

# iSeries OS/400 Overview

*2001 Announcements  
ITSO Technical Overview  
May 2001*

**IBM** @server. For the next generation of e-business.

## Security

- Object Signing
- New Password Support
- Service Tools Security

## Integrated File System

## NetServer

## Miscellaneous Enhancements

## Release to Release Compatibility

Note: The term "AS/400" is used throughout this presentation to include both AS/400 and iSeries systems running V5R1 unless otherwise noted.

# Security and Integrity

**IBM @server.** For the next generation of e-business.

# Object Signing

**IBM @server.** For the next generation of e-business.

Purpose is to:

- Detect objects that have been tampered with after they have been signed
- Enforce integrity of objects
- Identify issuer of objects

Most cipher products are designed for stream files - OS/400 objects are more complicated than stream files

Requirement to bypass or enforce signatures to be present or to be valid upon restore

Interface for verifying object integrity within the system

Objects are more and more distributed via the Internet, quite often as stream files in the form of text files or executables. When the issuer of the signed material wants to help the recipient be sure that the material has not been tampered with, and is originated from the supplier, digital signatures are added to objects to verify the integrity and origin of the object.

Beginning in V5R1, iSeries 400 provides support for using certificates to digitally "sign" objects and to verify the digital signatures on objects. Digitally signing objects provides a way to ensure the integrity of the contents of an object as well as the source of an object's origin.

Object signing support augments traditional iSeries system tools for controlling who can change objects. Traditional controls cannot protect an object from unauthorized tampering while the object is in transit across the Internet or other untrusted network, or while the object is stored on a non-iSeries system. Using digital signatures on objects protects the objects from unauthorized change.

As of V5R1 all eligible OS/400 objects are signed. However, the certificates used for signing and verification of these OS/400 objects are not externalized.

# Signing an Object

Certificate is used to create a signature

Object itself is not encrypted

Certificate Authority needs to be trusted

Eligible objects:

- Save Files (\*SAVF)
- Programs (\*PGM), Service Programs (\*SRVPGM), SQL Packages (\*SQLPKG), Modules (\*MODULE), Java Programs

Only V5R1 supports object, including stream file, signing

Placing a digital signature on an object consists of using a certificate's private key to add an encrypted mathematical summary of the data in an object. The signature protects the data from unauthorized changes. The object and its contents are not encrypted and made private by the digital signature; however, the summary itself is encrypted to prevent unauthorized changes to it. Anyone who wants to ensure that the object has not been changed in transit and that the object originated from an accepted, legitimate source can use the signing certificate's public key to verify the original digital signature. If the signature no longer matches, the data may have been altered. In such a case, you can avoid using the object and can instead contact the signer to obtain another copy of the signed object.

If you decide that using digital signatures fits your security needs and policies, you should evaluate whether you should use public certificates versus issuing private certificates. If you intend to distribute objects to users in the general public, you should consider using certificates from a well-known public Certificate Authority (CA). Using public certificates ensures that others can easily and inexpensively verify the signatures that you place on objects that you distribute to them. If, however, you intend to distribute objects solely within your organization, you may prefer to use Digital Certificate Manager (DCM) to operate your own CA to issue certificates without purchasing them from an outside CA.

The signature on an object represents the system that signed the object, not a specific user on that system (although the user must have the appropriate authority to use the certificate for signing objects). As part of the process of verifying digital signatures, you must decide which Certificate Authorities you trust and which certificates you trust for signing objects. When you elect to trust a CA, you can elect whether to trust signatures that someone creates by using a certificate that the trusted CA issued. When you elect not to trust a CA, you also are electing not to trust certificates that the CA issues or signatures that someone creates by using those certificates.

There are two methods that you can use for signing objects. You can write a program that calls the Sign Object API, or you can use Digital Certificate Manager (DCM) to sign objects. You can use the certificates that you manage in DCM to sign any object that you store in the system's integrated file system, except objects that are stored in a library.

Since V5R1 is currently the only release that supports object signing, there is no support for previous target release objects.

Software vendors who sign objects ship:

- Signed objects
- Verification certificates
- Renewed versions of certificates upon expiry of originals

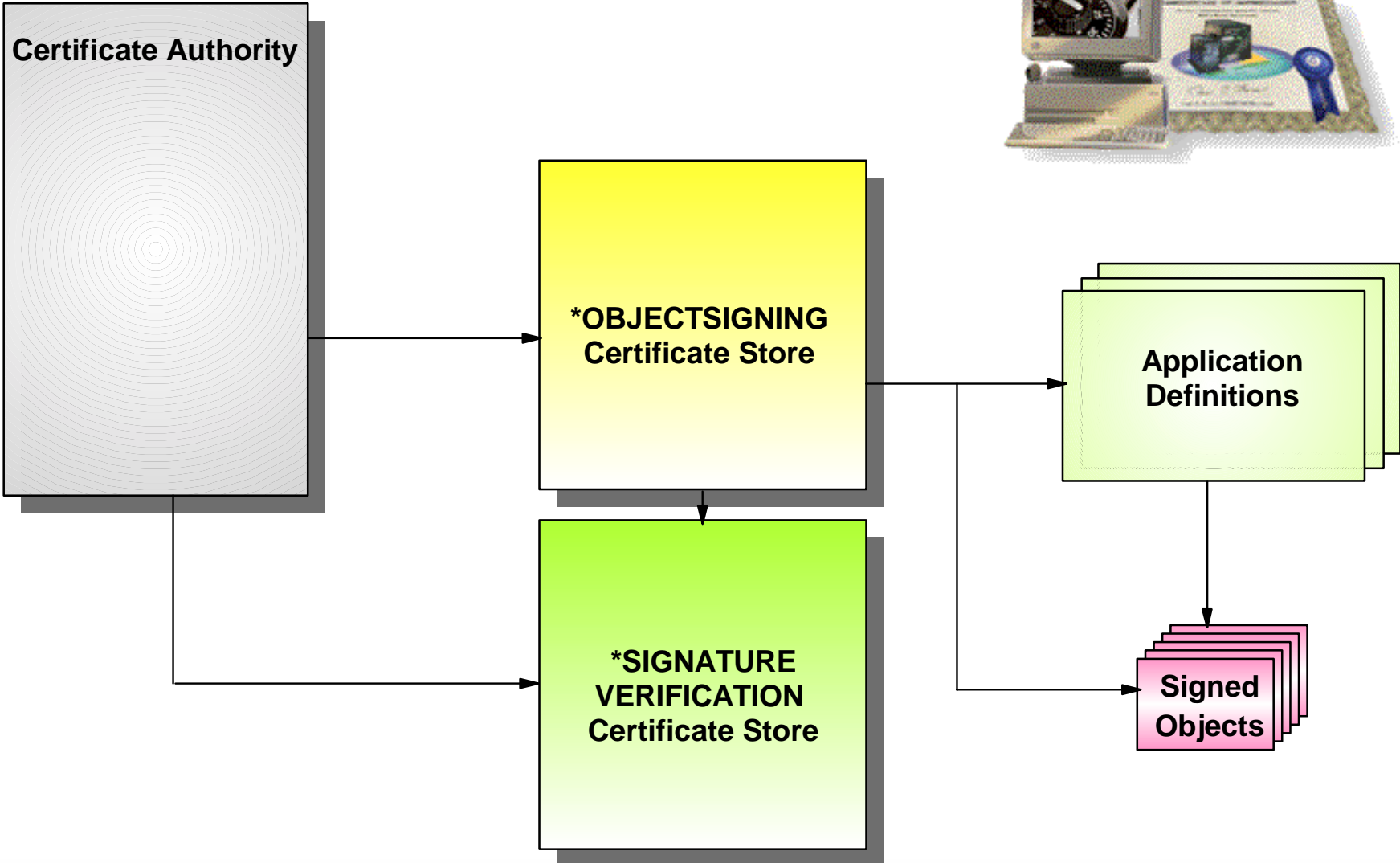
Customers need to:

- Install the certificates
- Verify restore settings
- Validate regularly signature
- Maintain current versions of certificates

A software vendor who wants to increase the integrity of the objects he is distributing, or who wants to prevent tampering with his products and therefore selects to validate it using certificates, must also take care that he distributes verification certificates to his customers and renew these certificates when they expire. He might also advise his customers on how to modify the settings of the verification functions during the restore (see the page on *Restoring Signed Objects* in this presentation) or on how to verify the validity of the signatures of the objects he has distributed (see the page on the *Check Object Integrity Command* in this presentation).

A customer who installs applications that use object signatures will want to include in his system security procedures a practice to ensure the integrity of the signed objects; this will probably include a regular checkup on the signed objects he has installed. The customer also has to implement a life cycle management procedure to refresh certificates which have expired.

# Components for Object Signing



These are the components needed to perform both object signing and object verifying.

There are two types of certificates you need for signing objects:

- A certificate to sign objects, which uses the private key associated with the public key to create the signature. Typically, an \*OBJECTSIGNING certificate will be used by business partners to provide an extra level of integrity that their products have not been tampered with between the creation and distribution time, although everyone can sign an object.
- A certificate to verify an object contains only the base certificate, which holds the public key needed to decrypt the signature made with the private key of the \*OBJECTSIGNING certificate. This certificate is derived from the \*OBJECTSIGNING certificate.

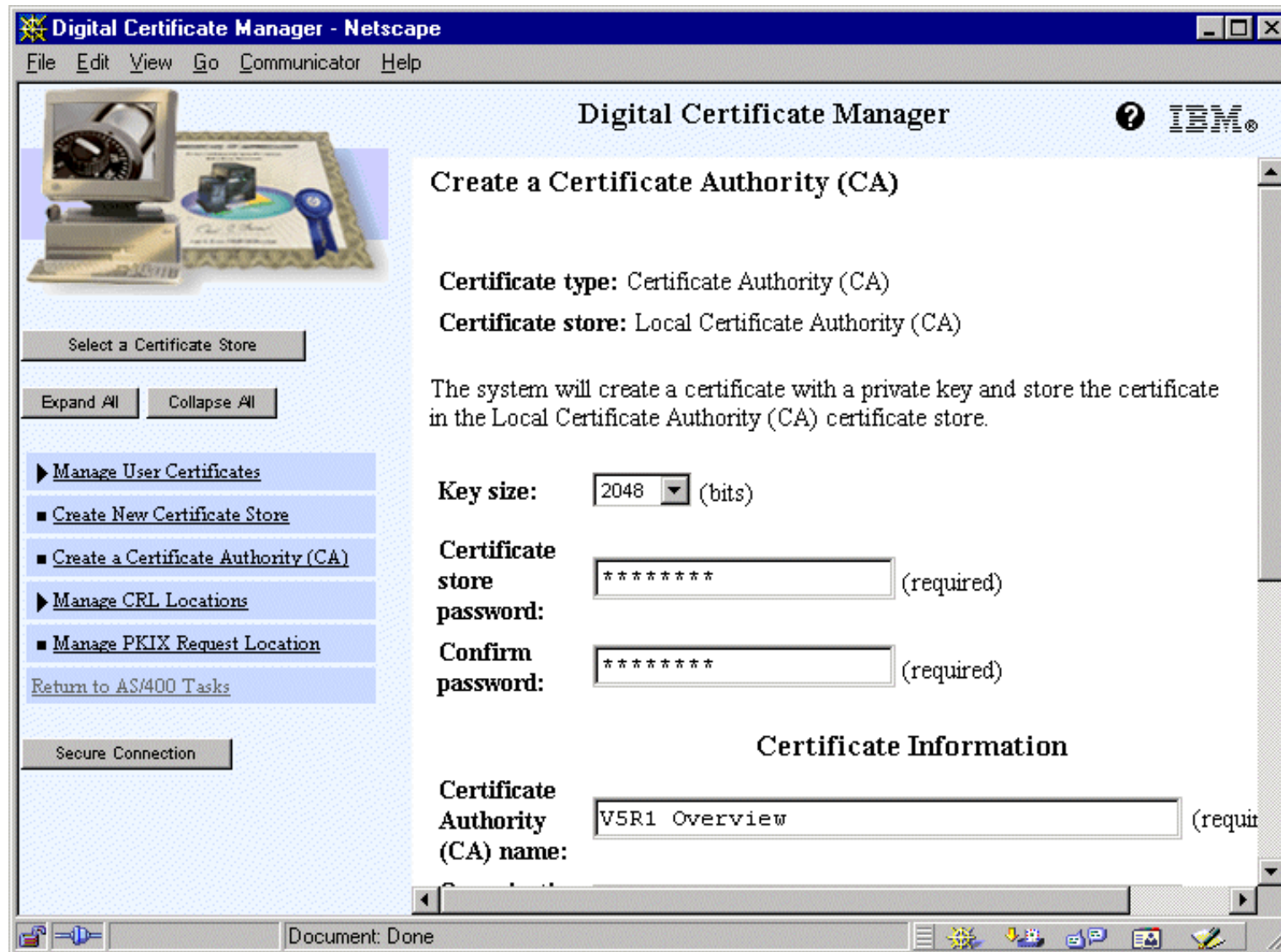
The Digital Certificate Manager (DCM) will automatically validate the certificate before they are added to the system's \*SIGNATUREVERIFICATION store by checking that the Certificate Authority (CA) that signed the certificate is valid; his public key will be used to verify that the certificate is valid. This may take several steps if the certificate is signed by someone who, in turn, is signed by someone else. The iSeries ships several CA certificates for well known CAs.

The \*SIGNATUREVERIFICATION store is saved as part of the Save Security Data (SAVSECDTA) command and restored as part of the Restore User Profile (RSTUSRPRF) command.

If an Object Signing certificate has expired, it can no longer be used to sign objects and needs to be renewed. When a certificate has expired, and when a user tries to validate the signature, the verification succeeds.

DCM is required to create a \*SIGNATUREVERIFICATION certificate store, but you do not need DCM to perform signature verification with the Check Object Integrity (CHKOBJITG) command.

# Create a Certificate Authority

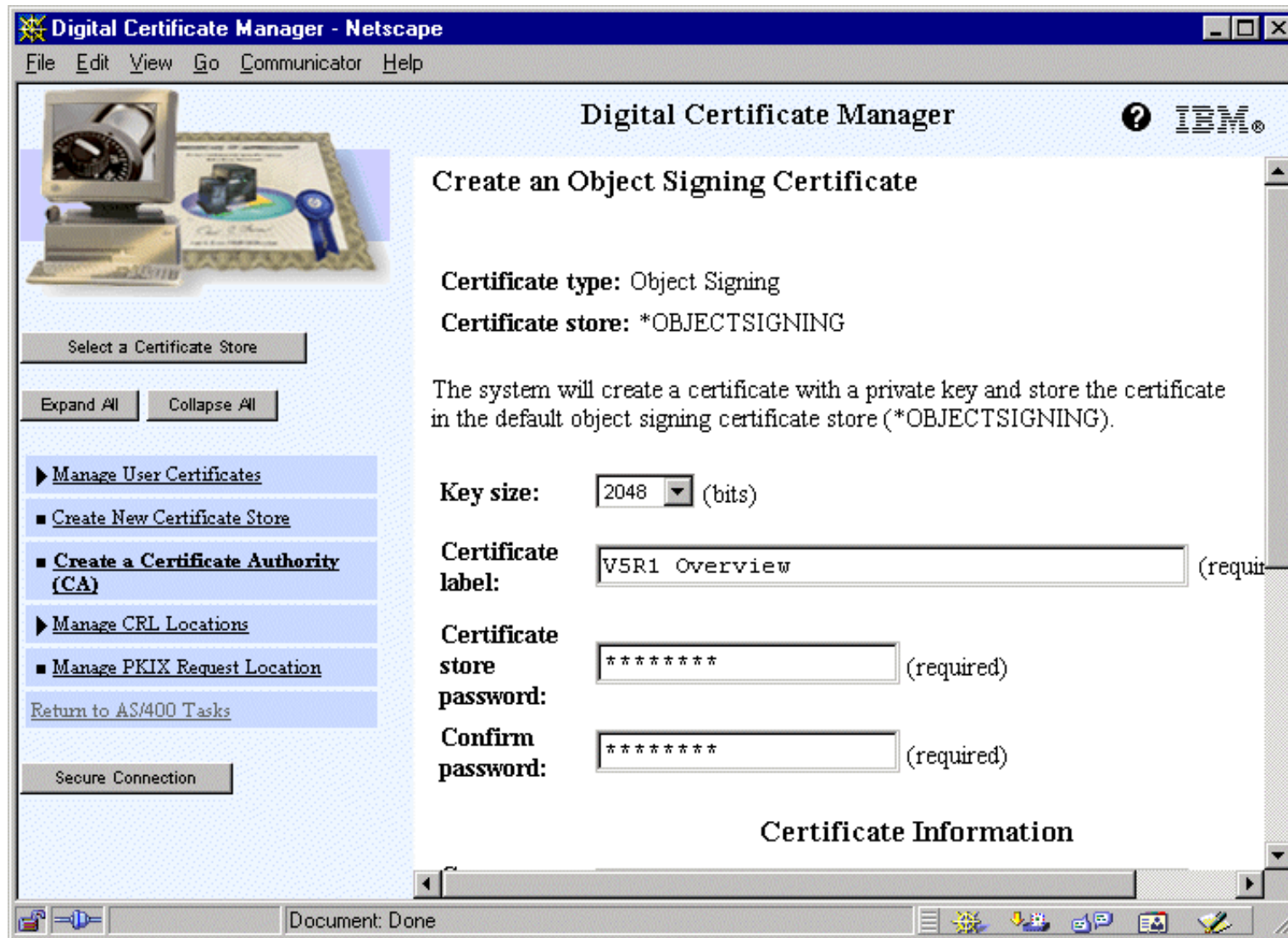


You can sign objects with certificates that you purchase from a public Internet Certificate Authority (CA) or that you create with a private CA in DCM. The process of signing certificates is the same, regardless of whether you use certificates from a local CA or from an Internet CA. For simplicity reasons in this example, we selected to use a local CA.

