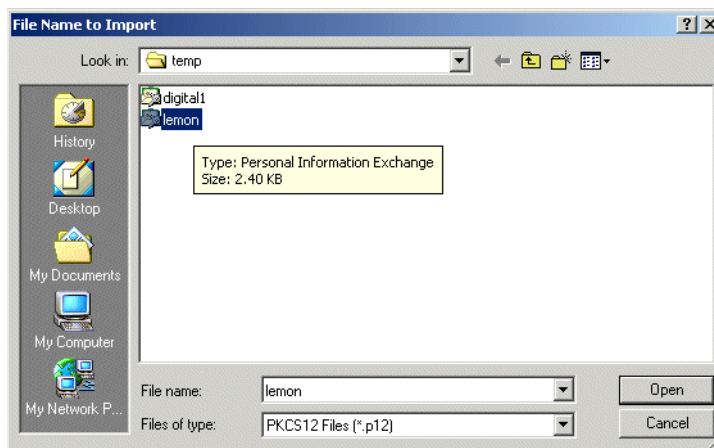


Figure 77. File name to import screen

A confirmation message Your certificates have been successfully imported prompt is then displayed, as shown in Figure 78 on page 66:

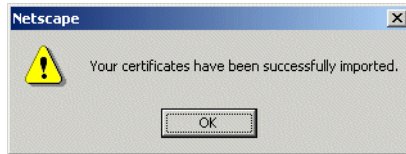


Figure 78. Confirmation prompt

Now that the digital certificate has been imported we can view the information about the digital certificate, by highlighting the digital certificate and clicking on View. As shown in Figure 79:



Figure 79. Your Certificates panel

Figure 80 on page 67 displays information about the digital certificate i.e. who the certificate belongs to and who was the issuer etc.





Figure 80. Details of digital certificate

---

## 4.2 RACF's web interface to digital certificate generation

As you may have determined from reading through 4.1, "Digital Certificate Introduction" on page 51, to install a client certificate using the procedures outlined there, is not the most user friendly way.

RACF APAR OW45211 and SAF APAR OW45212 introduces a Web based front-end into digital certificates generation for clients using a Web browser such as, Netscape Navigator or Microsoft's Internet Explorer (IE).

This is extending on Public Key Infrastructure (PKI) services we supplied with APAR OW31933, that provided the RACF Auto Registration (RAR) Application. The RAR application shipped sample HTML and REXX code to provide Web based front-end into linking a client's digital certificate with it's associated RACF User ID.

These new services provide again sample HTML and REXX code to generate client digital certificates, as is it aimed at replacing the functionality supplied earlier by the CA Servlet that shipped with the IBM HTTP Server for OS/390. As of OS/390 Version 2 Release 10, this function is no longer shipped.

## 4.2.1 Overview

Figure 81 shown an overview of the components involved, when generating a client's digital certificate using the Web based front-end.

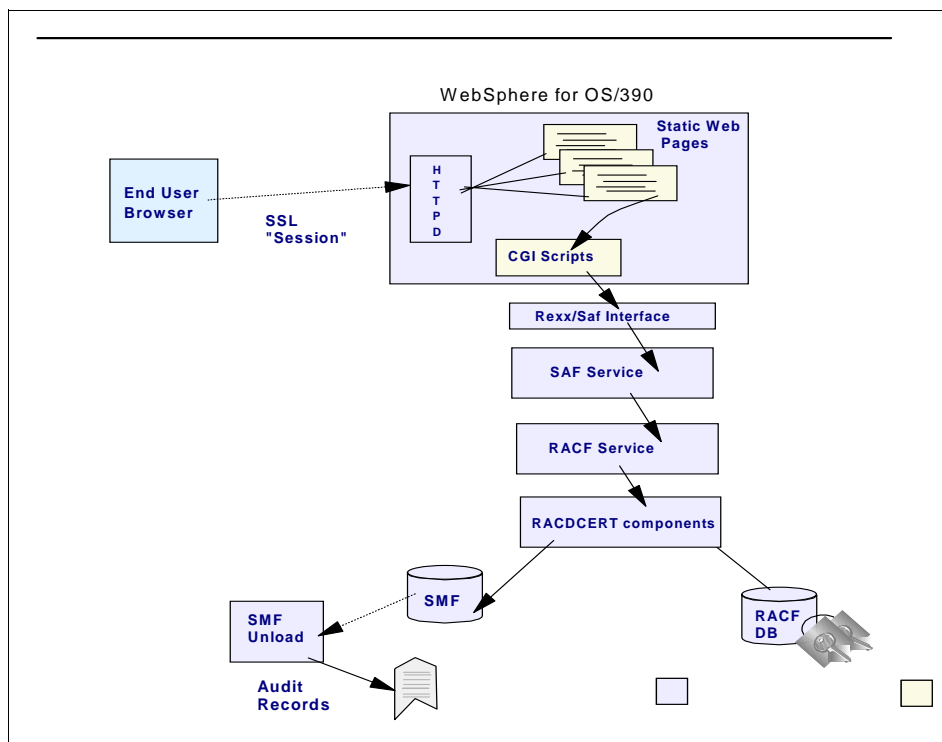


Figure 81. Overview of the Web based process of generating client certificates

The certificate request flow is as follow:

1. The client selects the web page, either linked or directly, to generate a digital certificate over an SSL established session.

**Note:** SSL server side authentication only.

2. The client fills in the web page with the required information to make up the distinguished name of the client like Common Name, Organization etc.

**Note:** The required fields can be tailored, according to the options defined in the configuration file, which we will discuss later.

3. The client submits the request and new RACF callable services will invoke the existing RACF `RACDCERT` facilities to generate a digital certificate request.
4. The client is notified of the successful or unsuccessful certificate request.

The client is issued a transaction identifier, that is used to pick up the approved certificate.

5. The client can now pick up his certificate based on the transaction identifier and install it into it's browser. The client also has the option to defer the pick up of his certificate to a later date or time.

**Note:** There is no formal approval process involved here. The setup assumes the client is authorized by RACF to generate a certificate and have it signed by the Certificate Authority (CA) selected. This is explained later in the customization section.

#### 4.2.2 Directory structure provided

RACF APAR OW45211 and SAF APAR OW45212 supplies the files and separate directories according to the following directory structure, as shown in Figure 82.

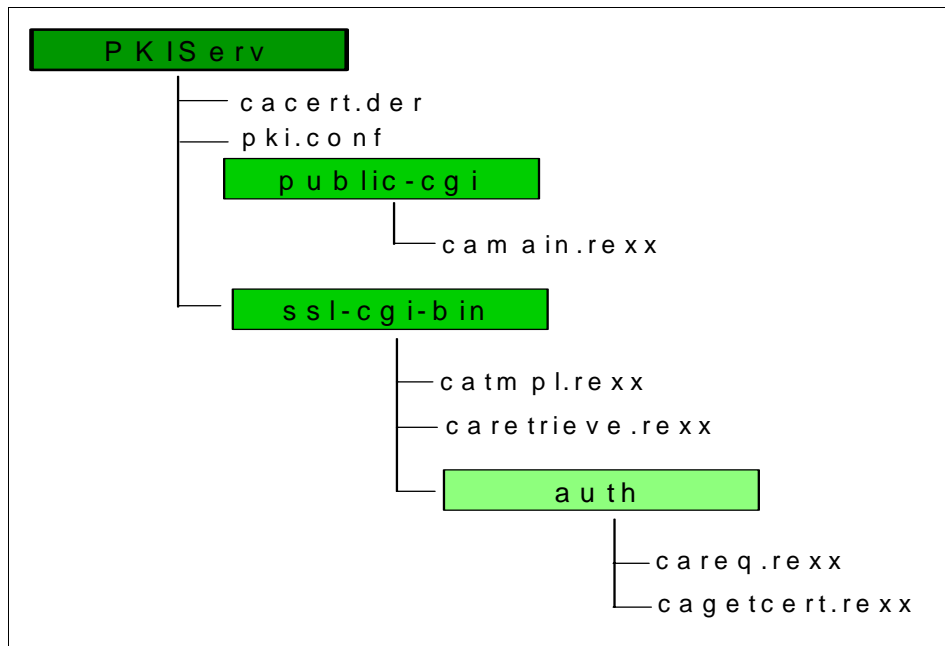


Figure 82. Directory Structure provided by APAR

The directory structure consists of the following root directory:

- PKIServ** this is the root directory and contains the following files:
- cacert.der** this is the local certificate authority (CA) signing certificate
  - pki.conf** this is the main configuration file, and basically controls the setup of the new functions provided.

The `PKIServ` directory contain two sub directories, called:

- public-cgi** this directory contains the main program, called `CAMAIN.REXX`
- ssl-cgi-bin** this directory contains the executables, that need to be executed only when an SSL connection is established between the client and the Web server.