To use DCM to create and operate a local CA, follow these steps:

- Start a DCM session (use a browser to access the HTTP admin server on your iSeries on port 2001 for non-SSL admin, 2010 for SSL admin).
- In the navigation frame of DCM, select Create a Certificate Authority (CA) to display a series of forms. These forms guide you through the process of creating a CA and completing other tasks needed to begin using digital certificates for SSL, object signing, and signature verification.
- Complete all the forms for this guided task. In using these forms to perform all the tasks that you need to set up a working Certificate Authority (CA), you:
  - Choose how to store the private key for the CA certificate (this step is included only if you have an IBM 4758-023 PCI Cryptographic Coprocessor installed on your iSeries 400. If your system does not have a cryptographic coprocessor, DCM automatically stores the certificate and its private key in the Local Certificate Authority (CA) certificate store).
  - Provide identifying information for the CA.
  - Choose the policy data for your CA.
  - Use the new CA to issue an object signing certificate that applications can use to digitally sign objects. This subtask creates the \*OBJECTSIGNING certificate store; this is the certificate store that you use to manage object signing certificates. This step is illustrated on the next foil.

# Create an Object Signing Certificate



**Digital Certificate Manager - Netscape**

File Edit View Go Communicator Help

**Digital Certificate Manager** ? IBM

### Create an Object Signing Certificate

**Certificate type:** Object Signing  
**Certificate store:** \*OBJECTSIGNING

The system will create a certificate with a private key and store the certificate in the default object signing certificate store (\*OBJECTSIGNING).

**Key size:** 2048 (bits)

**Certificate label:** V5R1 Overview (required)

**Certificate store password:** \*\*\*\*\* (required)

**Confirm password:** \*\*\*\*\* (required)

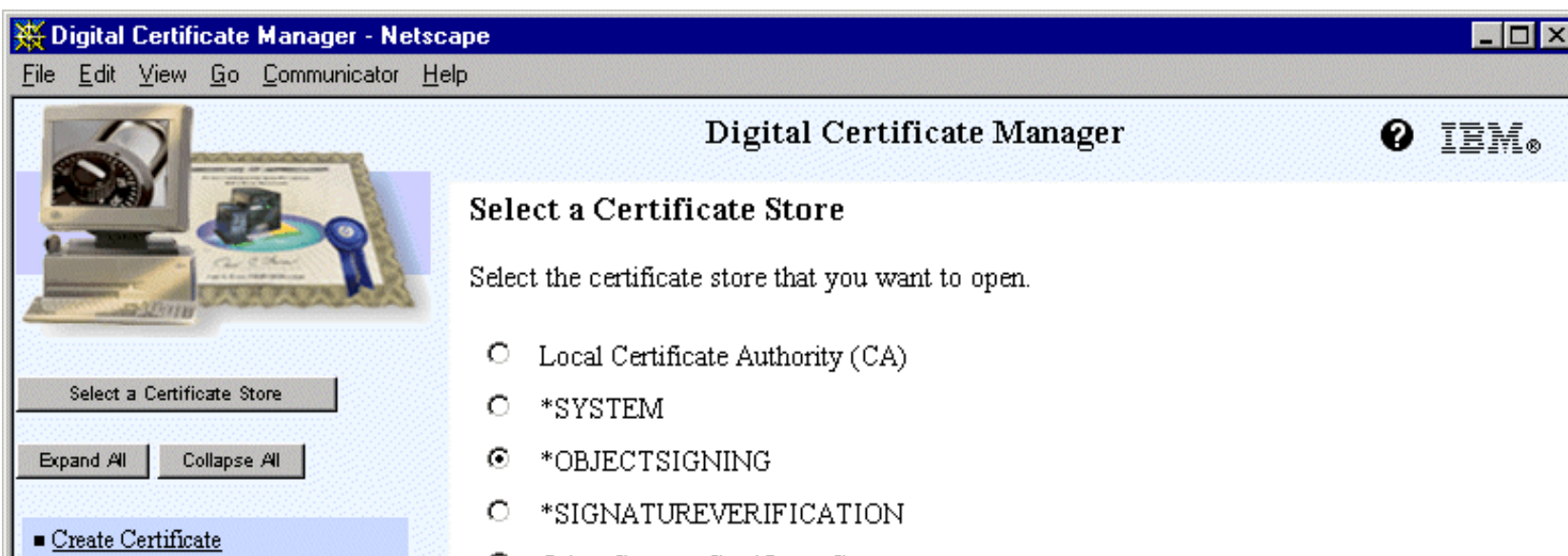
**Certificate Information**

Return to AS/400 Tasks

Secure Connection

Document: Done

# Select Certificate Store



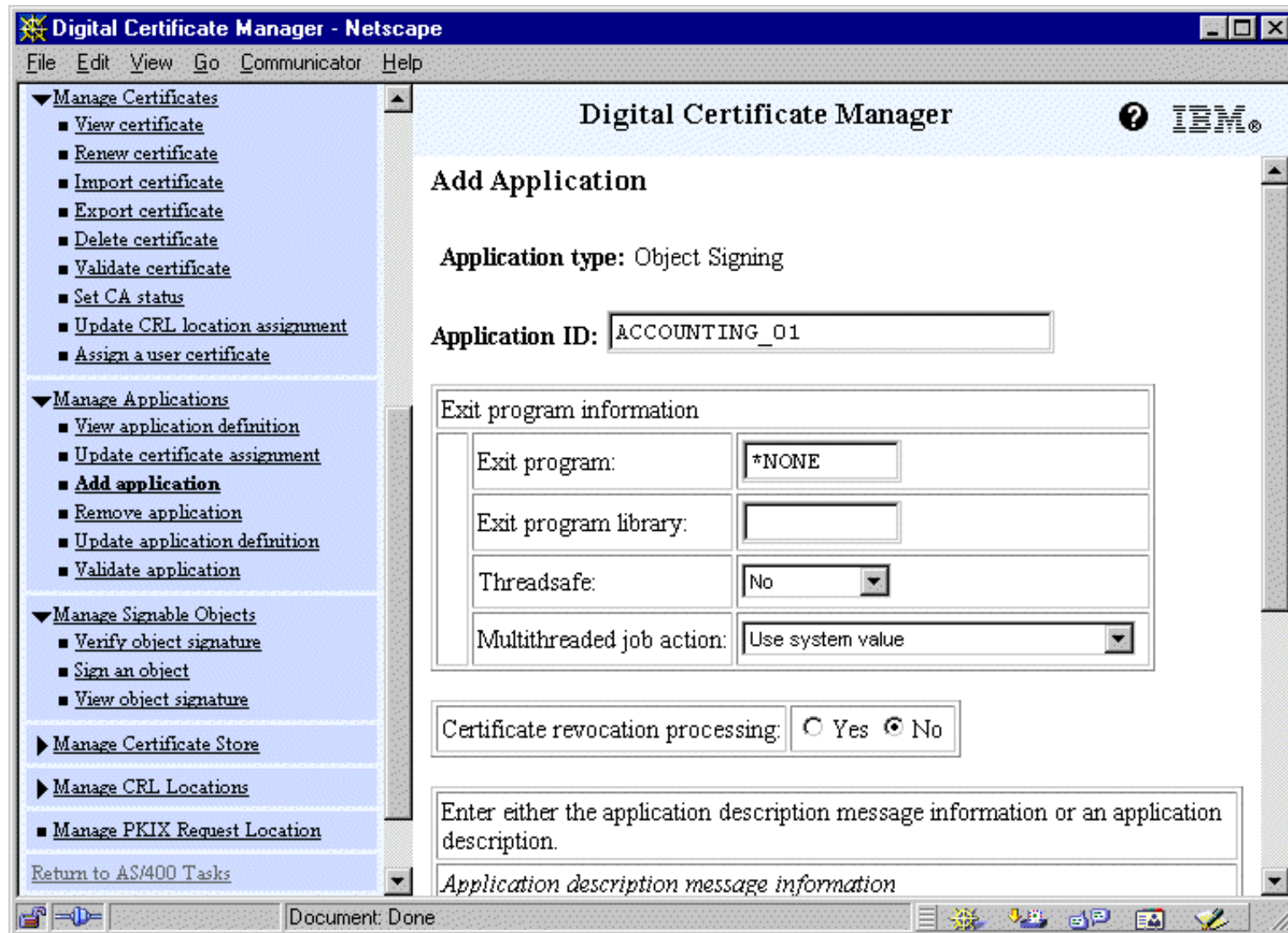
The following pages will guide you through the navigation, required to sign and verify objects using DCM.

Before you can use DCM (or the Sign Object API) to sign objects, you must ensure that certain prerequisite conditions are met:

- You must have created the \*OBJECTSIGNING certificate store, either as part of the process of creating a private CA or by choosing the *Create new Certificate Store* option as part of the process of managing object signing certificates from a public Internet CA.
- The \*OBJECTSIGNING certificate store must contain at least one certificate, either one that you created by using a private CA or one that you obtained from a public Internet CA.
- You must have created at least one object signing application definition to use for signing objects.
- You must have assigned a specific certificate to the object signing application definition that you plan to use to sign objects.

To create an object signing application definition, check the following pages.

# Add an Application Definition

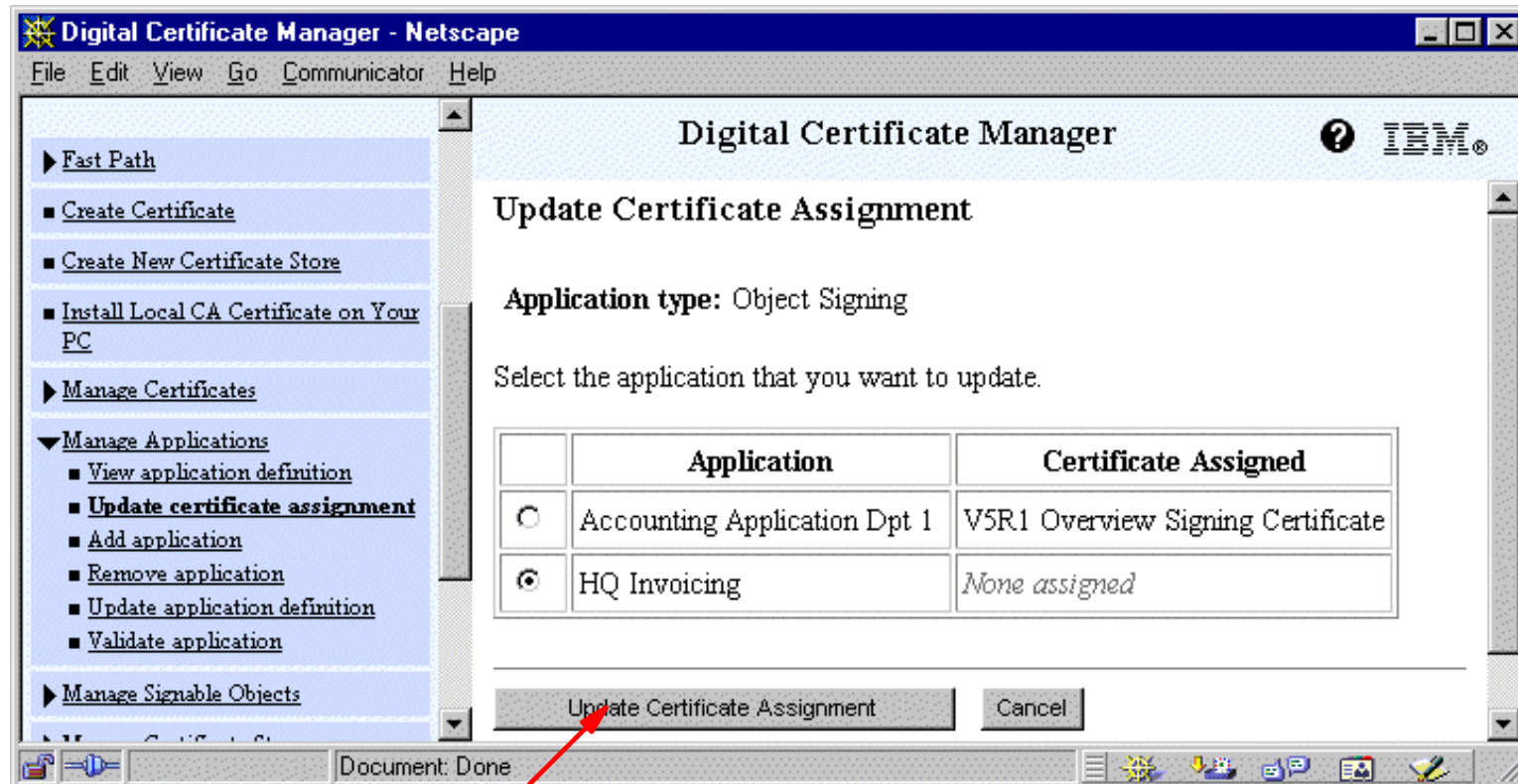


There are two types of application definitions that you can work with in DCM: application definitions for server or client applications that use SSL and application definitions that you use for signing objects. Unlike an SSL application definition, an object signing application does not describe an actual application. Instead, the application definition that you create should describe the type or group of objects that you intend to sign. You must be working in the \*OBJECTSIGNING certificate store to create an object signing application definition.

To create an application definition, follow these steps:

- Start a DCM session.
- Click **Select a Certificate Store** and select the \*OBJECTSIGNING certificate store.
- When the password page displays, provide the password that you specified for the certificate store when you created it and click **Continue**.
- In the navigation frame, select **Manage Applications** to display a list of tasks.
- Select **Add application** from the task list to display a form for defining the application.
- Complete the form and click **Add**.  
This form will prompt for a number of parameters to add, such as an optional exit program identification, whether or not Certificate revocation processing has to be invoked and an identification of the application, stored in a message id or hardcoded.

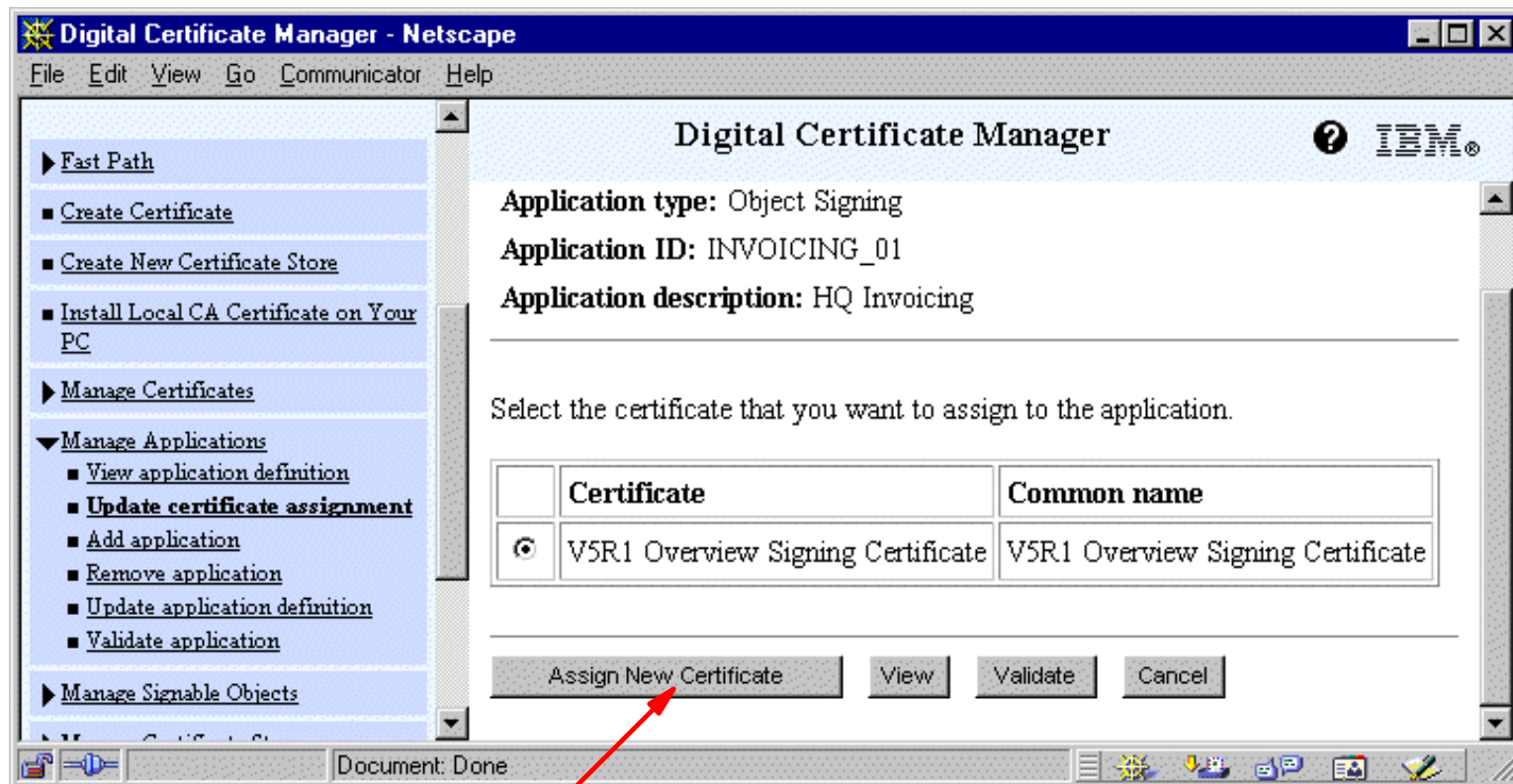
# Update Certificate Assignment



To assign a certificate to an application follow these steps:

- From the task list, select **Update certificate assignment** to display a list of applications for which you can assign a certificate.
- Select the application from the list and click **Update Certificate Assignment** to display a list of certificates that you can assign to the application as shown on the next page.

# Select a Certificate for an Application

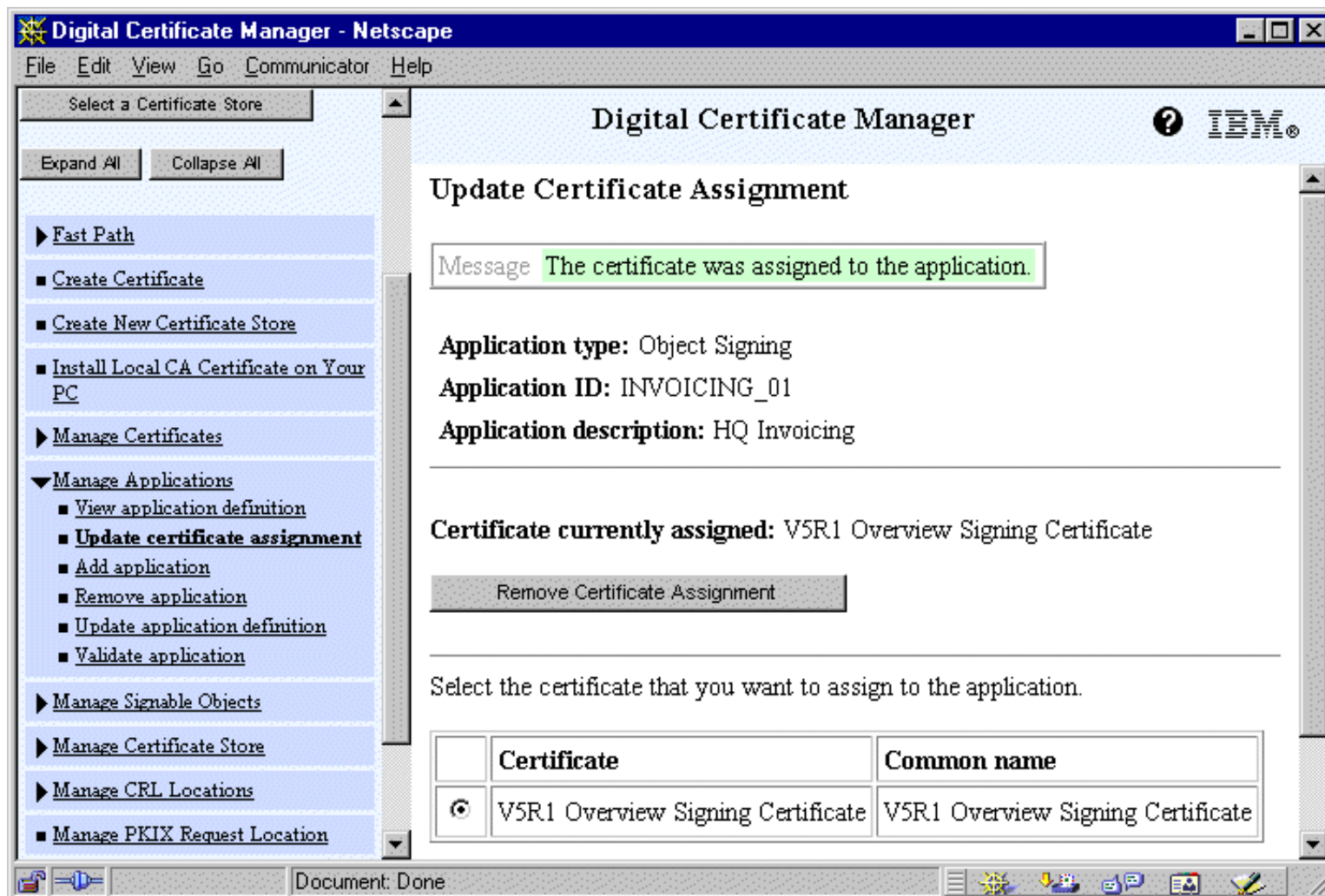


# Notes: Select a Certificate for an Application

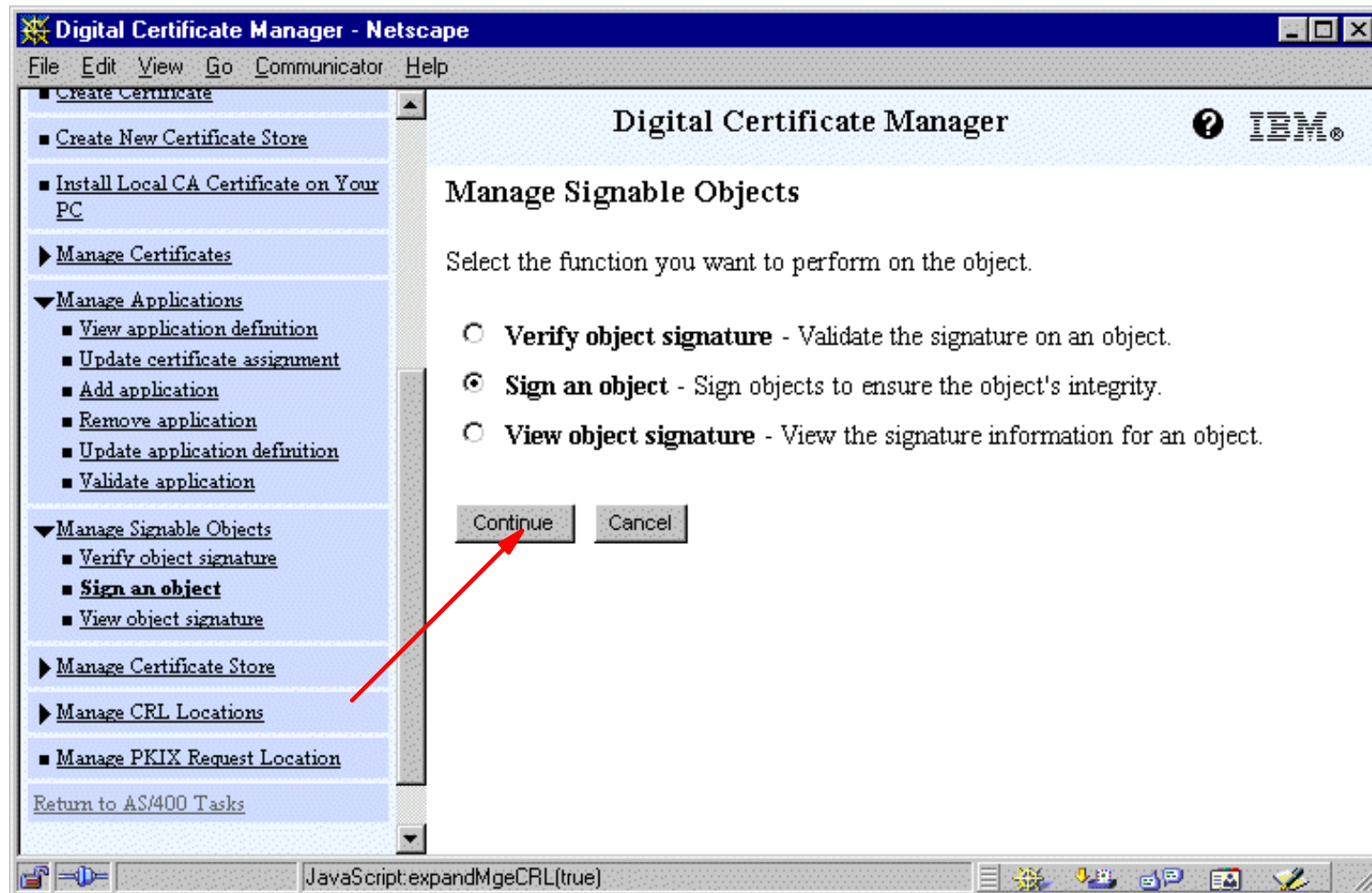
The DCM will present you with all the certificates that you have created. To assign a certificate to an application:

- Select a certificate from the list and click **Assign New Certificate**.
- Upon successful completion, the Results screen, as shown in the next foil, will be presented.

# Update Certificate Assignment - Result



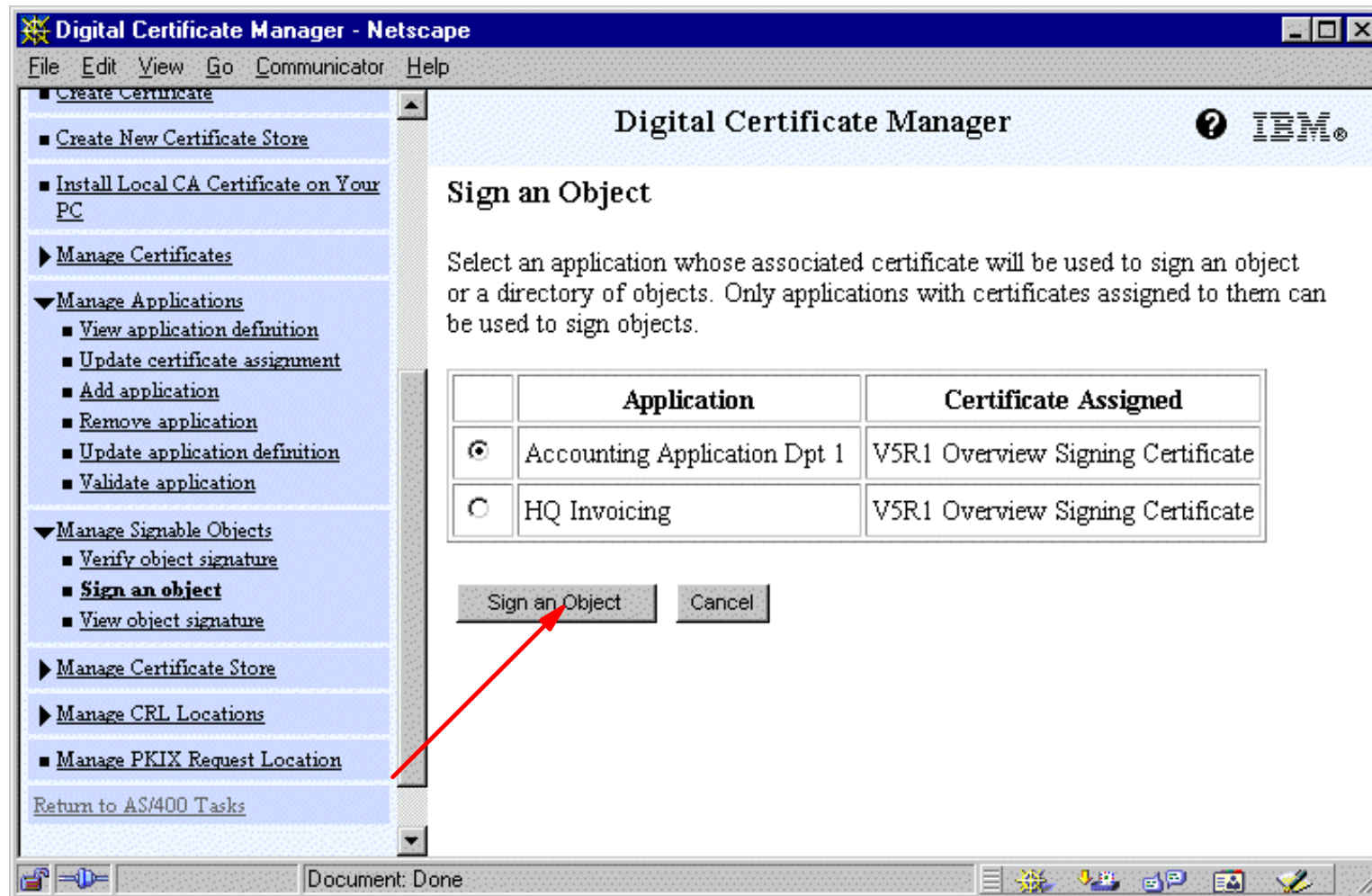
# Signing an Object



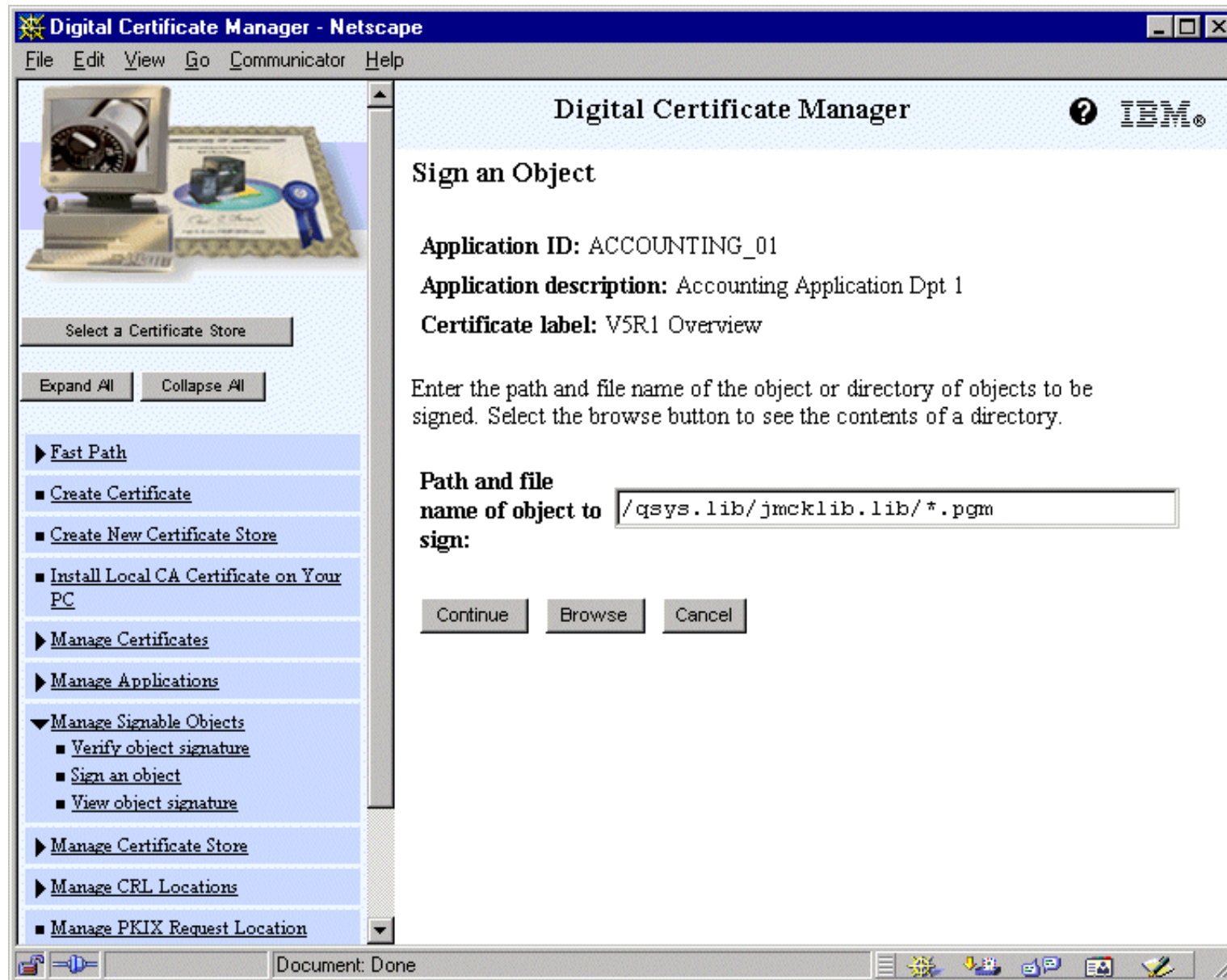
To sign an object, follow these steps:

- Select **Manage Signable Objects** to display a list of tasks.
- From the list of tasks, select **Sign an object** to display a list of application definitions that you can use for signing object, as shown on the next page. These application definitions have been made using the interfaces as described in the previous pages.