The `SSL-CGI-BIN` directory contains one more sub directory, called:

- auth** this directory contains the code required to request and retrieve a digital certificate and requires both an SSL established session and a RACF User ID and password.

**Note:** The supplied directory structure can be changed if required, but would require changes to the files itself. Of course the main directory PKIServ can be installed as a subdirectory structure within your file system if required.

### 4.2.3 Explanation of the configuration file

The configuration of the new PKI services is done through a file, called `PKI.CONF` located in the root directory. It introduces the concept of certificate templates which define the fields that comprise a specific certificate request. These templates are examined by the REXX connector logic within the web server to establish defaults and identify the fields which can be changed on a certificate request. Certificate templates are shipped in pseudo HTML, which enables you to easily modify to suit your installation's needs. In essence, much of the intelligence with respect to certificate generation is contained within the certificate template, with the new SAF service providing the basic storage and management primitives.

#### 4.2.3.1 Structure of the configuration file

The file contains a mixture of true HTML and HTML like tags. The main tags divide the file into sections, `APPLICATION`, `TEMPLATE`, and `INSERT`, where `APPLICATION` and `TEMPLATE` may contain various subsections, named fields, and substitution variables as explained next.

##### **<APPLICATION NAME=appl-name> ... </APPLICATION>**

This section identifies the applications that will make use of PKI Services For OS/390. The product ships with one application defined, "WEBCA". This sections contains one subsection, `CONTENT`.

1. `<CONTENT> ... </CONTENT>`

This subsection contains the HTML to be presented to the end user requesting and retrieving certificates. The subsection should contain one or more named fields identifying certificate templates to be used for requesting or managing certificates through this application. (See next for a description of named fields.) These template names should match the HTML selection value associated with them.

##### **<TEMPLATE NAME=tpl-name> ... </TEMPLATE>**

This section defines the certificate templates referenced in the `APPLICATION` sections. Applicable subsections are:

1. `<CONTENT> ... </CONTENT>`

This subsection contains the HTML to be presented to the end user requesting certificates of this type. Any named fields in this subsection are interpreted as certificate field names defined by `INSERT` sections. (See next

for a description of INSERT sections.) For WEBCA, the `INSERT` sections are included as part of the HTML presented to the end user. (i.e., the end user provides values for these fields.) Named fields in this subsection are considered optional if the named field contains more than one word within the `%%` delimiters, e.g., `%%AltName (Optional)%%`. The user need not supply a value for `AltName`

2. `<APPL> ... </APPL>`

This subsection identifies certificate fields that the application itself should provide values for. This subsection should contain named fields only, one per line. Currently, the only supported named field allowed in this section is "Userld"

3. `<CONSTANT> ... </CONSTANT>`

This subsection identifies certificate fields that have a constant (hardcoded) value for everyone. This subsection should contain named fields only, one per line. The syntax for specifying the values is `%%field-name=field-value%%`, e.g., `%%KeyUsage=handshake%%`

4. `<SUCCESSCONTENT> ... </SUCCESSCONTENT>`

This subsection contains the HTML to be presented to the end user when the certificate request was submitted successfully. Any named fields in this subsection are interpreted as content inserts defined by `INSERT` sections. For WEBCA, the `INSERT` sections are included as part of the HTML presented to the end user.

5. `<FAILURECONTENT> ... </FAILURECONTENT>`

This subsection contains the HTML to be presented to the end user when the certificate request submit failed. Any named fields in this subsection are interpreted as content inserts defined by `INSERT` sections. For WEBCA, the `INSERT` sections are included as part of the HTML presented to the end user.

6. `<RETRIEVECONTENT> ... </RETRIEVECONTENT>`

This subsection contains the HTML to be presented to the end user to enable certificate retrieval. Any named fields in this subsection are interpreted as content inserts defined by `INSERT` sections. For WEBCA, the `INSERT` sections are included as part of the HTML presented to the end user.

7. `<RETURNCERT> ... </RETURNCERT>`

This subsection contains the HTML to be presented to the end user upon successful certificate retrieval. For WEBCA, if the certificate being retrieved is a browser certificate, then this section must contain a single line containing a browser qualified `INSERT` name, e.g.,

%%returnbrowsercert[browsertype]%%. Additionally, `INSERTs` for Netscape (returnbrowsercertNS) and Internet Explorer (returnbrowsercertIE) containing browser specific HTML for returning certificates must be defined elsewhere in the configuration file. If the certificate being retrieved is a server certificate, this section should contain the HTML necessary to present the certificate to the user as text

8. `<INSERT NAME=insert-name> ... </INSERT>`

This section contains HTML that either describes a certificate field or defines other common HTML that may be referenced in the `TEMPLATE` sections. `INSERTs` are referenced elsewhere by using a *named field* of the form `%%insert-name%%`

Named Fields are delineated with `%%`, e.g., `%%Label%%`. Their meaning is specific to the section they are contained in. Named fields are case sensitive. Named fields are also using to reference common includeable HTML. Note, WEBCA treats named fields that begin with a dash as just includeable code. Any special meaning a named field may have, given the section it's contained in, is ignored if it begins with a dash. For example, if `%%-pagefooter%%` was specified in a `TEMPLATE CONTENT` section, `-pagefooter` would not be considered a certificate field name. However, the `INSERT` with the name `-pagefooter` would be included in the HTML page presented to the end user.

Substitution variables are delineated with square brackets, e.g., `[base64cert]`. They represent variables that get replaced with an actual value at run time. Substitution variables are case sensitive. The valid substitution variables are:

<b>transactionid</b>	Unique value returned from a certificate request
<b>tmplname</b>	Certificate template name. Primed from the HTML tag <code>&lt;SELECT NAME="Template"&gt;</code> in the <code>&lt;APPLICATION NAME=WEBCA&gt;</code> section. This is selected by the end user on the first web page.
<b>base64cert</b>	The requested certificate base64 encoded
<b>iecert</b>	The requested certificate in a form the Microsoft Internet Explorer accepts
<b>browsertype</b>	Special substitution variable to be used to qualify named fields only. It use enables the different browsers, Netscape and Internet Explorer, to perform browser specific operations, i.e., Netscape uses a <code>KEYGEN</code> HTML tag to generate a public/private key pair while Internet Explorer uses <code>ACTIVEX</code> controls. For example, if <code>%%PublicKey[browsertype]%%</code> was specified in a <code>TEMPLATE CONTENT</code> section referenced by a user with the Netscape

Navigator browser then `INSERT PublicKeyNS` would be included. Likewise, if the user's browser was the Microsoft Internet Explorer, `INSERT PublicKeyIE` would be included

**optfield**

Special substitution variable that should be placed in any certificate field name `INSERT` where the value may be supplied by the end user. It enables the field to be displayed as optional if desired

**Note:** Depending on where a substitution variable is used, it may not have a valid meaning, e.g., `base64cert` would be meaningless prior to the certificate being retrieved. The value of `[base64cert]` would be the empty string (aka NULL) in this case.



#### 4.2.4 User Interface

The end user Interface is browser based static HTML pages. These are produced by the REXX connector CGIs by reading the installation customized configuration file. The user is required to have a RACF user ID and password for authentication. This is similar in concept to the RACF Auto-Registration (RAR) samples. The User's (or application's) access to new and existing RACF FACILITY class profiles determine if the user is authorized to generate and retrieve certificates through this interface. Installation customizable certificate templates contained within the configuration file, control which fields are user customizable via the HTML dialogues.

Figure 83 shows the first screen in a flow to request a digital certificate.

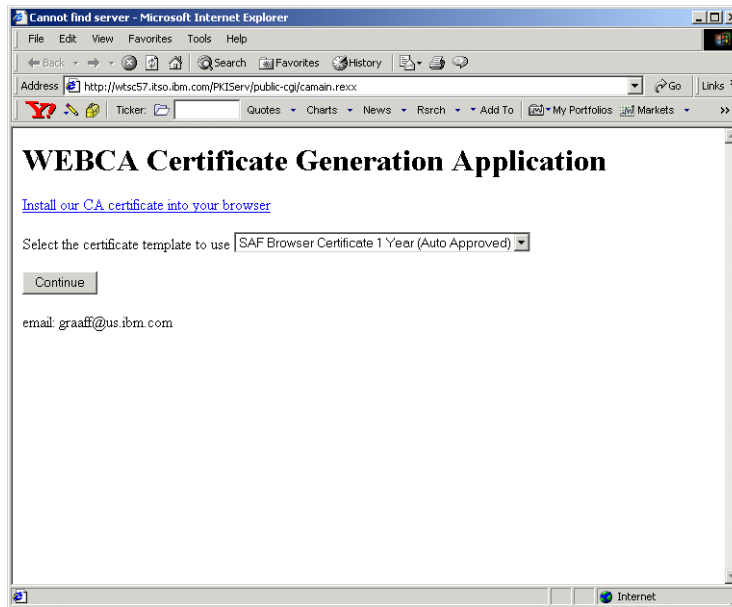


Figure 83. WEBCA certificate generation application page

This is the main page where the user can select a certificate template (model). The REXX CGI (`CAMAIN.REXX`) produces this page by reading the configuration file and answering everything in the `<CONTENT>` section under `<APPLICATION NAME=WEBCA>`. This page is not SSL protected but subsequent pages in this flow are. The Hypertext link “downloads” and installs OS/390 Root CA certificate into the user's web browser.