# Signing Objects - Select Application



# Identify the Object

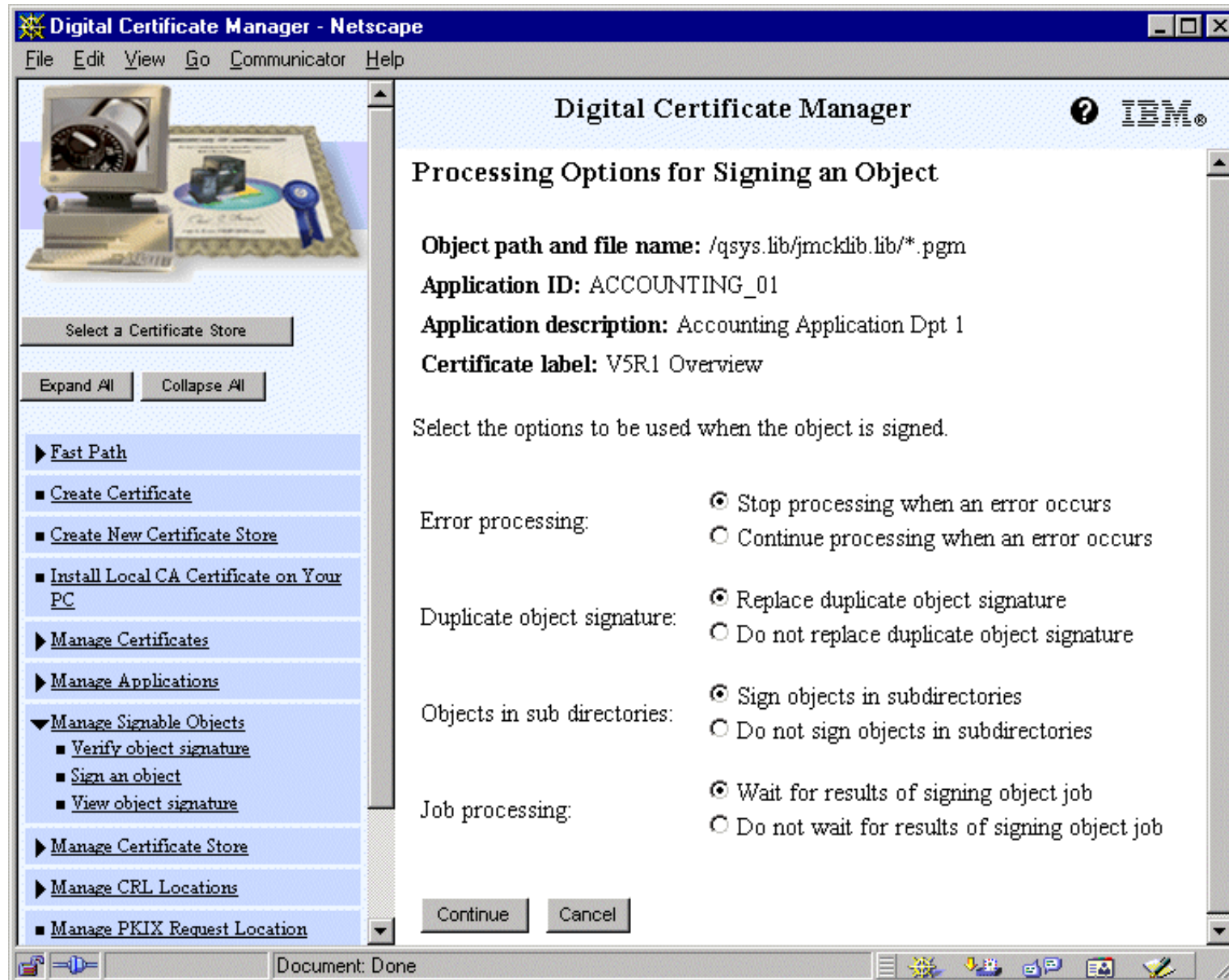


# Notes: Identify the Object

Enter the fully qualified path and file name of the object or directory of objects that you want to sign and click **Continue**. Or, enter a directory location and click **Browse** to view the contents of the directory to select objects for signing. The browse function is used to find the right directory or library. Since you probably want to sign several or all objects in a library or directory, you may need to add the wildcard after returning from the Browse panel.

You must start the object name with a leading slash or you may encounter an error. You can also use certain wildcard characters to describe the part of the directory that you want to sign. These wildcard characters are the asterisk (\*), which specifies "any number of characters," and the question mark (?), which specifies "any single character." For example, to sign all the objects in a specific directory, you could enter */mydirectory/\**; to sign all the programs in a specific library, you could enter */QSYS.LIB/QGPL.LIB/\*.PGM*. You can use these wildcards only in the last part of the path name; for example, */mydirectory\*/filename* results in an error message. If you want to use the Browse function to see a list of library or directory contents, you should enter the wildcard as part of the path name prior to clicking **Browse**.

# Processing Options



Use this page to specify the processing options to use when signing the specified object or objects.

## ■ Error Processing:

**Stop processing when an error occurs:** If the signing process for one object in the sequence fails, the signing process stops without creating signatures on any remaining objects.

**Continue processing when an error occurs:** The process attempts to sign all objects in the job and the job results are sent to the file that you specified.

## ■ Duplicate object signature:

Specifies how the application should handle the signing process when the application is using a certificate to re-sign an object.

**Replace duplicate object signature:** The application replaces the original signature, if it is a duplicate signature, with a new digital signature.

**Do not replace duplicate object signature:** The application leaves the original signature in place and continues processing.

## ■ Objects in subdirectories

**Sign objects in subdirectories:** Signs objects in any subdirectories found in the path and file name that you specified. For example, you specified a path name and file of '/jkl/\*', and this path has two subdirectories within it, '/jksub1/' and '/jksub2/'. If you select this option, the signing process individually signs all objects in the main directory and both subdirectories.

**Do not sign objects in sub directories:** Signs only those objects in the main directory that you specified in the path and file name, but not to individually sign each object within any subdirectories. When you select this option, the signing process ignores any subdirectories in the path and file name that you specified.

## ■ Job processing

**Wait for results of signing object job:** DCM waits until the object signing process finishes for all objects before displaying signing results.

**Do not wait for results of signing object job:** DCM runs the object signing job in batch mode and write the results of the job to a file. You can then check the job results at a later time. Selecting this option is useful when you are signing a large number of objects because you can continue using DCM to perform other tasks while the signing process runs.



**Digital Certificate Manager - Netscape**

File Edit View Go Communicator Help

**Digital Certificate Manager** ? IBM

### Job Results File for Signing an Object

**Object path and file name:** /qsys.lib/jmcklib.sys/\*.pgm  
**Application ID:** ACCOUNTING\_01  
**Application description:** Accounting Application Dpt 1  
**Certificate label:** V5R1 Overview

Enter the path and file name where the signing job results should be stored. Select the browse button to see the contents of a directory.

**Path and file name to store job results:**

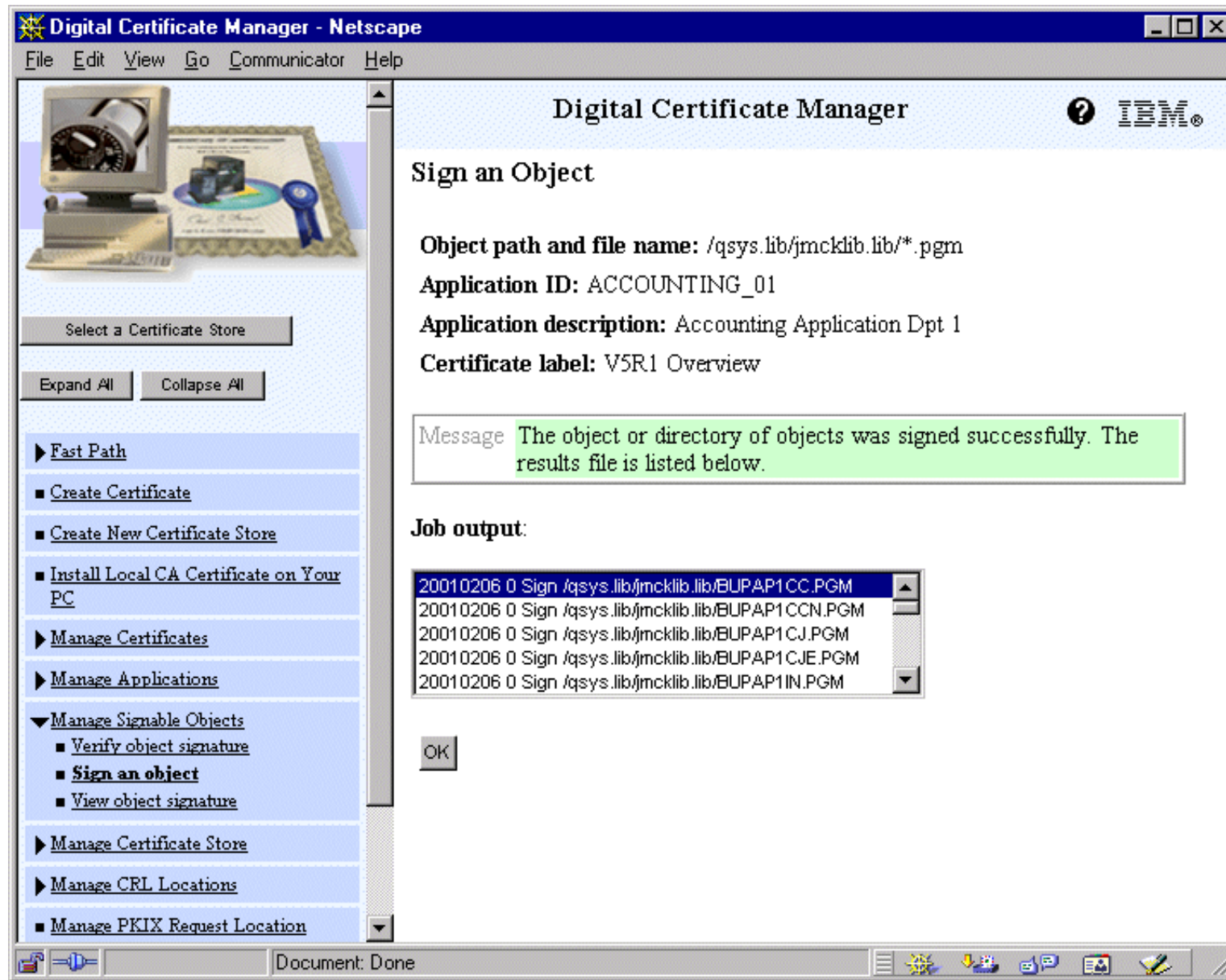
**Left Sidebar:**

- Select a Certificate Store
- Expand All Collapse All
- Fast Path
  - Create Certificate
  - Create New Certificate Store
  - Install Local CA Certificate on Your PC
- Manage Certificates
- Manage Applications
- Manage Signable Objects
  - Verify object signature
  - Sign an object**
  - View object signature
- Manage Certificate Store
- Manage CRL Locations
- Manage PKIX Request Location

Use this page to specify a path and file name for storing the object signing job results. You can choose to create a new file for storing the results, or you can click Browse to select an existing file from a directory.

Results files can not reside under /QSYS.LIB and are required to have a CCSID 13488.

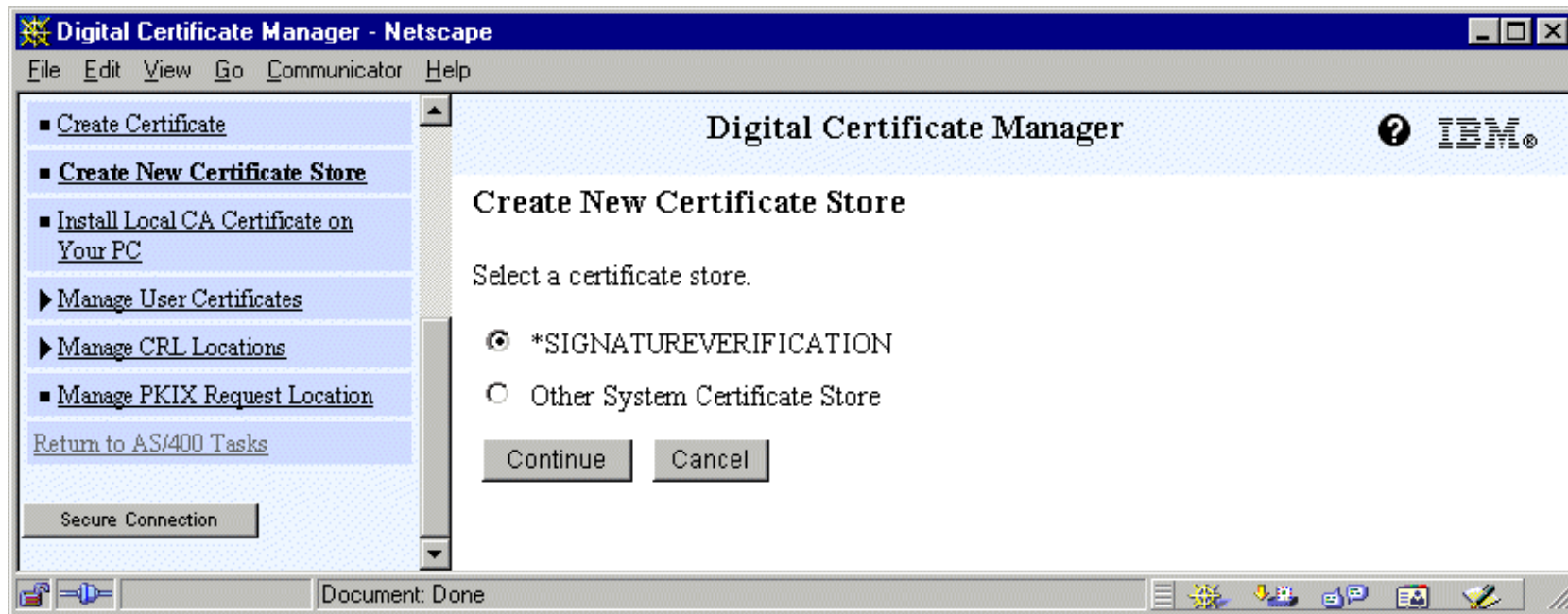
# Completion Messages



# Notes: Completion Messages

Use this page to verify the success of the object signing process, or to review error messages if the process failed. These messages are displayed in the **Signing job output** list box.

# Setup Signature Verification Store



You can use Digital Certificate Manager (DCM) to manage the signature verification certificates that you use to validate digital signatures on objects. As we have seen, you use a certificate's private key to sign an object and thus create the signature. When you send the signed object to others, you must include a copy of the certificate that signed the object. You do this by using DCM to export the object signing certificate (without the certificate's private key) as a signature verification certificate. You can export a signature verification certificate to a file that you can then distribute to others. Or, if you want to verify signatures that you create, you can export a signature verification certificate into the \*SIGNATUREVERIFICATION certificate store.

To validate a signature on an object, you must have a copy of the certificate that signed the object. You use the signing certificate's public key, which the certificate contains, to examine and verify the signature that was created with the corresponding private key. Therefore, before you can verify the signature on an object, you must obtain a copy of the signing certificate from whomever provided you with the signed objects.

You must also have a copy of the Certificate Authority (CA) certificate for the CA that issued the certificate that signed the object. You use the CA certificate to verify the authenticity of the certificate that signed the object. DCM provides copies of CA certificates from most well-known CAs. If, however, the object was signed by a certificate from a public CA or another private CA, you must obtain a copy of the CA certificate before you can verify the object signature.

So, before you can use DCM to verify signatures on objects:

- You must have created the \*SIGNATUREVERIFICATION certificate store to manage your signature verification certificates.

Note: You can perform signature verification while working within the \*OBJECTSIGNING certificate store in cases where you are verifying signatures for objects that were signed on the same system. The steps that you perform to verify the signature in DCM are the same in either certificate store. However, the \*SIGNATUREVERIFICATION certificate store must exist and must contain a copy of the certificate that signed the object even if you perform signature verification while working within the \*OBJECTSIGNING certificate store.

- The \*SIGNATUREVERIFICATION certificate store must contain a copy of the certificate that signed the objects.
- The \*SIGNATUREVERIFICATION certificate store must contain a copy of the CA certificate that issued the certificate that signed the objects.

# Copy Signing Certificate



# Notes: Copy Signing Certificate

If you have certificates in the \*OBJECTSIGNING certificate store, you should choose to export them as verification certificates. If you sign an object with an object signing certificate, you may want to use DCM to verify the authenticity of the signature. DCM can verify the authenticity of the signature only if the signing certificate exists as a verification certificate in the \*SIGNATUREVERIFICATION store.

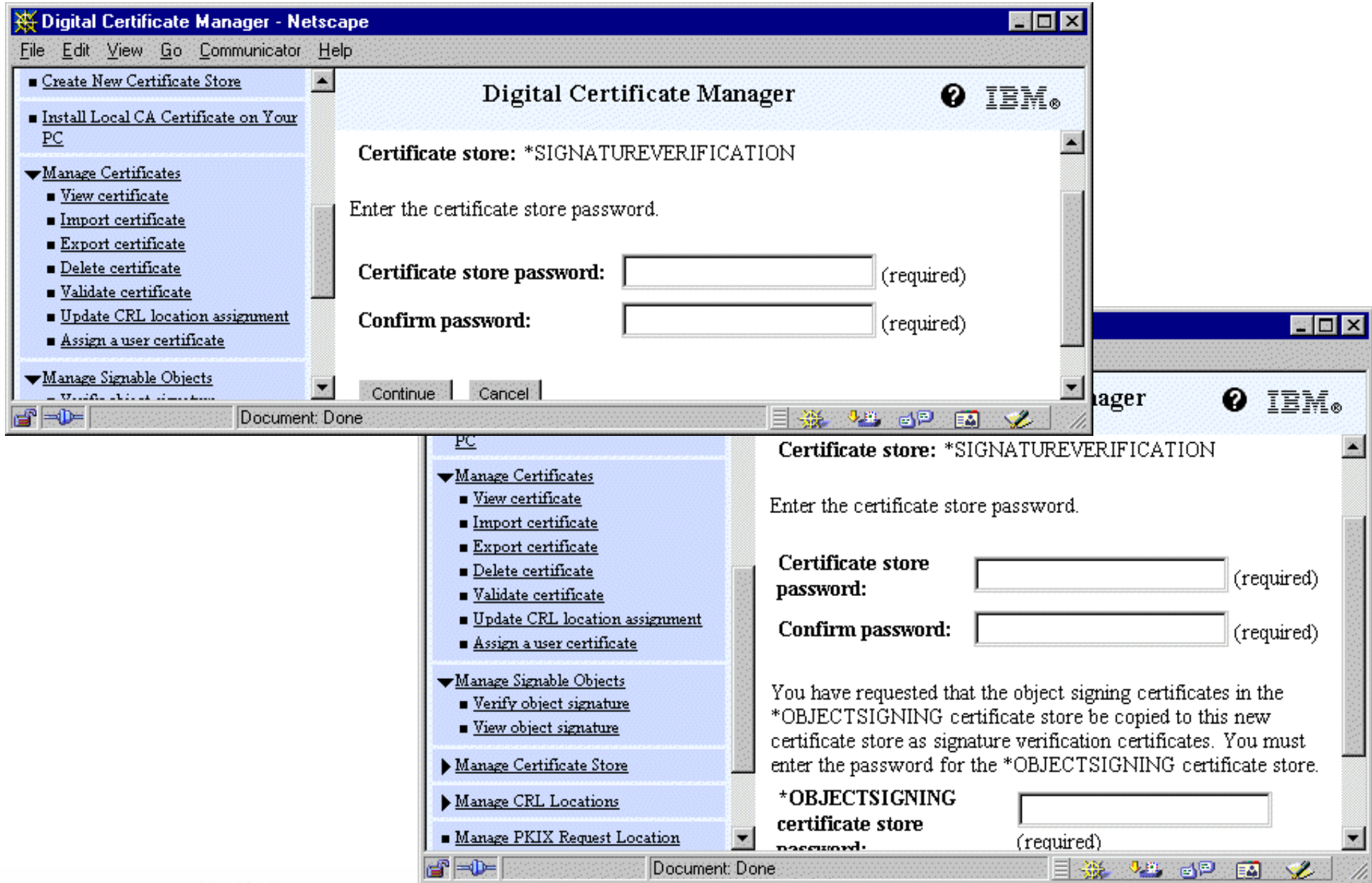
Click **Yes** to copy certificates in the \*OBJECTSIGNING certificate store to use as signature verification certificates when you create the store. DCM copies the object signing certificates without copying the private keys for the certificates. You use a certificate's public key to verify a signature. The public key is an integrated part of a certificate. You use a certificate's private key to sign a certificate. The private key is not part of the actual certificate and is stored separately for security reasons.

Click **No** to create the store without copying the object signing certificates in the \*OBJECTSIGNING store to use as signature verification certificates in this store. You can add an object signing certificate to the \*SIGNATUREVERIFICATION store as a verification certificate after the store exists. Access the \*OBJECTSIGNING certificate store, select **Manage Certificates** from the navigation pane, and use the **Export certificate** task.

When you complete this task, the new certificate store contains a default set of Certificate Authority (CA) certificates for validation purposes.

When you are exporting your verification signature to another system, create the \*SIGNATUREVERIFICATION store on this system and import the CA certificate (if needed) and the verification signature into this store from your originating \*SIGNATUREVERIFICATION store.

# Create Verification Certificate/Store

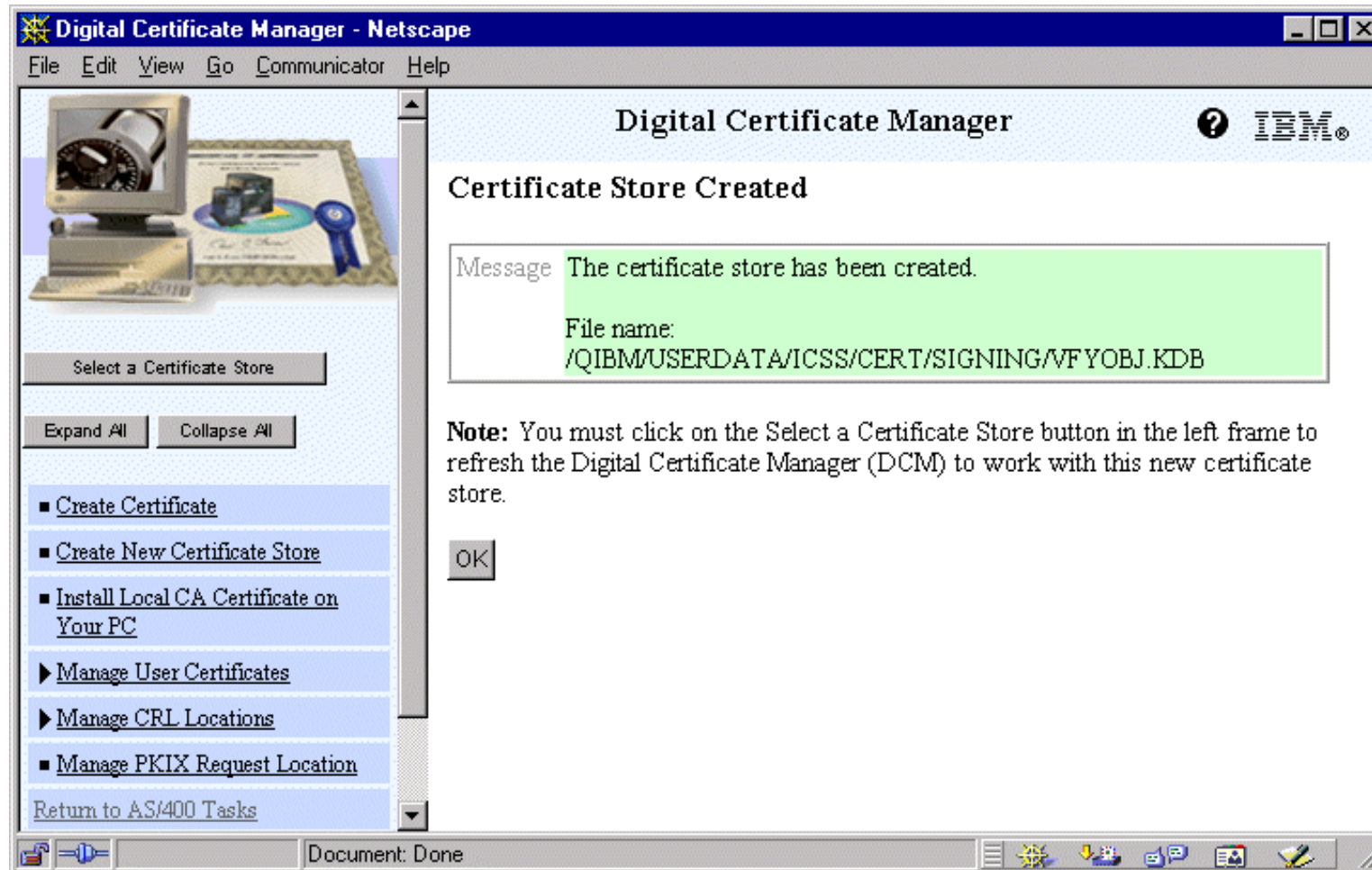


The page in the lower right-hand corner displays when you create the \*SIGNATUREVERIFICATION certificate store and have selected **Yes** on the previous page to copy the \*OBJECTSIGNING certificates in this store. You enter the password for the store and the password for the \*OBJECTSIGNING store.

When you did select **No** on the previous page, it will prompt you only for a password for the store.

Upon successful completion of the creation of the store, the system will bring up a screen that identifies the location of the store (see next page).

# Location of Verification Store



# Verify Object Signature

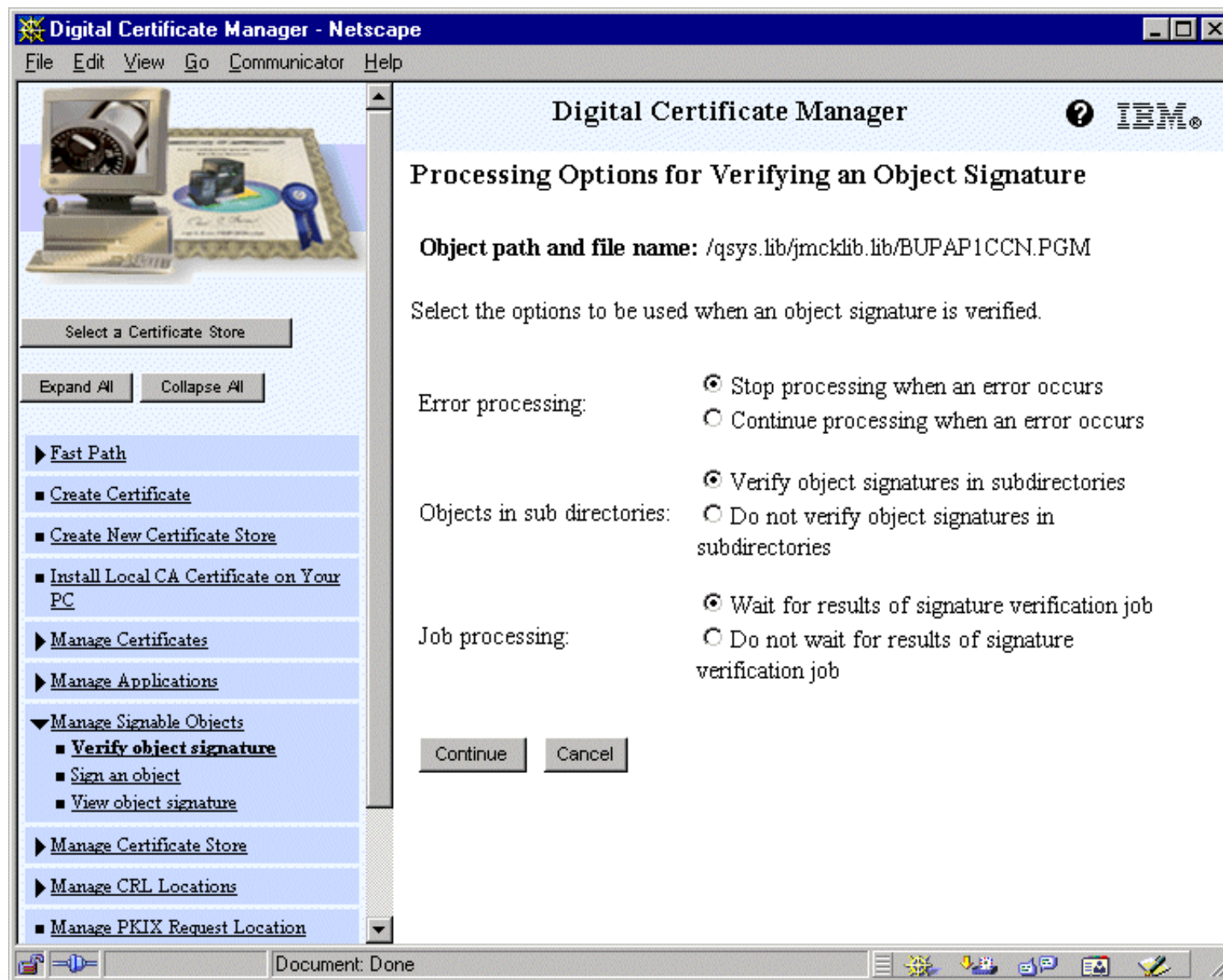


You can use Digital Certificate Manager (DCM) to verify the authenticity of digital signatures on objects. When you verify the signature, you ensure that the data in the object has not been changed since the object owner signed the object.

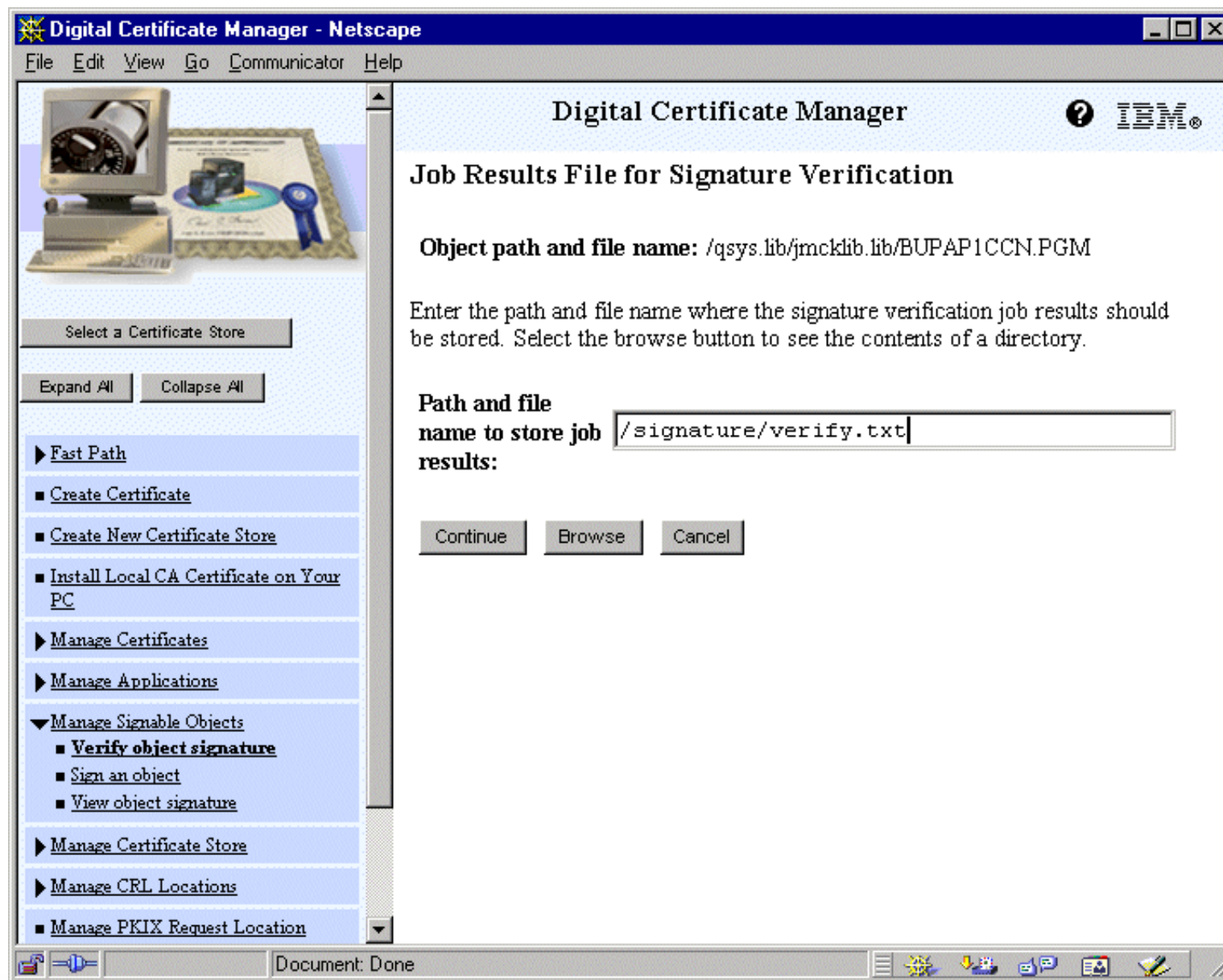
To verify an object signature:

- In the navigation frame, click **Select a Certificate Store** and select \*SIGNATUREVERIFICATION as the certificate store to open.
- Enter the password for the \*SIGNATUREVERIFICATION certificate store and click **Continue**.
- After the navigation frame refreshes, select **Manage Signable Objects** to display a list of tasks.
- From the list of tasks, select **Verify object signature** to specify the location of the objects for which you want to verify signatures.
- In the field provided, enter the fully qualified path and file name of the object or directory of objects for which you want to verify signatures and click **Continue**. Or, enter a directory location and click **Browse** to view the contents of the directory to select objects for signature verification.
- Select, as shown on the following pages, the processing options that you want to use for verifying the signature on the selected object or objects and click **Continue**. If you select the option to run the job in batch mode, you must next select a file for storing the job results.
- Specify the fully qualified path and file name to use for storing job results for the signature verification operation and click **Continue**. Or, enter a directory location and click **Browse** to view the contents of the directory to select a file for storing the job results. A message displays to indicate that the job was submitted to verify object signatures. To view the job results, see job **QOBJSGNBAT** in the job log. Remember however, that there will be a joblog only when errors were encountered.