There are two selection choices to make on this page, to request a digital certificate:

1. SAF Browser Certificate 1 Year (Auto Approved)

This option is intended to request a digital certificate for a end user/client. The certificate has a validity period of one year (365 days) and is automatically approved.

2. SAF Server Certificate 1 Year (Auto Approved)

This option is intended to request a digital certificate for a server, like the HTTP, LDAP, TN3270 or other server. Again the certificate has a validity period of one year (365 days) and is automatically approved.

**Note:** These are the options that come as samples, but can be customized to suit the installations needs, as we discuss later.

Pressing the **Continue** button moves use to the next dialogue in the sequence, as shown Figure 84.

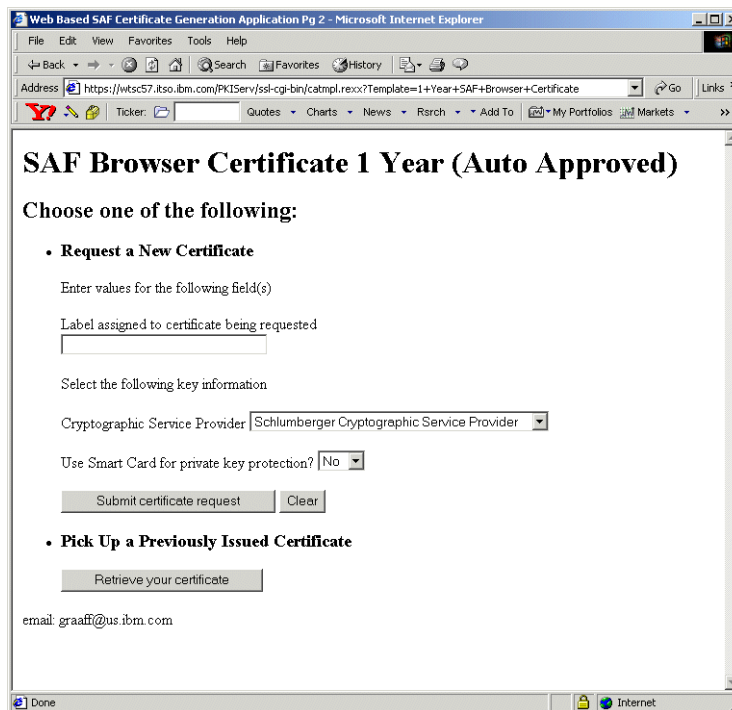


Figure 84. WEBCA certificate generation application - browser certificate page (1)

Once a template is selected, the user then enters the information permitted for that template (in this case the certificate label and key size). The REXX CGI (`CATMPL.REXX`) produces this page (Figure 84 on page 76) by reading the configuration file and answering everything in the `<CONTENT>` section under the `<TEMPLATE>` selected by the user. The page is SSL protected.

There are only two fields generated on this page that require a selection:

1. `%%Label%%`

This field indicates to specify of the label associated with the RACF certificate being generated.

2. `%%PublicKey browsertype%%`

This field generates a so called `INSERT` based on the browser used. In this case we use Internet Explorer and select the `INSERT PublicKeyIE`. This provides us with a selection of the Cryptographic Service Providers (CSPs) Internet Explorer supports to generate a public/private keypair.

**Note:** Depending on the operating system you are using and the version of the browser, various CSPs may or may not show.

In our example we are using Windows 2000 Professional Edition and Microsoft's Internet Explorer 5.5 with a cipher strength of 128 bit.

Again the page can be customized to suit your needs and allow for additional fields used in the certificate to be specified, like:

- Common Name (CN)
- Organization (O)
- Organizational Unit (OU)

This page can also be used to retrieve a certificate from a request that was submitted earlier.

If any required fields are not filled in, the CGI would produce an error.

When the **Continue** button is pressed, the user would get prompted to enter a valid user ID/password, as shown in Figure 85 on page 78.



Figure 85. RACF User ID and password prompt window

To be able to request a digital certificate a User ID requires authorization to various RACF FACILITY class profiles. These profiles are discussed in section 4.2.5.3, “Determine RACF access to the RACDCERT services” on page 85.

Next you enter your User ID and password and if you are authorized to personal certificate services, the next page is displayed, as shown in Figure 86.

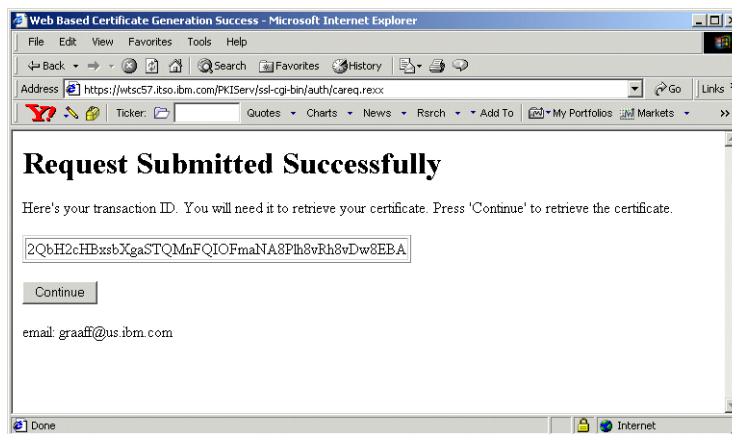


Figure 86. Certificate request submitted successful page

The user provided fields are gathered along with fields specified in the <APPL> and <CONSTANT> sections under the <TEMPLATE> selected by the user. All field values become parameters and the request is submitted to RACF (IRRSPXGL is called). If successful, a *transaction ID* (aka Certificate ID) is returned. The *Certificate ID* uniquely identifies the newly created certificate in RACF. The

REXX CGI (CAREQ.REXX) produces this page by reading the configuration file and answering everything in the <SUCCESSCONTENT> section under the <TEMPLATE> selected by the user. The digital certificate has now been generated and can be picked up using the so called “transaction ID”. If we take a look at RACF, we can see the certificate as well off course using a RACDCERTLIST command, as shown in Figure 87.

```

racdcert list(label('Paul IE cert 09/24/2000'))

Digital certificate information for user GRAAFF:

Label: Paul IE cert 09/24/2000
Certificate ID: 2QbH2cHBxsbXgaSTQmFQIOFmaN8Plh8vRh8vDw8EBA
Status: TRUST
Start Date: 2000/09/24 00:00:00
End Date: 2001/09/24 23:59:59
Serial Number:
    >02<
Issuer's Name:
    >CN=IBM-ITSO PDG CA.OU=ITSO.O=IBM.L=Poughkeepsie.SP=New York.C=US<
Subject's Name:
    >OU=ITSO.O=IBM.C=US<
Key Usage: HANDSHAKE
Private Key Type: None
Ring Associations:
*** No rings associated ***

```

Figure 87. RACDCERT LIST output from the generated certificate using the WEBCA application

As you can see from the RACDCERT LIST output, it shows the LABEL name we requested, the *transaction ID* (Certificate ID) and it indicates that there is no private key.

**Note:** The private key is at the workstation, not on the host (RACF) side.

To complete the digital certificate request process, we need to retrieve the certificate from RACF. To do this, we press the **Submit** button, to progress to the next page.

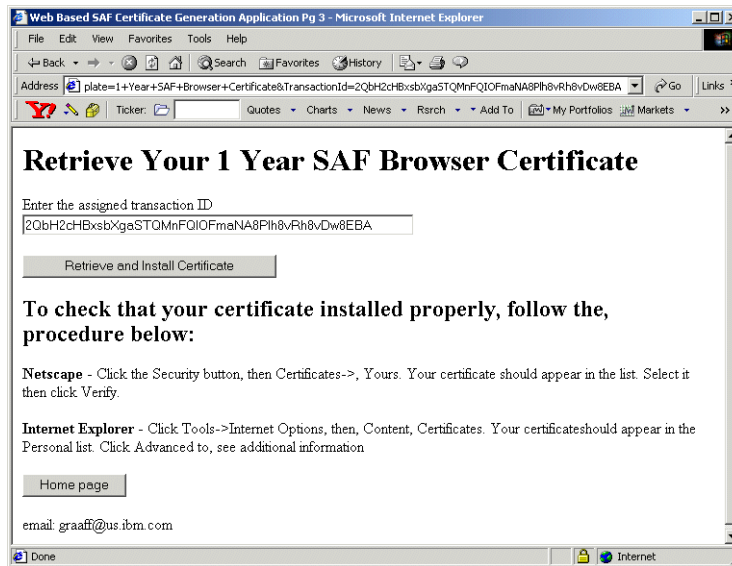


Figure 88. Retrieve browser certificate page

Pressing the **Submit** button presents the user with the page, as shown in Figure 88, to retrieve the certificate. The REXX CGI (`CARETRIEVE.REXX`) produces this page by reading the configuration file and answering everything in the `<RETRIEVECERT>` section under the `<TEMPLATE>` selected by the user.

Figure 88 also explains how you can check that the certificate is indeed properly installed.

To retrieve the certificate, we press the **Retrieve and Install Certificate** button. Figure 89 on page 81 shows the install page for Internet Explorer.

If the certificate being retrieved is a browser certificate, pressing the submit button will have the certificate downloaded directly into the browser certificate store. If the certificate being retrieved is a server certificate, the user will be presented with a page that enables the user to cut and paste the certificate to a file. The REXX CGI (`CAGETCERT.REXX`) produces this page by reading the configuration file and answering everything in the `<RETURNCERT>` section under the `<TEMPLATE>` selected by the user.

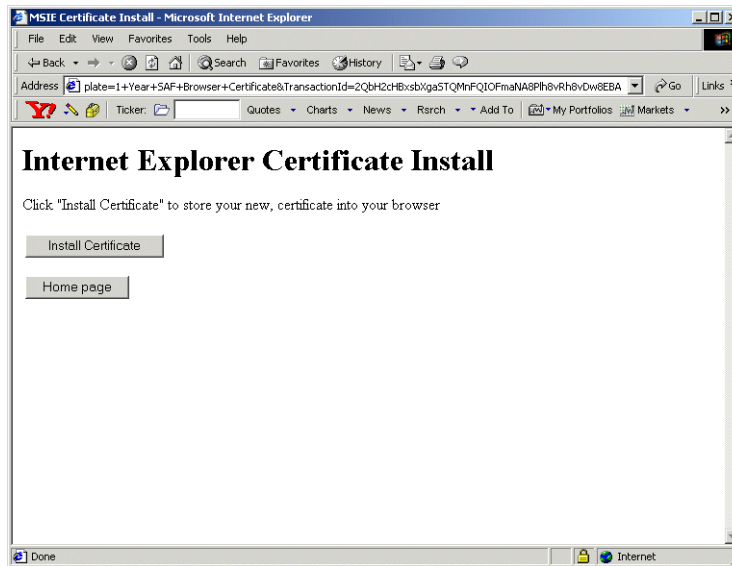


Figure 89. Internet Explorer certificate install page

Internet Explorer requires one more step than Netscape's Navigator. So again we press the **Continue** button, to install the certificate into the browser (IE). We receive a confirmation message that the install was successful as shown in Figure 90.

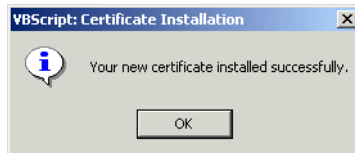


Figure 90. Certificate Installation successful message window

To check whether you have indeed successfully installed your certificate, for Internet Explorer you have to take the following steps:

1. select Tools from the main menu
2. select Internet options from the pull down list
3. select the content tab and then select certificates, as shown in Figure 91 on page 82

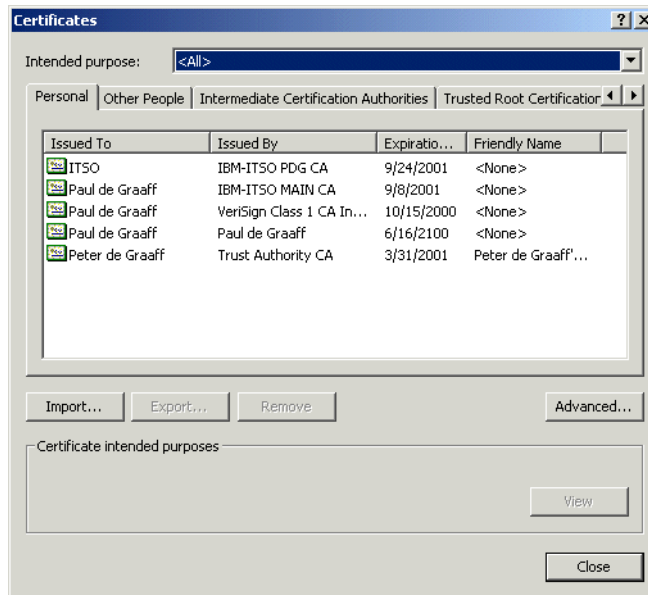


Figure 91. Internet Explorer Certificates Window

When you select your certificate, by double clicking on it, you can see the details about your certificate as shown in Figure 92 on page 83.



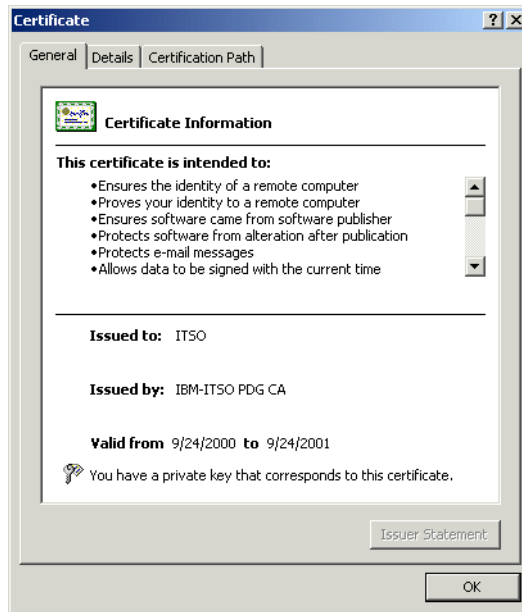


Figure 92. Certificate details window

You can see that the certificate was signed by the Certificate Authority (CA) IBM-ITSO PDG CA and issued to ITSO. Now where did he get that information from? We did not specify any of that information when we submitted our request for a certificate. The configuration file `PKI.CONF` holds the key to that information. Next we discuss the actual installation of the application and the configuration of `PKI.CONF` to make it better suit your needs.

#### 4.2.5 Installation and configuration

This section details the steps needed to install the Web based digital certificate application. In logical order the following steps are needed:

1. SMP/E Install the PTFs related to RACF APAR 45211 and SAF APAR 45212

See 4.2.5.1, "SMP/E install the PTF" on page 84.

2. Determine and setup the RACF access control needed to protect the `R_PKIServ` callable service, through profiles in the RACF FACILITY class.

See 4.2.5.2, "Determine RACF access to the `R_PKIServ` callable service" on page 84.

3. Determine and setup the RACF access control needed to protect the `RACDCERT` services, through profiles in the RACF FACILITY class.

See 4.2.5.3, "Determine RACF access to the `RACDCERT` services" on page 85.

4. Create your Certificate Authority certificate and make it available to your end users.

See 4.2.5.4, "Create your Certificate Authority (CA) Certificate" on page 86.

5. Determine the server root directory for the web application and install the sample REXX scripts and `PKI.CONF` configuration file

See 4.2.5.5, "Installing the sample code" on page 86

6. Configure the IBM HTTP Server (IHS) for OS/390 to use SSL

See 4.2.5.6, "Configure the IBM HTTP Server for SSL mode" on page 87

7. Add the directives necessary to the `HTTPD.CONF` file to enable the WEBCA application

See 4.2.5.7, "Other changes to make to the `HTTPD.CONF` file" on page 91

8. Customize the sample `PKI.CONF` configuration file and/or REXX scripts as needed for your organization's use

See 4.2.5.8, "Customizing the `PKI.CONF` file" on page 92

#### 4.2.5.1 SMP/E install the PTF

Depending on your maintenance level of OS/390 Version 2 Release 10, you may or may not need to install the PTF's related to APAR 45211 and 45212.

#### 4.2.5.2 Determine RACF access to the `R_PKIServ` callable service

The `R_PKIServ` callable service is protected by new profiles in the RACF FACILITY class called `IRR.RPKISERV.<function>`.

The function we are securing is called `GENCERT` and `EXPORT` like a `RACDCERT GENCERT`. The authority given to a User ID controls subsequent `RACDCERT` access checks, as follows:

**NONE** Access to the callable service is denied.

This is also true if no profile is defined (RC4).