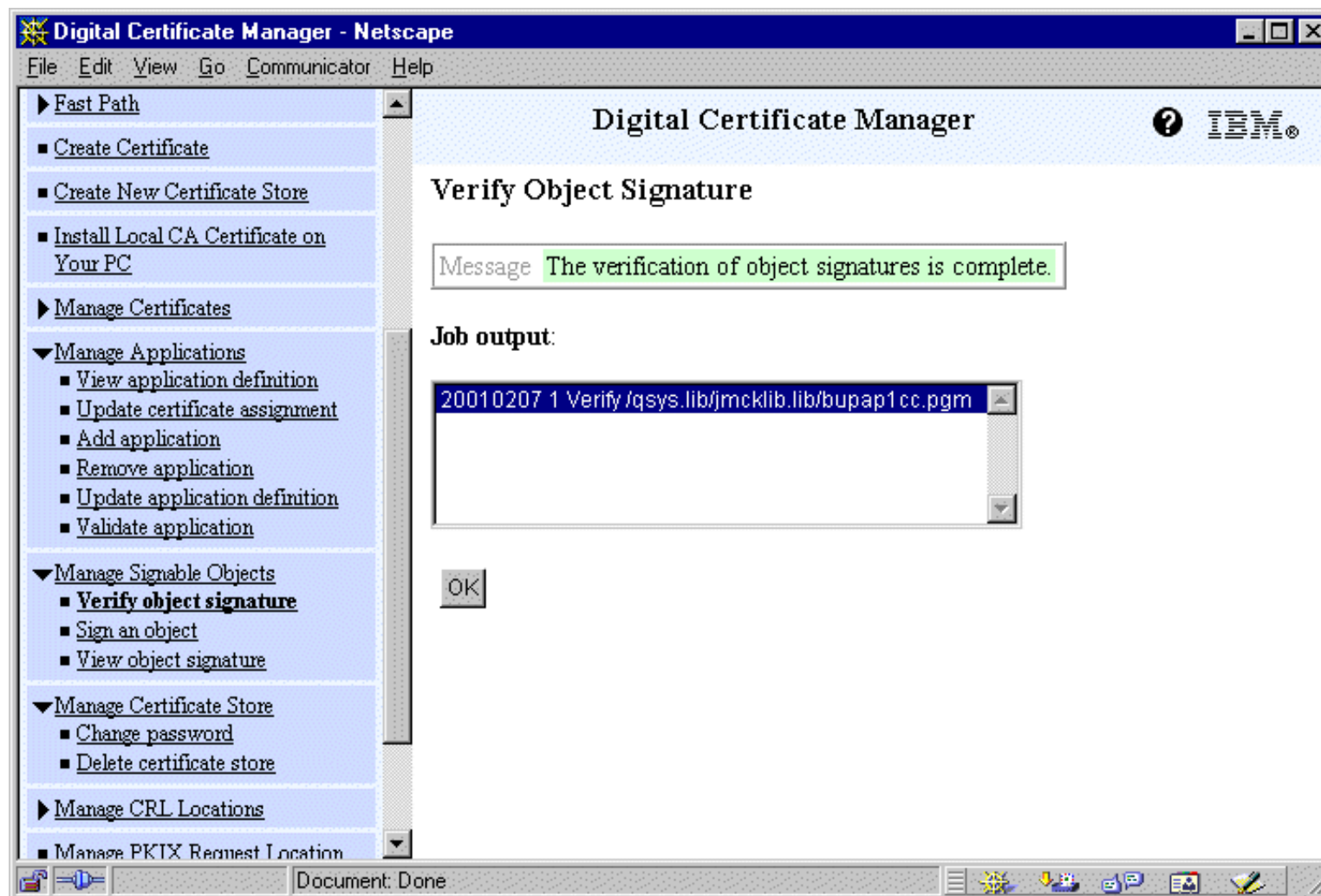
# Verification Processing Options



# Verification Results File



# Verification Completion Message



# Check Object Integrity Command

## Check Object Integrity (CHKOBJITG)

Type choices, press Enter.

```
User profile, or . . . . . _____ Name, generic*, *ALL
Object . . . . . > '/QSYS.LIB/JMCKLIB.LIB/*.PGM'

File to receive output . . . . . > VERIFYSIG Name
Library . . . . . > JMCKLIB Name, *LIBL, *CURLIB

Output member options:
Member to receive output . . . *FIRST Name, *FIRST
Replace or add records . . . . *REPLACE *REPLACE, *ADD
Check domain . . . . . *YES *YES, *NO
Check program and module . . . *YES *YES, *NO
Check command . . . . . *YES *YES, *NO
Check signature . . . . . > *ALL *SIGNED, *ALL, *NONE
Directory subtree . . . . . *none *NONE, *ALL
```

# Notes: Check Object Integrity Command

This command checks either the objects owned by a specific user profile, the objects that match a specified path name or all objects on the system to determine if any objects have integrity violations. If an integrity violation has occurred, the object name, the library or path name, the object type, its owner and the type of failure are logged into a database file. This function will also log all objects that do not have a digital signature but that can be signed, objects that could not be checked and objects whose format requires changes to be used on this system implementation (IMPI to RISC conversion). Objects that cannot be checked are compressed, damaged, saved with storage freed or in debug mode.

You need \*AUDIT authority to execute this command.

New System Value Verify object on restore (QVFYOBJRST)

Provides granularity for restore options of signed objects

- Do not verify
- Verify, restore all objects
- Verify, restore unsigned objects and signed objects with a valid signature
- Verify, do not restore unsigned but restore any signed object
- Verify, do not restore unsigned objects, restore signed objects only if signature is valid

The system value, *Verify object on restore* (QVfyOBJRST), allows to specify what option is to be taken when signed or unsigned objects are restored onto a system. This way, the system can check or ignore the presence and the validity of signatures attached to an object which is being restored.

If an object has a signature, and the system value is set to check the signature, you need to have the certificate that the originator created for verification and his certificate authority installed on the system, otherwise the verification of any signed object will fail. If an object is signed by an untrusted certificate (e.g. the certificate is not present in the \*SIGNATUREVERIFICATION certificate store), it will be treated as an unsigned object

If auditing is being used, and the auditing value in the *Security auditing level* (QAUDLVL) includes \*SAVRST, entries will be put in the auditing journal that reflect the status of the restored object (signed or not) and the result of the verify action (accepted or not). The entries in the auditing journal have a code **T** and a type **OR**. No record is entered if the object did not restore. A **ZC** entry will be created for each object of which the signature was removed during the restore due to retranslation.

Signatures for save files are verified at the time objects are restored from them, not when the save file itself is restored. The signature of the save file will be checked, but restores will be allowed from unsigned save files.

A *Clear Save File* (CLRSAVF) will remove all signatures from the save file. An empty or cleared save file cannot be signed.

It should be clear that choosing either option *Verify, do not restore unsigned but restore any signed object* or *Verify, do not restore unsigned objects, restore signed objects only if signature is valid* has a major impact if both signed and unsigned objects are available on the system.

# Password Support

**IBM @server.** For the next generation of e-business.

# Password Security Changes: Requirements

Pre-V5R1 security setup relies on one single system value (QSECURITY) / procedures to define authorities and rights

Not all accesses to security information logged

Passwords coming from less secure systems (e.g. NetServer environment) must be removable

Up to V4R5 OS/400 security relied primarily on the correct definition of the QSECURITY system value and upon a number of definitions of associated system values and of procedures established around the protection of a system and its resources, which was synchronized with the actual implementation of authorities and rights of the OS/400 objects. Although this implementation had a proven record of being virtually "unhackable", the exposure of attacks to reveal and abuse user profiles and their passwords became over time bigger and bigger. If a value of 30 or more for the QSECURITY system value was used, and if the number of user's holding Security Officer (\*ALLOBJ, \*SECADM) rights or having Save System (\*SAVSTS) and/or service (\*SERVICE) authorities was kept to a strict minimum, the access to security information was limited. However, it implied that the users with any of these attributes were highly trusted, since all of their actions could be audited, with the exception of access to the Service Tools and what was done in those applications.

Also, users accessed the iSeries resources via NetServer often using the user ID and password from their Windows desktop logon; storing these profiles on the iSeries presents a clear exposure since the passwords on the PCs can easily be retrieved, giving potential hackers thereby an entry to system objects.

# Password Level Definition

System Value Password Level controls which password security is implemented

## Display System Value

```
System value . . . . . : QPWDLVL  
Description . . . . . : Password level
```

```
Password level . . . . . : 0
```

- 0            User profile passwords with a length of 1-10 characters are supported.
- 1            User profile passwords with a length of 1-10 characters are supported. AS/400 NetServer passwords for Windows 95/98/ME clients will be removed from the system.
- 2            User profile passwords with a length of 1-128 characters are supported.
- 3            User profile passwords with a length of 1-128 characters are supported. AS/400 NetServer passwords for Windows 95/98/ME clients will be removed from the system.

Please refer to the documentation on *Operations Navigator* to obtain an overview of the interface implementation for this system value.

The shipped value for the *Password Level* (QPWDLVL) system value is "0". Passwords for this level and for level "1" can consist of characters A-Z, 0 - 9, \$.@ # and underscore. You need to have \*ALLOBJ and \*SECADM special authorities to change this system value, and changes will take effect only at the next IPL. The *Display Security Attributes* (DSPSECA) (see next page) command will reflect the settings and, optionally, the pending changes for the Password Level attribute. Level "2" and "3" support upper and lower case for any character. Since the password level is implemented on a system-wide base, it is obvious that a single user cannot choose the level at which she/he wants to have his/her password secured.

Starting with V5R1, all passwords are also encrypted using SHA-1. If the value of "0" or "1" is being used, the DES encryption is being used to signon as was done before. If a value of "2" or "3" is selected, an SHA-1 generated password token is used to signon.

To enable migration, the DES encryption value for level "0" to "2" is kept on the system for each user ID and password. At level "0" to "2", newly created user IDs and passwords will continue to have both a DES and a SHA-1 encrypted version of their password. For level "3", only the SHA-1 version is created, stored and used.

The clear text password will be encrypted producing a 20-byte password token (also referred to as '*passphrase*') as follows:

- Conversion of the clear text password to Unicode CCSID 13488
- Conversion of the clear text user ID to Unicode CCSID 13488
- Use of SHA-1 to hash the ID and password producing the 20-byte token.

Other changes to the system values that define the password definitions:

- The *Password Validation Program* (QPWDVLDPGM) value can now refer to \*REGFAC, a special value indicating that the password validation program will be retrieved from the registration facility.
- If QPWDLVL is set at 2 or 3, the value entered in the *Limit characters in password* (QPWDLMTCHR) will be ignored; at the same time, the values for *Maximum* and *Minimum password length* (QPWDMAXLEN and QPWDMINLEN) are modified to accept a value of 128.

# Display Security Attributes

Display Security Attributes (DSPSECA) enhanced to display password level

Pending password level change - change is only effective after IPL

```
Display Security Attributes                                     System:  AS80
User ID number . . . . . : 320
Group ID number . . . . . : 105
Security level . . . . . : 50
Password level . . . . . : 0
Pending password level . . . . . : 1
```

# Password Level Cross-chart

	<i>Level 0</i>	<i>Level 1</i>	<i>Level 2</i>	<i>Level 3</i>
<i>Maximum Length</i>	10	10	128	128
<i>Encryption Used</i>	DES	DES	SHA-1	SHA-1
<i>Encryption Stored</i>	DES SHA-1	DES SHA-1	DES SHA-1	SHA-1
<i>NetServer Passwords Removed</i>	No	Yes	No	Yes

# Notes: Password Level Cross-chart

This chart summarizes the changes implied with the password level definition. It should also help you to understand the following chart, which details how changes from one level to the other might impact the system.

# Changing the QPWDLVL Setting

## Level 0 to level 1:

- AS/400 NetServer works no longer for Windows 95/98/ME clients
- Removes all AS/400 NetServer passwords

## Level 0/1 to level 2:

- AS/400 NetServer for Windows 95/98/ME works as long as password is 1-14 characters long
- Requires V5R1 on other AS/400s with QPWDLVL=2
- Requires to have support on other systems for passwords > 10 characters

## Upgrading to level 3:

- AS/400 NetServer works no longer for Windows 95/98/ME clients
- Removes all AS/400 NetServer passwords
- Requires V5R1 on other AS/400s with QPWDLVL=2
- Requires to have support on other systems for passwords > 10 characters

Changing the password level setting requires careful planning, since users might no longer be able to sign on to certain services.

As pointed out, level 1 and 3 will no longer support SMB for Windows 95/98/ME.

The biggest impact however happens when the level is set to 2 or 3, allowing to have long passwords and when using passphrases. Any client or service which uses password substitution will not work correctly at those levels if the client or service has not been updated to use the new password substitution or passphrase scheme. These clients or services are:

- Telnet
- Client Access
- iSeries Host Servers
- QFileSrv.400
- AS/400 NetServer Print support
- DDM
- DRDA
- SNA LU6.2

Before moving to level 2 or 3, make sure that every user id has a password for this level. Use the *Print User Profile* (PRTUSRPRF) command with option TYPE(\*PWDINFO) to obtain a list that identifies which password for which level are defined for each user id and make sure that all passwords are set for the level you want to migrate to.

It is highly recommended that the security data be saved prior to changing to a password level, making it easier to go back to a lower level.

We would **not** recommend however to change to a lower password level unless absolutely necessary.

Check with Chapter 7, *Designing Security* in the *Security Reference* manual (SC41-5302-05) to find out more.

# Notes: Changing the QPWDLVL Setting-2

Note, a new OS/400 5250 display file is used for the signon screen when the 128 character password ("pass phrase") is used. The following are the default sign on display files shipped with V5R1:

- QDSIGNON for 10 character passwords
- QDSIGNON2 for 128 character passwords

# Service Tools

**IBM @server.** For the next generation of e-business.

Service Tools (SST and DST) provide access to different functions to define or diagnose the system

Improved Security for:

- Dedicated Service Tools functions (5250, Operations Console access)
- Service Tools via the STRSST command functions (5250) and Operations Navigator access
  - Operations Navigator Disk Unit management, LPAR management

V5R1 implements:

- Service tools user profiles
- Service tools device profiles
- Service tools security data

Service tools are used to perform a wide variety of service functions. They include running diagnostics, defining new hardware components, manage DASD and LPAR configurations and many more. Most of these functions are available only via a 5250-interface.