**READ** Access is permitted based on subsequent `RACDCERT` access checks against the client's (end user's) User ID.

**UPDATE** Access is permitted based on subsequent `RACDCERT` access checks against the daemon's User ID.

**Note:** This does not apply to the Web Server daemon. This is intended for future use.

**CONTROL** Access is permitted with no subsequent `RACDCERT` access checks made.

If your User ID is RACF SPECIAL then no subsequent checks are made as well.

The subsequent access control check are made against the RACF FACILITY class profile `IRR.DIGTCERT.<function>`, as discussed next.

#### 4.2.5.3 Determine RACF access to the `RACDCERT` services

To be able to generate a digital certificate with the RACF `RACDCERT` command, you need RACF access to the following profiles in the RACF FACILITY class:

1. `IRR.DIGTCERT.ADD`
2. `IRR.DIGTCERT.GENCERT`

The WEBCA application uses a Certificate Authority Certificate to sign all digital certificates indicated by the `%%SignWith%%` field in the configuration file `PKI.CONF`. The Certificate Authority (CA) certificate is likely to be owned by a User ID that is not used for signon purposes. So to allow somebody to sign his/her certificate with that CA certificate the following authorities are needed for the WEBCA application to work:

1. `READ` access to the `IRR.DIGTCERT.ADD` profile in the RACF FACILITY class;
2. `CONTROL` access to the `IRR.DIGTCERT.GENCERT` profile in the RACF FACILITY class;

As an alternative you can choose to use a surrogate User ID, like WEBCA to be authorized to this service, because you do not want everybody to have `CONTROL` access to `IRR.DIGTCERT.GENCERT` FACILITY profile. User ID WEBCA requires the following authorities:

1. `UPDATE` access to the `IRR.DIGTCERT.ADD` profile in the RACF FACILITY class;
2. `CONTROL` access to the `IRR.DIGTCERT.GENCERT` profile in the RACF FACILITY class;

Depending on your security requirements and policy, access to these profiles can be allowed to all User IDs defined on the system.

**Note:** A RACF User ID with the SPECIAL attribute is exempt from any access control checks to these profiles.

#### 4.2.5.4 Create your Certificate Authority (CA) Certificate

In order to create and sign digital certificates for others you need to define a Certificate Authority certificate and associated private key. This is done using the RACF `RACDCERT GENCERT` command. Before issuing the command, decide what the Certificate Authority's distinguished name will be. Typically, Certificate Authorities have distinguished names in the following form:

```
OU=<your-CA's-friendly-name>.O=<your-organization>.C=<your-2-letter-country-abbreviation>
```

The `RACDCERT GENCERT` command to create our CERTAUTH certificate for use in our examples is :

```
RACDCERT GENCERT CERTAUTH SUBJECTSDN(DN('IBM-ITSO PDG CA') OU('ITSO')
O('IBM') C('US')) NOTBEFORE(DATE(2000-09-25)) NOTAFTER(DATE(2005-09-24))
WITHLABEL('ITSO PDG CA'))
```

**Note:** Set the validity date of the CA certificate to more years than what you will use for your personal certificates, to avoid date inconsistencies.

For more information on the `RACDCERT` command, defining profiles, and granting access see the *SecureWay Security Server for OS/390 (RACF) Security Administrator's Guide, SC28-1915* and *SecureWay Security Server for OS/390 (RACF) Command Language Reference, SC28-1919*.

#### 4.2.5.5 Installing the sample code

We assume the IBM HTTP Server is installed and functional for at least normal non-SSL mode before going on to this step. The default location for the server root is:

```
/usr/lpp/internet/server_root
```

Depending where you want to install the WEBCA application, the root might be:

```
/usr/lpp/internet/server_root/PKIServ
```

Create the directory structure for PKIServ as defined in 4.2.2, "Directory structure provided" on page 69. Obtain the sample TAR file from the RACF website:

```
http://www.s390.ibm.com/products/racf/webca.html
```

Next do a file transfer in binary mode using FTP to OS/390. Then untar the file using the following command:

```
cd /usr/lpp/internet/server_root
tar -xvf webca.tar
```

Remember to set the permission bits on all files. The permission bits for the executables can be the octal value of "644". Permissions bits can be changed using the `CHMOD` command, as shown:

```
CHMOD 644 CAMAIN.REXX
```

The permissions bits for the configuration file `PKI.CONF` and the Certificate Authority (CA) certificate `CACERT.DER` can be set according to your installation's security requirement.

#### 4.2.5.6 Configure the IBM HTTP Server for SSL mode

The WEBCA application requires that the IBM HTTP Server operate in both normal and SSL mode. To be able to use SSL, your server will need to obtain a digital certificate. You can choose to purchase one from an external Certificate Authority (e.g. Verisign) or create one using RACF.

**Note:** If your server is already operating in SSL mode you can skip the next "Obtain a Certificate Using RACF" step.

Depending on your level of the IBM HTTP Server for OS/390, the utility you create a server certificate with is different. Prior to IHS Release 5.3, you use `IKEYMAN`, with IHS Release 5.3 you may use `GSKKYPAN` or a RACF keyring using the `RACDCERT ADDRING` command.

#### *Obtaining a Certificate for your Web Server*

The procedure we describe next applies to the IBM HTTP Server (IHS) Release 5.3 that ships with OS/390 Version 2 Release 10. To be able to use SSL, your server needs a digital certificate to identify the server, like we did earlier when we created a digital certificate for you (an individual). IHS allows you to choose between two options:

1. create a digital certificate using a utility called `GSKKYPAN`, or
2. create a digital certificate using RACF, using the `RACDCERT GENCERT` command

Next the digital certificate of your server needs to be installed in a so called key ring, that is used by your server. The `GSKKYPAN` utility creates a key database or key ring for you. The RACF `RACDCERT ADDRING` and `RACDCERT`

CONNECT commands can do similar functions. The difference between the two keyrings is the location of the key ring and the security of the key ring. The key ring created by GSKKYMAN is a UNIX file and the security relies on permission bits and a password to access the file. The RACF key ring as the name implies is stored in the RACF database, and requires RACF authorizations to be read and updated.

You need to decide who (what Certificate Authority Certificate) is signing the server's certificate, as we see later in this section. You may use the same CA certificate that signs the browser's (user) digital certificates.

### **Using GSKKYMAN to obtain a Server Certificate**

1. Determine the location for your key database (key ring) and change to that directory, for example CD to /usr/lpp/internet/etc and invoke  
/usr/lpp/gskssl/bin/gskkyman
2. Choose option **1** to create a key database. Type in a name or let it default to key.kdb and enter the desired password. When asked to "work with the database now?" Enter **1** for yes.
3. Choose **3** - Create new key pair and certificate request. Answer the prompts for file name, label, key size (1024 recommended), and subject name fields. (Note, Common Name should be your server's symbolic IP address (e.g., www.ibm.com)). Exit GSKKYMAN when done.

4. From TSO, OGET the certificate request file to an MVS dataset, e.g.,

```
OGET certreq.arm '/usr/lpp/internet/etc/certreq.arm'
```

5. Use the RACF RACDCERT command to read the request and generate the server certificate, e.g.,

```
RACDCERT GENCERT(certreq.arm) ID(websrv) SIGNWITH(CERTAUTH LABEL('ITSO  
PDG CA')) WITHLABEL('SSL Cert')
```

6. Export both the new server certificate and the Certificate Authority certificate to MVS datasets then OPUT these to HFS files, e.g.,

```
RACDCERT EXPORT(LABEL('SSL Cert')) ID(websrv) DSN(cert.arm)  
FORMAT(CERTB64)
```

```
OPUT cert.arm '/usr/lpp/internet/etc/cert.arm'
```

```
RACDCERT EXPORT(LABEL('ITSO PDG CA')) CERTAUTH DSN(cacert.der)
```

```
FORMAT(CERTDER)
```

```
OPUT cacert.der `/usr/lpp/internet/etc/cacert.der' BINARY
```

7. CD to `/usr/lpp/internet/etc` and invoke `/usr/lpp/gskssl/bin/gskkyman` again
8. Choose option **2** to open the key database you created before. Reply to the name and password prompts.
9. Choose option **6** to Store a CA certificate and specify the `/usr/lpp/internet/etc/cacert.der` file. When asked to "exit ikeyman?" Enter **0** for No.
10. Choose option **4** to Receive a certificate issued for your request and specify the `/usr/lpp/internet/etc/cert.arm` file. Again enter **0** when asked to "exit ikeyman?"
11. Choose option **11** to Store encrypted database password

### Using RACF to obtain a Server Certificate

The following steps detail the setup of a RACF key ring and a server certificate for your Web server:

1. The User ID associated with the IHS started task needs to be authorized to the following RACF FACILITY class profiles:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(websrv) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(websrv) ACCESS(CONTROL)
```

2. Next we need to set up the RACF key ring for use by the Web server:

```
RACDCERT ID(websrv) ADDRING(WEB57)
```

**Note:** You need RACF SPECIAL to be able to define the RACF key ring.

3. You generate a server certificate for your Web server, using the RACF RACDCERT command, as shown:

```
RACDCERT ID(websrv) GENCERT SUBJECTSDN(CN('WTSC57.ITSO.IBM.COM') O('IBM')
OU('ITSO') L('Poughkeepsie') C('US') SP('New York')) SIZE(1024)
WITHLABEL('WEB57CERT') SIGNWITH(CERTAUTH LABEL('ITSO PDG CA'))
```

4. You then need to connect the server certificate and the Certificate Authority certificate to the keyring we created in step 2, using the RACF RACDCERT CONNECT command, as shown:

```
RACDCERT ID(websrv) CONNECT(ID(websrv) LABEL('WEB57CERT') RING(WEB57)
DEFAULT USAGE(PERSONAL))
RACDCERT ID(websrv) CONNECT(CERTAUTH LABEL('ITSO PDG CA') RING(WEB57)
USAGE(CERTAUTH))
```

5. Configure your Webserver to use the RACF keyring. (APAR PQ39311 provides the function to do this)

### Turning on SSL mode

To turn on SSL in your Web server, you need to make the following changes in your configuration file HTTPD.CONF:

1. set SSLMODE ON
2. designate a port for SSL usage, like SSLPORT 443
3. set the parameter KEYFILE to indicate either the UNIX file name for the key ring to be used like. KEYFILE /usr/lpp/internet/etc/key.kdb or indicate the label name of the RACF keyring to be used.



#### 4.2.5.7 Other changes to make to the HTTPD.CONF file

The following directives are to be added to the HTTPD.CONF file to be able to use the WEBCA application:

```

Protection PublicUser { 1
    ServerId      PublicUser
    UserID       PUBLIC
    Mask         Anyone
}

Protect /PKIServ/public-cgi/*    PublicUser
Protect /PKIServ/ssl-cgi-bin/*   PublicUser
Protect /PKIServ/*               PublicUser

Protection AuthenticatedUser { 2
    ServerId      AuthenticatedUser
    AuthType     Basic
    PasswdFile   %%SAF%%
    UserID       %%CLIENT%%
    Mask         All
}

Protection SurrogateUser { 3
    ServerId     SurrogateUser
    AuthType     Basic
    PasswdFile   %%SAF%%
    UserID       WEBCA
    Mask         All
}

Protect /PKIServ/ssl-cgi-bin/auth/*      AuthenticatedUser
Protect /PKIServ/ssl-cgi-bin/auth/careq.rexx SurrogateUser

Redirect /PKIServ/ssl-cgi/*      https://<server-domain-name>/PKIServ/ssl-cgi-bin/*
Exec    /PKIServ/public-cgi/*    <server-root>/PKIServ/public-cgi/*
Exec    /PKIServ/ssl-cgi/*       <server-root>/PKIServ/ssl-cgi-bin/*
Exec    /PKIServ/ssl-cgi-bin/*   <server-root>/PKIServ/ssl-cgi-bin/*
Pass    /PKIServ/*               <server-root>/PKIServ/*

AddType .cer    application/x-x509-user-cert    ebodic 0.5 # Browser Certificate
AddType .der    application/x-x509-ca-cert      binary 1.0 # CA Certificate

```

Figure 93. HTTPD.CONF file required changes

The setup requires three protection setups:

- PublicUser (1)** used for the “public” resources
- AuthenticatedUser (2)** used to determine the client requesting the certificate.
- SurrogatUser (3)** used for the actual generation of the certificate, because we do not want every user to be authorized with CONTROL authority to the IRR.DIGTCERT.GENCERT profile in the RACF FACILITY class.

**Note:** if you choose not to use a surrogat User ID to issue certificates, then replace *SurrogatUser* with the *AuthenticatedUser*.

The PROTECT, PASS and EXEC directives are their to insure the proper protection and execution of the WEBCA application.

There is one REDIRECT directive to insure that SSL is invoked for the WEBCA application.

#### 4.2.5.8 Customizing the PKI.CONF file

Depending on your installation’s needs and security requirements, you may decide to change your configuration file `PKI.CONF`. One change that is definitely required is to indicate the label of the Certificate Authority Certificate that is going to sign the certificates requested.

In the <constant> section of the each <template>, the `%%SignWith=%%` parameter indicates the label of the CERTAUTH certificate that signs the certificate issued with the WEBCA application. In our example we use the CERTAUTH certificate generated earlier, like `%%SignWith=SAF:CERTAUTH/ITSO PDG CA%%`.

Another area where you may want to change things is the information necessary to build the browser’s certificate. The only two fields displayed today are the label and public key type (CSP) on the certificate request page, as shown in Figure 84 on page 76. The following fields can be added to the request page to provide us with more meaningfull information:

**CommonName** the name of the individual, like Paul de Graaff

**Note:** If you leave the `%%CommonName%%` field in the constant section, it uses the `NAME` field from your RACF User profile as the Common Name.

**OrgUnit** the organizational unit the individual resides, like ITSO

Org	the organization the individual works for, like IBM
Locality	the locality, like Poughkeepsie
StateProv	the State or Province you live in, like New York
Country	the country you live in, abbreviated to two characters, like US
AltEmail	your email address, like graaff@us.ibm.com
AltDomain	your internet domain, like www.ibm.com
AltURI	your universal resource identifier, like wtsc57.itso.ibm.com
AltIPAddr	numeric IP address, like 9.12.14.247

Depending on your security requirements, you may have the user enter his/her, CommonName, Org, OrgUnit and Locality. To make these changes, you need to edit the `PKI.CONF` file.

Figure 94 on page 93 shows part of the `PKI.CONF` file, before we made the changes to include the new fields on the certificate request page.

```

<TEMPLATE NAME=1 Year SAF Browser Certificate>
.....
<p> Enter values for the following field(s)
%%Label%%
%%PublicKey%browsertype"%%
.....
<CONSTANT>
%%Org=IBM%%
%%OrgUnit=ITSO%%
%%KeyUsage=handshake%%
%%NotAfter=365%%
%%Country=US%%
%%SignWith=SAF:CERTAUTH/ITSO PDG CA%%
</CONSTANT>

```

Figure 94. `PKI.CONF` file before changes

To make the changes, you need to move the fields you want to include in the certificate request page from the `<constant>` section to the line under "Enter values for the following Field(s)" and remove the value specified, as shown in