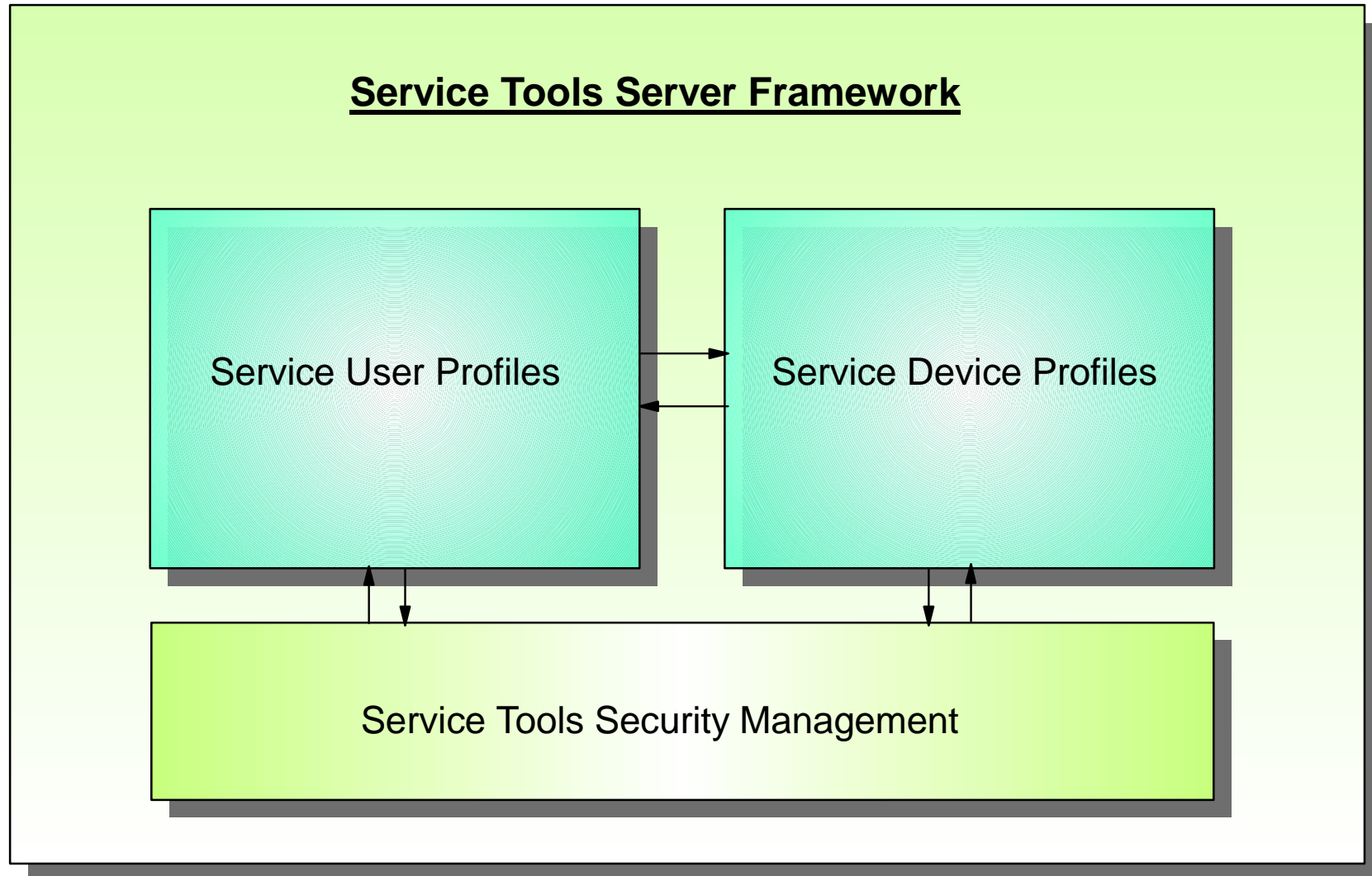
This release continues to bring out more service functions via a graphical user interface; these include functions such as LPAR management. Many of the other functions are not available once OS/400 has been started, but run in what is called "limited paging mode"( during IPL, before OS/400 is started), and therefore cannot rely upon the availability of TCP/IP communication services. This makes it difficult to make those functions available to platforms running Operations Navigator and/or Management Central which are rapidly becoming the interface by choice for running service functions.

Furthermore, once the service functions become available through a known communication protocol, such as TCP/IP, instead of via a proprietary link, such as twinax, security was enhanced to authenticate service users accessing service tools in limited paging mode.

For an overview of all new service tool Operations Navigator graphical interfaces for Disk Units and Logical Partitioning, please look at the Operations Navigator presentation.

The security settings and the definition of the Service Tools Server are addressed in the next foils.

# Service Tools Server Components



# Notes: Service Tools Server Components

Starting with V5R1 OS/400 security is enhanced to control access to service functions in Dedicated Service Tools (DST), System Service Tools (SST) and equivalent graphical interfaces function under Operations Navigator. The enhancements are summarized below and the following foils discuss the OS/400 side of configuration and setup considerations.

The Hardware presentation describes the client side of considerations, specifically focusing on new for V5R1 Operations Console with the LAN ("LAN Console").

The system is now shipped with 4 pre-configured service tool user profiles (formerly known as DST user IDs), QSECOFR, 22222222, 11111111, and QSRV. Each profile has unique privileges for using service functions. These profiles are shipped with passwords as having been expired, which means a new password must be specified on first use. Use the QSECOFR service tool user profile from DST to change these passwords.

Additional service tool user profiles may be configured with unique functional privileges from DST.

Service tool user IDs may be 10 characters in length with a case-sensitive password up to 128 characters in length.

Certain graphical user interfaces like Operations Console attached using LAN or secondary partition remote operation panel require service tool **device profiles and passwords**. Device profiles are used as a method of device authentication across the network for service tool functions.

Two easy steps:

- Configure the Service Tools Server for DST
- Configure the Service Tools Server for OS/400

DST Configuration:

- Identify the LAN adapter and the TCP/IP information if using Operations Console over the LAN

OS/400 Configuration:

- Add the Service Tools Server to the service table

Operations Navigator\* - Application Administration authorization

\*Operations Navigator access is optional)

The configuration of the Service Tools Server (STS) is a two-step operation, consisting of defining the STS to DST and to OS/400 "rules."

The DST configuration can be accessed only if the system is in DST mode. If you use the Operation Console with LAN, you do not need to reconfigure the system, since it comes preconfigured. To enable STS with its own network interface card:

- From the DST screen, select option 5 (*Work with DST environment*) and press **Enter**.
- From the *Work with DST environment*, select 2 (*System devices*) and press **Enter**. The *Work with System Devices* screen appears.
- From the *Work with System Devices*, select 6 (*Console mode*) and press **Enter**. The *Select Console Type* screen appears.
- From the *Select Console Type* screen, press **F11** (*Configure*), which allows you to enter the LAN adapter and TCP/IP information on the *Configure Service Tools Adapter* screen. Once this information has been entered, press **F7** (*Store*) to commit your changes and **F14** (*Activate*) to activate the adapter.

You must then add the STS to the service table in order to access DST functions (Operations Console functions) and the SST functions over the LAN using TCP/IP.

To do this:

- Use the Add Service Table Entry (ADDSRVTBLE) command to add the following parameters:  
ADDSRVTBLE SERVICE ('as-sts')  
    PORT(3000)  
    PROTOCOL('tcp')  
    TEXT('Service Tools Service')  
    ALIAS('AS-STs')
- End TCP (ENDTCP) and then Start TCP (STRTCP)

## Configuring STS continued

- Verify that the Service Tools Server (as-sts) is listening to port 3000. The easiest way to do this is to use the OS/400 NETSTAT command with OPTION(\*CNN). Scroll down until you see the as-stst entry with a State of "Listen."

Opt	Remote Address	Remote Port	Local Port	Idle Time	State
—	*	*	as-sts	040:58:20	Listen

- If you are going to use Operations Navigator interfaces to Disk Units, Logical Partitioning, and Cluster Management to a specific system, in addition to having a Service Tools user profile to sign on with and the Service Tools server listening on Port 3000 you need to explicitly enable corresponding functions on the AS/400 or iSeries system.

On the same system, under Operations Navigator, right click on the system name. Select Application Administration. Click OK on the first window brought up. This will bring up another window. Select the third tab (Host Applications). Expand Operating System/400; expand Service, and ensure Disk Units is checked, and Logical Partition entries are checked, depending on which functions you want Operations Navigator user who has a Service Tools Security user id and password to be able to perform. Select OK.

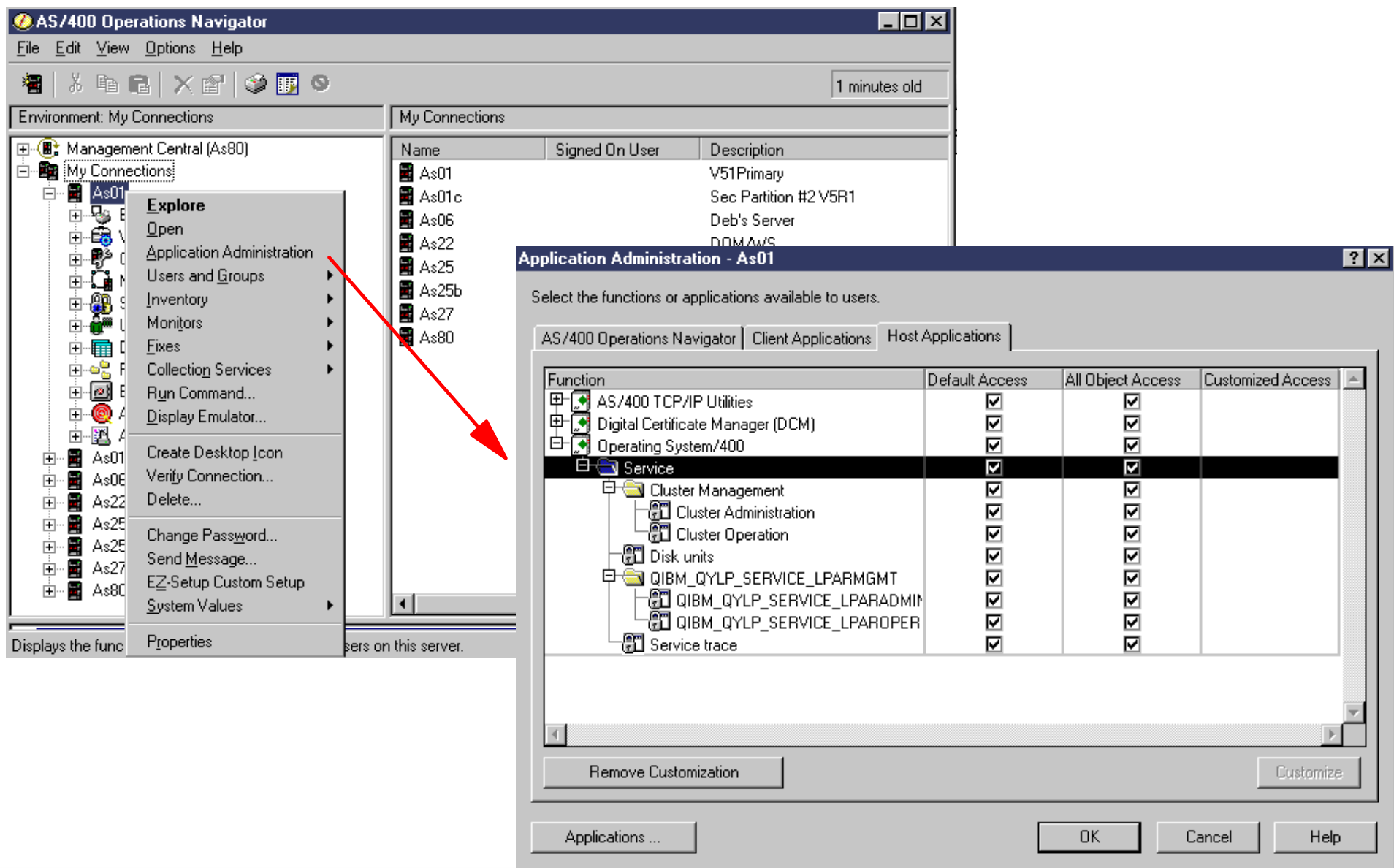
You will only ever have to do all these steps, once per system.

The next foil shows an example of the expanded Application Administration - Service window just discussed.

Following foils give more details on the host server configuration of Service Tools user profiles and Device profiles.

The Hardware presentation shows the required corresponding Client Access configuration for Operations Console over the LAN.

# Client Access Application Administration



The screenshot displays the AS/400 Operations Navigator interface. The main window shows a tree view of connections under 'My Connections' and a table of connection details. A context menu is open over the 'As01' connection, with a red arrow pointing to the 'Application Administration' option. A secondary dialog box, 'Application Administration - As01', is open, showing a table of functions and their access levels.

Function	Default Access	All Object Access	Customized Access
AS/400 TCP/IP Utilities	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Digital Certificate Manager (DCM)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Operating System/400	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Service	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cluster Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cluster Administration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cluster Operation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Disk units	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
QIBM_QYLP_SERVICE_LPARGMT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
QIBM_QYLP_SERVICE_LPADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
QIBM_QYLP_SERVICE_LPAPER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Service trace	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

**Note:** Cluster Management is shown as it must be "enabled" if configuration and management is to be performed through Operations Navigator. At the time this presentation was created, signing on to the Service Tools server security was not required for cluster management through Operations Navigator.

Clicking either Disk Units or Logical Partitioning branches in Operations Navigator will require successful signing on to the Service Tools before proceeding.

V5R1 limits access to service tools to:

- Specific user profiles
- Specific functions
- Up to a maximum of 96 users

```
Start Service Tools (STRSST) Sign On
                                     SYSTEM: M01
Type choice, press Enter.
Service tools user . . . . .
Service tools password . . .
F3=Exit      F9=Change Password    F12=Cancel
```

Before V5R1 any authorized user could access any service tool on an iSeries, provided they had \*SERVICE authority. With this type of authority, all service tools could be accessed, and no log traces were available to monitor changes which were implemented using these interfaces.

V5R1 allows you to create user-defined profiles that you can grant functional privileges to specific tools, or to a group of tools. Up to 96 different profiles can be created.

The IBM supplied profiles (QSECOFR and 2222222) now also contain a QSRV profile, with about the same privileges as 22222222, minus the Display/Alter/Dump capability, have their password set to expired. You are advised to change immediately the passwords for these profiles.

The prompt for service tools user profile and password will also appear when an Operations Navigator user attempts to use Cluster Management, Disk Units, or Logical Partitioning functions (provided Client Access Application Administration has authorized these functions to be used by the OS/400 user profile signed on to Operations Navigator).

# DST Security - Logical Concept

## User Level

- ID, password
- Privileges

DST User ID	DST Password	Functional Privileges
11111111	11111111	Basic
22222222	22222222	Limited
QSRV	QSRV	Service
QSECOFR	QSECOFR	Security Officer
* user id	xxxxxxx	specific grant/revoke


## Device Level

- ID, password
- Privileges

Device ID	Device Password	Functional Privileges
QCONSOLE	QCONSOLE	console + panel
* user device name	xxxxxxx	yyyyyyyyy

\*Can be added by OS/400 security administrator

Operative	Device Auth.	Device Privileges	User Auth.	User Privileges	Other
Time	No	n/a	No	n/a	No
LPAR	No	n/a	Yes	n/a	Yes
Virtual Control Panel	Yes	panel x	No	LPAR	No

**IBM**  server. For the next generation of e-business.

This foil represents the logical concept of Service Tools user profile privileges (authorizations) and device profile privileges (authorizations).

As you can see, OS/400 is shipped with 4 DST Service Tools user profiles and associated privileges. Likewise OS/400 is shipped with a device profile QCONSOLE.

In this example we highlight the capability to add additional Service Tools user profiles and/or device profiles. We recommend defining at least one additional user profile for service tools with authorities identical to QSECOFR. If you are also using Operations Console over the LAN, we also recommend creating one additional device profile with authorities identical to QCONSOLE. This gives you at least one backup should one of these profiles become disabled.

When you are familiar with this service tools profile and, where appropriate, the device profile support, you can use different profiles with different privileges if you wish to.

The table in the lower portion of the foil is a logical representation of the integration of a Service Tools user profile and its associated privileges and device profile with its associated privileges.

This presentation focuses on the host (AS/400, iSeries side of configuration). The Hardware presentation describes Operations Console over the LAN client workstation configuration.

The Operations Navigator presentation uses the Service Tools user profile in its Disk Unit and LPAR examples.

The next foil expands on Service Tools device profiles.

## LAN based Operations Console

### Associate functions with devices

- Authentication services for each device
- Up to 50 device profiles

V5R1 introduces the notion of the LAN based Operations Console, allowing to use a device on a LAN to act as a console for a single system or for a number of systems - LPAR'ed or not. This introduction removes the limitation of the physical proximity to a system and allows a more flexible console assignment for your operations, although in some cases the proximity of the system may still be a requirement, e.g. for loading a CD in the CD reader.