```

<TEMPLATE NAME=1 Year SAF Browser Certificate>
.....
<p> Enter values for the following field(s)
%%Label%%
%%CommonName%%
%%Org%%
%%OrgUnit%%
%%PublicKeyYbrowsertype"%%
<CONSTANT>
%%KeyUsage=handshake%%
%%NotAfter=365%%
%%Country=US%%
%%SignWith=SAF:CERTAUTH/ITSO PDG CA%%
</CONSTANT>

```

Figure 95. PKI.CONF file after changes

When you have made your changes, save them and invoke the WEBCA application again. When you make the request for a browser certificate, you will see the following screen, including the new fields added, as shown in Figure 96 on page 94.

Figure 96. Certificate request page after PKI.CONF updates

## 4.2.6 Program Control Enhancements

This section describes the enhancements made to Program Access to Data Sets (PADS). UNIX System Services need RACF Program Control support to provide good security and integrity for UNIX servers and daemons. RACF will also produce new diagnostic messages.

## 4.2.7 Overview

One of the continuing objectives of OS/390 is to encourage server consolidation, enabling more applications to run on OS/390. As this occurs, however, and especially as customers and vendors create new server applications (or port existing ones) they may experience difficulty configuring program control using the RACF PROGRAM class for traditional MVS libraries and the program controlled extended attribute for OS/390 UNIX files. Program controlled is required if servers are to run properly with good security and integrity.

## 4.2.8 Introduction

The support provided by Program Control Enhancements will make it easier for a customer to determine which programs, either in traditional MVS load libraries or in the OS/390 UNIX hierarchical file system (HFS), they need to define as controlled programs. Plus it will help preserve the security and integrity afforded to servers and daemons while they *run*, prior to Version 2 Release 10 this was used at start up time only, by preventing the introduction of uncontrolled programs into the execution of the server or daemon. This can prevent compromise of security or integrity of the server or daemon.

By protecting the `BPX.DAEMON` and `BPX.SERVER` resources in the FACILITY class, with fewer difficulties in defining programs to RACF and the HFS, customers will also get good security and integrity for their servers and daemon running under OS/390 UNIX. RACF and/or OS/390 UNIX will provide messages to aid in the definition of the appropriate programs

Support will also with the definition of programs for use with RACF PADS and execute controlled processing. This will occur through the improved messages that will result when execution time errors occur, and through the ability of the `RACROUTE REQUEST=AUTH` post processing exit (ICHRCX02) to determine that a PADS request failed because the environment was not controlled.

### 4.2.9 Description

RACF will provide a new service, IRRENS00. A bit in the RACF Communication Vector Table(RCVT) indicates that the service is available. The address of the service is located by the RCVT, as it is for several other RACF services. The new service will provide the following functions:

Mark the environment uncontrolled (dirty)

- The mark-uncontrolled function first checks that the environment is allowed to become uncontrolled (dirty). It may become uncontrolled if it is already uncontrolled or if the environment is currently controlled (clean), it can become uncontrolled if keep-controlled is not in effect. If the environment was controlled when the service was called, it is marked uncontrolled, and a message provided by the caller is saved describing the reason it became uncontrolled.
- If the environment is controlled, and cannot become uncontrolled (keep-controlled is in effect), then IRRENS00 displays any previously saved message via WTO to help the administrator correct the problem. The message was saved when a caller requested that the environment be kept controlled, and provided a message on that request.

Keep environment controlled (clean).

- The keep-controlled function marks the current environment as keep-controlled (keep-clean) if it is currently controlled, and saves a message provided by the caller describing the reason the environment must remain controlled. If the environment is already uncontrolled (dirty), the request fails and IRRENS00 will issue any previously saved message via WTO to help the administrator correct the problem. This message was saved when a caller requested when the environment be marked uncontrolled. If there is no saved message, a general message is issued.

Reset keep-controlled

- This function turns off the specified keep-controlled indicators and clears the related saved message when an authorized caller knows it is safe to do so. In general, OS/390 UNIX turns off the OS/390 UNIX keep-controlled indicator, and RACF turns off the RACF keep-controlled indicator. Support is provided for resetting both.

In addition to the messages saved by IRRENS00, RACF will provide messages to aid diagnosis in some cases, specifically when it denies:

- Use of PADS for a dataset due to a dirty environment.

- Use of PADS due to the absence of the currently executing program in the conditional access list, or due to presence of other programs defined with PADCHK in the environment that are not found in the conditional access list of the dataset profile.
- Authority to load an uncontrolled module due to the presence of execute-controlled modules in the environment
- Authority to load any module to the existence of open PADS datasets in the environment.

These messages should make it easier for the security administrator to determine what PROGRAM definitions or WHEN(PROGRAM(...)) conditional access list entries need modification.

#### 4.2.10 RACF Services

RACF will provide a new service IRRENS00 (Environment Service) located via the RCVT. IRRENS00 provides the following functions:

- Keep-Controlled, this function will verify that the environment is currently controlled, and if so will set flags indicating that it must remain controlled. It will also save a message supplied by the caller indicating the reason it must remain controlled. It will keep separate flags for UNIX System Services requests and RACF requests.
- Mark-Uncontrolled, this function will determine whether the environment must remain controlled by inspecting the keep-controlled indicators and if necessary the PADS dataset list from the Accessor Environment Element (ACEE) and the CDEs for modules in the address space, and if not will mark it uncontrolled and will also mark any existing TCBs and CDEs as uncontrolled. However if a keep-controlled request is outstanding, mark-uncontrolled will return an error code and either issue a WTO any saved message from a previous keep-controlled request or will generate messages to indicate what RACF needed the environment kept controlled. For a successful request, mark-uncontrolled will also save a caller provided message indicating why the environment became uncontrolled, which it can issue on subsequent keep-controlled requests if necessary.
- Reset Keep-Controlled, this function will reset the keep-controlled indicators and remove any saved messages relating to previous keep-controlled requests. It is intended only for use by OS/390 Unix System Services when they have determined that the environment no longer needs to be kept controlled.

OS/390 UNIX System Services should use IRRENS00 to tell RACF to keep the environment clean, to mark the environment dirty, and to reset the keep-controlled indicator.

If you have set up conditional access lists specifying WHEN(PROGRAM(...)) for DATASET profiles will receive additional error messages for failed attempts to access datasets. If these conditional access lists are not being used to allow access, they should be deleted.

#### 4.2.11 Program Control Enhancements Example Errors

For documentation purposes we deliberately removed a dataset from the RACF PROGRAM class, so we could demonstrate and show what error messages you would receive in your own installation.

We decided to remove dataset CEE.SCEERUN and stopped and re-started our LDAP and HTTP servers.

By using the RLIST command we can find and display the profile within the RACF PROGRAM class that contains the CEE.SCEERUN dataset profile.

```
RLIST PROGRAM *
```

From our RLIST command we found the dataset CEE.SCEERUN was in the RACF PROGRAM profile \*. See Figure 97

```
CLASS      NAME
-----
PROGRAM    *

MEMBER CLASS NAME
-----
EMBR

DATA SET NAME                VOLSER  PADS CHECKING
-----
CEE.SCEERUN                  NO
CSF.SCSFMODE0                NO
CSF.SCSFMODE1                NO
DSN510.SDSNLOAD              NO
DSN610.SDSNLOAD              YES
```

Figure 97. Output from the RLIST command

We then proceed to remove the dataset from the \* profile in the RACF class PROGRAM, for the test purpose.



```
RALTER PROGRAM * DELMEM( 'CEE.SCEERUN' )
```

Before we can test for any errors by stopping and starting LDAP and HTTP servers we must do a PROGRAM refresh:

```
SETR WHEN(PROGRAM) REFRESH
```

Now we can stop and restart the LDAP and HTTP servers. From the start up of the LDAP server the following message as shown in Figure 98 on page 99 is displayed, informing us that the program EDC\$EUEY in dataset CEE.SCEENRUN caused the environment to become uncontrolled (dirty) even though the LDAP server was up and running.

```
ICH420I PROGRAM EDC$EUEY FROM LIBRARY CEE.SCEERUN CAUSED THE  
ENVIRONMENT TO BECOME UNCONTROLLED.  
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR SERVER (BPX.SERVER)  
PROCESSING.  
ICH420I PROGRAM EDC$EUEY FROM LIBRARY CEE.SCEERUN CAUSED THE  
ENVIRONMENT TO BECOME UNCONTROLLED.  
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR SERVER (BPX.SERVER)  
PROCESSING.  
ICH420I PROGRAM EDC$EUEY FROM LIBRARY CEE.SCEERUN CAUSED THE  
ENVIRONMENT TO BECOME UNCONTROLLED.  
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR SERVER (BPX.SERVER)  
PROCESSING.
```

Figure 98. RACF error message from restart of the LDAP server

With the HTTP server we had no error messages displayed at startup or in the log, so at this point we do not know whether the environment is an controlled (clean) or uncontrolled (dirty) state. Upon a further test by trying to use HTTP, to obtain a communication connection with a digital certificate through the browser a message is displayed, as shown in Figure 99 on page 100. We can also check the log on OS/390 and we will find the same error messages as shown in Figure 98 on page 99



Figure 99. Error message displayed from requesting digital certificate

**Note:** It is possible after making changes or defining new applications the application will start up, however the application could be in an uncontrolled (dirty) state. And until the application is used in such a way i.e. a password verification or some authorized/trusted/restrictive command is used no error messages will be displayed.

After these tests we added back the CEE.SCEERUN dataset to the \* profile in the class PROGRAM.

```
RALTER PROGRAM * ADDMEM('CEE.SCEERUN' //NOPADCHK)
```

We now need to the refresh so other applications can use the CEE.SCEERUN dataset without getting RACF violations or unauthorized program checks.

```
SETROPTS WHEN(PROGRAM) REFRESH
```

---

## 4.3 RVARY Enhancement

This section describes the enhancement made to the RVARY password processing command.

**Note:** This enhancement depends greatly on physical security on the Master Console, we recommend that the Master Console is always protected from unauthorized access. Only Operational personnel should have access to the Master Console, which should reside a secure controlled area.

### 4.3.1 Introduction

To improve the availability of RACF, the default password YES will be accepted by RVARY in certain scenarios. This is in addition to any password set by the installation with SETROPTS RVARYPW. This is limited to the RVARY functions normally needed in recovery situations:

- RVARY NODATASHARE
- RVARY SWITCH
- RVARY ACTIVE

If one of these RVARY functions is request either from someone who access to the RVARY command or as an operator who has physical access to the master console, an operator reply of YES, or the installation defined password will be accepted, but only from the master console. This allows faster recovery, i.e. RACF database errors or DASD failures etc. The YES password may be used in these critical situations:

- The person who knows the RVARY password is unavailable.
- The RVARY password has been forgotten and no administrator is available to reset it with the SETROPTS RVARYPW command, or the system is not in a state that allows to be reset.
- The system has unknowingly been IPL'd with the wrong RACF databases which has a different password than expected.
- An error is encountered in encrypting the password and comparing it to the installation defined password.

### 4.3.2 Logical Console Security

During a planned outage for a RACF database change it is possible for the RACF administrator to amend their own User ID or the driver of the change User ID so they have OPER and CONSOLE authority under the class

TSOAUTH, along with amendments to the OPERPARM segment within the User ID.

**Note:** While this makes the change easier to implement it is not recommend that any User ID has this authority on a permanent basis, as we stated earlier in this amendment to the RVARY command we recommend that physical security is in place.

With a few commands we can give a User ID OPERPARMs, OPER and CONSOLE.

1. First we give the User ID read access to profiles OPER and CONSOLE within the PROFILE TSOAUTH:

```
PERMIT OPER CLASS(TSOAUTH) ID(BRIGGS) ACCESS(READ)
PERMIT CONSOLE CLASS(TSOAUTH) ID(BRIGGS) ACCESS(READ)
```

2. We now need to refresh the RACLISTed PROFILE TSOAUTH for the changes made in step 1:

```
SETR RACLIST(TSOAUTH) REFRESH
```

3. Amend the User ID now to add parameters to the OPERPARM segment:

```
ALU BRIGGS OPERPARM(AUTH(MASTER) LEVEL(ALL) CMDSYS(*) MSCOPE(*ALL)
ALTGRP(MASTER))
```

This now enables the above User ID to enter OPER and CONSOLE commands via their own workstation. This authority while useful for the individual user opens up various security exposures to your environment.

#### 4.3.3 RVARY Console Samples

Now that we have a User ID with the OPERPARMS segment correctly defined we can now issue the CONSOLE and OPER commands via our workstation. In this section we will show the main uses of the RVARY command with the password being the word YES, using the locally defined RACF subsystem '#'.

1. By entering TSO CONSOLE from any ISPF screen we obtain Master Console authority. We used the RVARY LIST command to check the status of the RACF databases as shown in Figure 100:

```

CONSOLE
#rvary list
CONSOLE

IRRA011I (#) OUTPUT FROM RVARY:
ICHL5013I RACF DATABASE STATUS:
ACTIVE  USE      NUMBER  VOLUME  DATASET
-----  -
YES     PRIM      1      PDGCAT  SYS1.RACFESA
YES     BACK      1      PDGSYL  SYS1.RACF.BKUP1
ICHL5020I RVARY COMMAND HAS FINISHED PROCESSING.
CONSOLE

```

Figure 100. RVARY LIST command and output

- By issuing the RVARY SWITCH command we can switch from the RACF Primary database to the RACF Backup database. After entering the RVARY SWITCH command we need to issue the outstanding WTOs command, to find out the correct WTO associated with the RVARY SWITCH before we can reply to the SWITCH command as shown in Figure 101:

```

#rvary switch
CONSOLE
d r,1
CONSOLE

IEE112I 14.01.21 PENDING REQUESTS 558
RM=1  IM=0  CEM=0  EM=0  RU=0  IR=0  NOAMRF
ID:R/K  T MESSAGE TEXT
        008 R *008 ICH703A ENTER PASSWORD TO SWITCH RACF DATASETS
                JOB=RACF  USER=BRIGGS

CONSOLE
r 08,yes

```

Figure 101. RVARY SWITCH process

- We now need to activate the RACF Primary database, using the RVARY ACTIVE command and as we did in step 2 find the WTO and reply to the outstanding WTO number as shown in Figure 102 on page 104

```

#rvary active,dataset(sys1.racfesa)
CONSOLE
d r,1
CONSOLE

IEE112I 14.03.51 PENDING REQUESTS 661
RM=1 IM=0 CEM=0 EM=0 RU=0 IR=0 NOAMRF
ID:R/K T MESSAGE TEXT
009 R *009 ICH702A ENTER PASSWORD TO ACTIVATE RACF JOB=RACF
USER=BRIGGS

CONSOLE
r 09,yes
CONSOLE
IEE600I REPLY TO 009 IS:SUPPRESSED

IRRA011I (#) OUTPUT FROM RVARY:
ICHL5013I RACF DATABASE STATUS:
ACTIVE USE NUMBER VOLUME DATASET
-----
YES PRIM 1 PDGSYL SYS1.RACF.BKUP1
YES BACK 1 PDGCAT SYS1.RACFESA
ICHL5020I RVARY COMMAND HAS FINISHED PROCESSING.

```

Figure 102. RVARY ACTIVE command

**Note:** The RVARY command cannot be protected. This is intentional during recovery; RACF must not be allowed to attempt to access the database. The RVARY command is always protected by an operator prompt, regardless of whether it is entered from TSO or as an operator command.

#### 4.4 Authorized Program Analysis Reports (APARs)

This section describes changes from authorized program analysis reports (APARs) and other service updates that have also been incorporated into this release.

##### APAR OW39128 Error Description

- Customers need to audit the use of programs. When they use a program class profile that includes more than one library in the member list the auditing is not effective since the dataset name is not included in the SMF record. Without the dataset name the customer cannot determine which library the program was loaded from

##### Problem Conclusion

- If a successful or failed attempt to load a controlled program is audited, RACF builds SMF record type 80, EVENT 2 documenting the RESOURCE ACCESS attempt. The program name is contained in Relocate Section 1

(x'01') which is included in the SMF record. To further identify the controlled program, the library (the partitioned data set) and the volser will be added to the record. The partitioned data set name will be contained in a new relocate section, 66 (x'42'), and the volume serial will be contained in an existing relocate section, 15 (X'0F') that had not previously been included in the EVENT 2 record for this case. Changes will also be made to the SMF UNLOAD utility and SAMPLIB members IRRADUTB and IRRADULD to allow the viewing of this new information. The additional information makes program control much easier to audit.

#### APAR OW38799 Error Description

- The SMF records written by the SETROPTS command processor have no indication that the LIST operand was requested. The output in the SMF records is the same for SETROPTS (with no operands) as it is for SETROPTS LIST. RACF should indicate the LIST operand was specified on the SETROPTS command.

#### Problem Conclusion

- The SMF Type 80 record has been modified to include LIST among the options specified or ignored in the SETROPTS command. LIST-specified is indicated by a bit positioned immediately after the NOADDCREATOR-specified bit. Likewise, LIST-specified-but-ignored is indicated by a bit positioned immediately after the NOADDCREATOR-specified-but-ignored bit. Also, the SMF Data Unload Utility (IRRADU00) was modified such that if LIST had been specified in the SETROPTS command, it will also be included in the list of options in its corresponding formatted output record.

#### APAR OW42092 Error Description

- In some circumstances the ls -l and whoami commands may return UID numbers instead of UID names. Specifically, both of these functions rely on a getpwuid() call to provide a name and groupname for the associated UID. Currently getpwuid will translate the UID # to a UID name and then use that name to determine a group name. If the group name found does not have an associated GID number then getpwuid will return with a bad return code causing the UID number not to be translated in the output of the above commands. The intention of this APAR is change the getpwuid function to return the groupname from BPX.DEFAULT.USER in the above circumstance. This will make the function of getpwuid consistent with getpwnam. For example, a user that did a chown username filename will not be failed if the default group of username has no GID (provided BPX.DEFAULT.USER has a group with a valid GID) yet the corresponding

ls -l command will show the owner of the file as the UID number of username instead of username. This APAR would cause ls -l to show a username.

#### Problem Conclusion

- RACF will always audit when a default-UID-user gets dubbed. This will facilitate an installation's ability to monitor usage of default UID, and to prevent unintended usage of the default UID. Also, RACF will issue a warning message if a user is assigned an OMVS UID, but the user's default group does not have an OMVS GID.
- Two new RACF warning messages have been documented for an ADDUSER and ALTUSER command:
- ICH01019I - This is a warning message that gets issued if a user with an OMVS UID gets added and has a default group which does not have a GID. As shown in Figure 103.

```
ICH01019I USER UGO IS ASSIGNED AN OMVS UID,  
BUT DEFAULT GROUP CICSJB DOES NOT HAVE A GID.PROCESSING CONTINUES.
```

Figure 103. ICH01019I Warning Message

- ICH21035I - This is a warning message that gets issued if a user with an OMVS UID gets changed and has a default group which does not have a GID. As shown in

```
ICH21035I USER UGO IS ASSIGNED AN OMVS UID, BUT DEFAULT GROUP  
CICSJB DOES NOT HAVE A GID. PROCESSING CONTINUES.
```

Figure 104. ICH21035I Warning Message

- To fix either of these problems, the default group should be assigned a GID, or the UID should be removed from the user profile.