***The LAN based Operations Console is the only configuration that can use the service tools device user profile.***

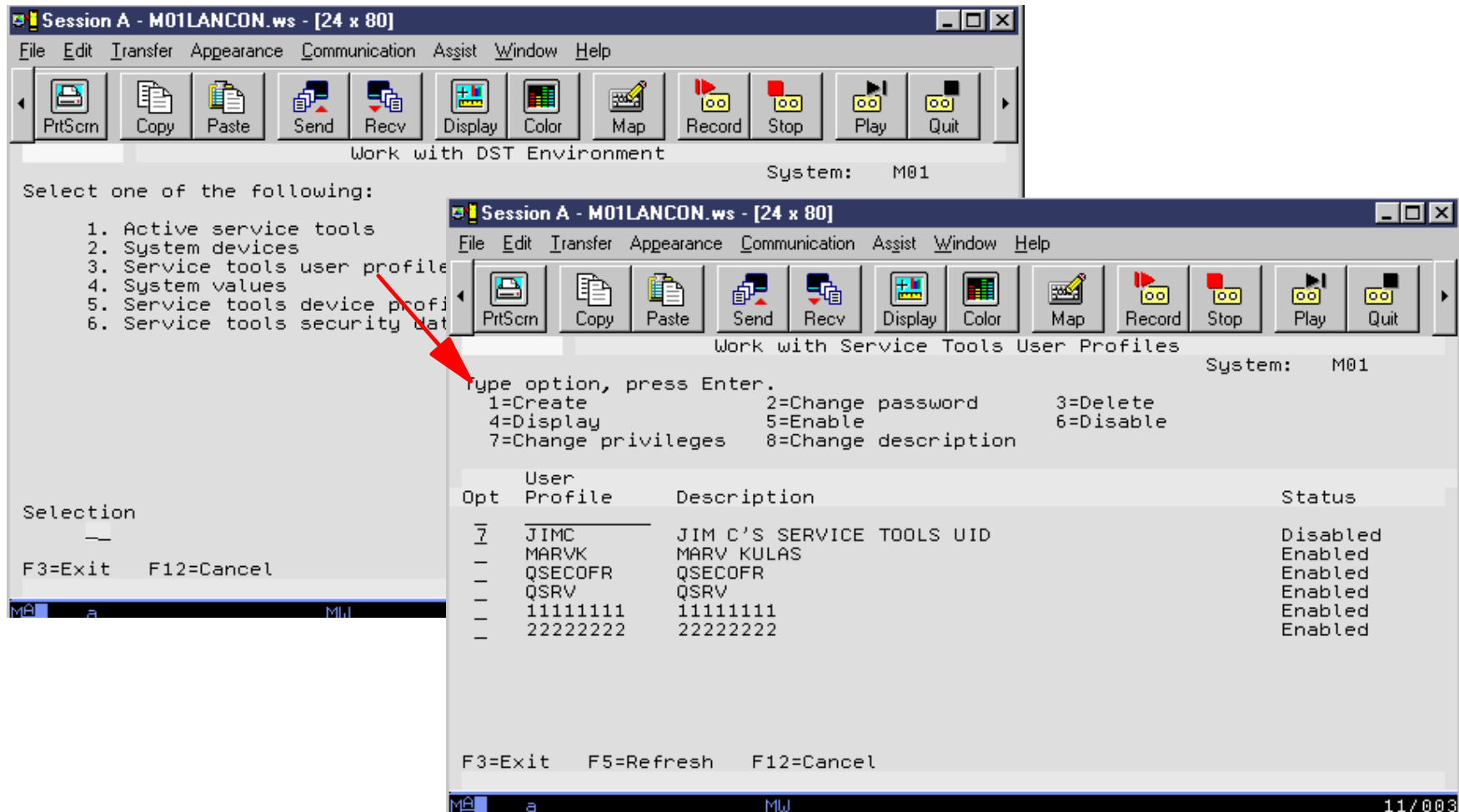
It is allowed to have up to 50 device profiles.

Note: Operations Console for the LAN requires both **general Service Tools user profiles** and **device profiles**.

# Setting up Service Tools user profile example

From the DST environment, select "Service tools user profiles:"

- Create an STS user profile, enable/disable privileges



The image shows two overlapping screenshots of a terminal window titled "Session A - M01LANCON.ws - [24 x 80]".

The top screenshot shows the main menu with the following options:

1. Active service tools
2. System devices
3. Service tools user profile
4. System values
5. Service tools device profile
6. Service tools security data

A red arrow points from option 3 to the second screenshot.

The bottom screenshot shows the "Work with Service Tools User Profiles" screen. It displays a list of user profiles with the following columns: Opt, User Profile, Description, and Status.

Opt	User Profile	Description	Status
7	JIMC	JIM C'S SERVICE TOOLS UID	Disabled
-	MARVK	MARV KULAS	Enabled
-	QSECOFR	QSECOFR	Enabled
-	QSRV	QSRV	Enabled
-	11111111	11111111	Enabled
-	22222222	22222222	Enabled

At the bottom of the terminal window, the text "11/003" is visible.

To set up the general service tools user profiles and the privileges (authorities), select the Service tools user profiles, option 3 on the left window shown in this foil. That brings up the right window shown.

In this example, we have already added JIMC by previously using the 1=Create function. Here we show 7=Change privileges to lead to the next foil.

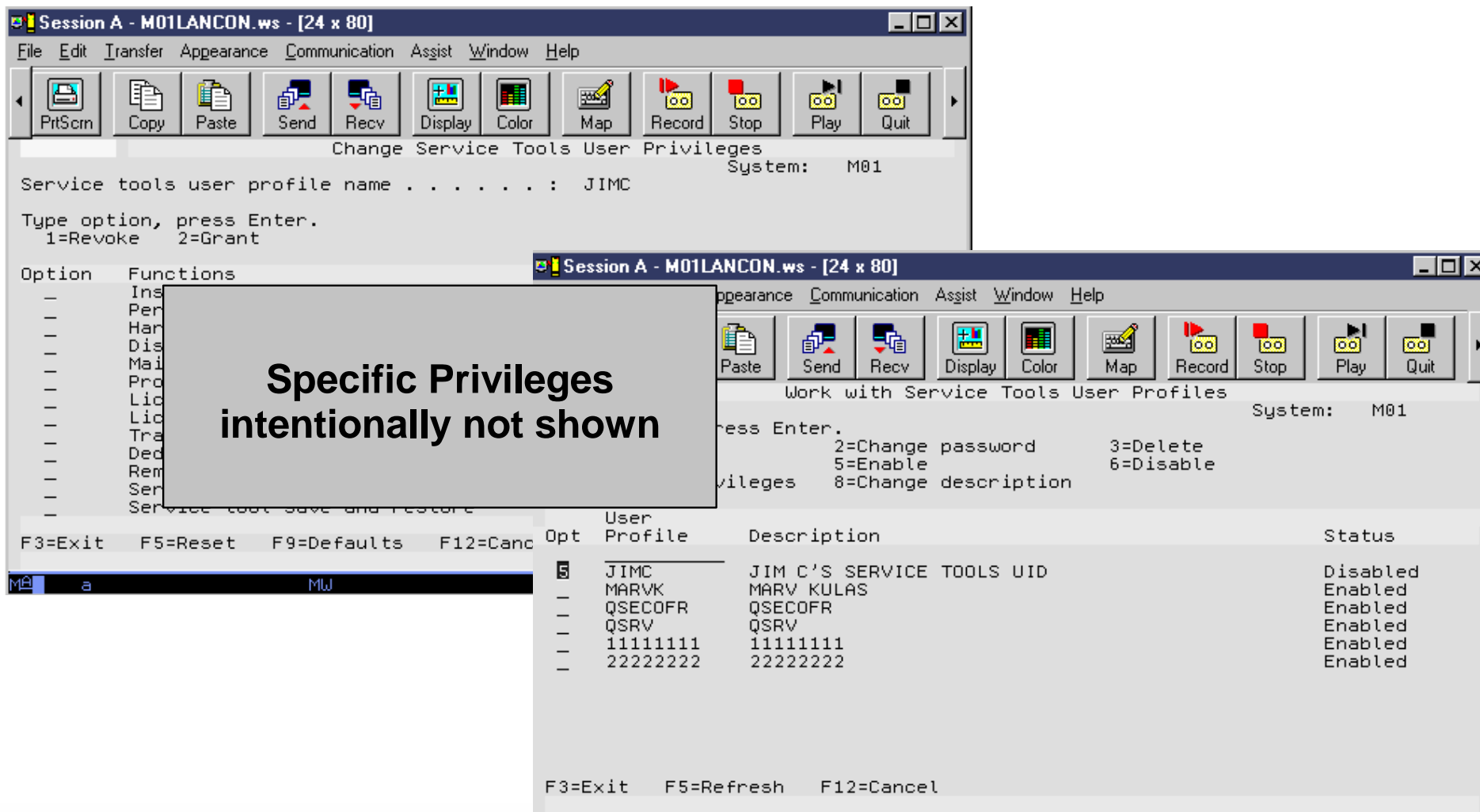
Note: We are not showing the steps required to set up device profiles (5 - Service tools **device profiles**) and security data (6-Service tools security data) on the left window. Configuring a device profile is similar to configuring a Service Tools user profile.

See the Hardware presentation for more details on the client workstation configuration for Operations Console over the LAN.

# Setting up Service Tools user profile example

From the DST environment - Change Service Tools User Privileges:

- Enable/disable specific privileges, enable Service Tools user profile



Change Service Tools User Privileges  
System: M01  
Service tools user profile name . . . . . : JIMC  
Type option, press Enter.  
1=Revoke 2=Grant

Option Functions

Work with Service Tools User Profiles  
System: M01  
press Enter.  
2=Change password 3=Delete  
5=Enable 6=Disable  
Privileges 8=Change description

Opt	User Profile	Description	Status
5	JIMC	JIM C'S SERVICE TOOLS UID	Disabled
-	MARVK	MARV KULAS	Enabled
-	QSECOFR	QSECOFR	Enabled
-	QSRV	QSRV	Enabled
-	11111111	11111111	Enabled
-	22222222	22222222	Enabled

Specific Privileges intentionally not shown

Monitor (log) effectiveness of security implementation:

- View entries in the logged security data
- Save and restore security data

Modify some password and OS security settings:

- Reset the QSECOFR to the default password
- Change the OS install security
- Change the password level

Once the profiles have been created, you can monitor how the service tools are being used by viewing the service tools security logs. In this log, all events, like a profile being created, modified or deleted, privileges being granted or revoked, or actions like the creation and handling of the logs, are being recorded.

New in V5R1 is also that the install of the operating system can be secured. This means that, if a OS install is secured, only users with full DST privileges will be able to perform an OS install. If the OS install is not secured, all users who can perform an IPL can also perform an OS install.

For more details on Service Tools security and Device Profile security, refer to :

- *V5R1 Tips and Tools for Securing your iSeries*, SC41-5300-05
- V5R1 Technical Overview presentation - Hardware

# Integrated File System Enhancements

Journaling of byte stream files and directories

Text file I/O conversion between CCSID with characters of differing lengths

PC file attributes can be manipulated by iSeries command or API

CPY command now supports a subtree

New objects access support

- Pipes, FIFOs, device/null character special file
- Stream I/O to savefile

Deadlock detection

Byte stream files can be Memory Mapped

Byte stream objects can be signed

User Defined File System (UDFS) can be mounted to an Independent Auxiliary Storage Pool (IASP)

IFS enhancements include:

- Byte stream files and directories can now be journaled - allows a customer to use third party software to provide replication on another iSeries server or to utilize the journaled information for other recovery and monitoring purposes.
- Text file I/O now supports converting between CCSIDs with characters differing lengths. Previously, it only supported single-byte characters to single-byte characters and double-byte characters to double-byte characters. A set of interfaces is included that allows you to specify pathnames in any CCSID.
- PC File attributes, such as Read-Only, can now be manipulated on the iSeries through a command/API interface. This means you can now manage files created by PCs even if they are marked Read-Only
- The CPY command is enhanced to allow the specification of a subtree. This gives you the ability to copy whole subtrees on the iSeries without using an interactive interface or programming the function yourself.
- Three new objects are supported along with stream I/O support for an existing object. Pipes and FIFOs are new with this release and provide program to program communication through file system objects. The third new object is the dev/null character special file. dev/null is a special file that can be written to forever, but is always empty when read. Many applications take advantage of dev/null to discard output from subapplications without having to change the subapplication.  
Stream I/O is now supported to savefiles. This allows you to extract the contents of a savefile through all sorts of methods, transport it through the network using stream file protocols and then place it back into another savefile
- File system APIs support parameters and buffers in Teraspace for large I/Os.
- Deadlock detection to help diagnose applications with conflicting lock ordering
- Byte stream file objects can now be signed to ensure they have not been modified. This would typically be used for objects that are transmitted over the internet. The support provides a higher level of object integrity.
- Internal performance trace points have been added as an aid in managing IFS performance.

IFS enhancements continued:

- Ability to define an Independent Auxiliary Storage Pool (IASP) that can be moved from one iSeries server to another. User Defined File Systems (UDFS) can be created in these IASPs. When the IASP is moved from one system to another, the UDFS on the IASP can then be mounted on the new system and made available to applications and users. See the Availability and Operations Navigator presentations for additional information.

Note: For V5R1 a UDFS can be mounted to an IASP. However, the journaling of byte stream files and directories in an IASP is not supported.

V5R1 iSeries can operate as the Logon Server for Windows clients

Reduces OS/400 user profile disablement due to Windows application invalid logons

New NetServer Setup wizard

Supports access to IFS file greater than 2 GB

Printer shares can be published to Directory Services (LDAP) directory

# Notes: NetServer Enhancements

AS/400 NetServer is enhanced so that the iSeries can operate as the Logon Server for Windows clients. The iSeries can be used to authenticate logging onto Windows, provide the home directory, and logon scripts to the Windows user. Additionally, Windows user profiles including Desktop, Start Menu, Favorites, and policies can be stored and retrieved from an iSeries server. A Windows NT or Windows 2000 server is no longer needed in the network to provide these functions.

AS/400 NetServer dramatically reduces the number of times that OS/400 user profiles become disabled due to Windows programmatically attempting invalid signons to access the OS/400 without compromising security.

Additionally, when users do cause their user profiles to become disabled due to several attempts with different invalid passwords, AS/400 NetServer provides new GUI support through a Disabled User IDs menu item off the AS/400 NetServer menu of Operations Navigator to re-enable those user profiles. This support has also been made available through an API on OS/400. These changes can reduce the number of times that user profiles become disabled and improve the ease with which disabled users can be managed.

The OS/400 enhanced password length and allowable characters in V5R1 support is more compatible with Windows operating system password support. length of a password to be more compatible with Windows. This helps customers who like to have their Windows and iSeries passwords match. AS/400 NetServer also provides support for the NTLMV2 password hash that the Windows PCs can be configured to use to provide better password protection on the network.

User IDs longer than 10 characters are now truncated to 10 characters when checking for an iSeries user id instead of being rejected. Now a userid such as Administrator on Windows would be the same as ADMINISTRA on the iSeries. This should help compatibility between Windows and iSeries user IDs.

.A new AS/400 NetServer Setup Wizard is now part of Operations Navigator that guides you through setting up your AS/400 NetServer based on the type of Client Access clients being used. This new Setup Wizard also helps the user configure logon support.

AS/400 NetServer now supports access to files larger than 2 GB in the Integrated File System.

Through Operations Navigator and APIs, a new Session Identifier can be used to allow better management and tracking of AS/400 NetServer Sessions. This is extremely important in a Windows Terminal Serving environment when many users have sessions through a single Windows system. Now sessions can be ended or properties observed on single sessions rather than all the sessions coming from a single system.

NT Background services can now access the AS/400 NetServer without user intervention.

Printer Shares can now be published in Directory Services (LDAP) for use by Windows 2000 systems using Active Directory to find printers.

Also, various performance and scalability improvements help customers consolidate file and print serving on an iSeries server.

# Miscellaneous Changes

**IBM @server.** For the next generation of e-business.

# Increased Maximum Capacities

- Jobs on the system tripled: up to 480,000 jobs (includes spool files ready for print)
  - Set via new system value QMAXJOB
- Spool files per job increased 100 times: up to 999,999 spool files
  - Set via new system value QMAXSPLF
- Libraries in the user portion of the library list increased 10 times: up to 250 libraries
  - Could impact existing programmers who retrieve this information (see memo to users)
- User profiles that can be saved tripled: up to 340,000 profiles
- Private authorities a user profile can have to be saved increased 25 times to support up to five million private authorities
- Database physical file size doubled: up to a 1 terabyte physical file
- Database LOBs (BLOB, CLOB, DBLOB) can now grow up to 2 GB
- Journal entry maximum length is 4 GB
- Save System (SAVSYS) and Save Security Data limits removed

Complete list of limits and changes for V5R1 can be found under the Technical Reference topic on the iSeries Technical Studio web site at <http://www.ibm.com/eserver/iseries/tstudio>.

## Library List enhancements

- V5R1 ships with number of entries on user library list limited to 25
- May be extended to maximum number of 250
- Number of entries in system library list remains at 15
- \*SHRRD lock for library entries in library list can be turned off

The number of entries in the user library list portion of an active job can be extended from its previous value of 25 to a maximum of 250. To achieve this, you need to delete (or rename) a data area QLILMTLIBT in library QUSRSYS (see next foil). Once this data area has been deleted, any new job will be able to allocate a maximum of 265 libraries in the library list: 15 (unchanged from the previous releases) in the system library list portion (and defined system wide via the system value QSYSLIBL, changeable on job level via the *Change System Library List* (CHGSYSLIBL) command), and 250 in the user library list portion.

At the same time, a system value called *Library Locking Level* (QLIBLCKLVL) is introduced that allows to set Work Management so that it does not create a \*SHRRD lock on every library which is in an active job's user portion of library list. This improves start up time of a job which has a huge amount of libraries in its library list, since the creation of locks will be minimized. It however will no longer prevent to delete or clear a library that is in a library list with this setting turned off. Access to any object during such a delete library attempt will still prevent the library to be deleted as before; however, if no single object is locked during the delete library command, the library will be deleted from the system and the entry in the library list will show up without an identifier but with an entry description set to \*DELETED. Changing this setting from one state to the other does not effect any active jobs: only newly created jobs will have this change in effect.

# Notes: Increased Maximum Capacities-2

This is the output of the data area QLILMTLIBL:

```
Display Data Area

Data area . . . . . : QLILMTLIBL
  Library . . . . . :   QUSRSYS
  Type . . . . . :   *CHAR
  Length . . . . . :   2000
  Text . . . . . :   LIMIT USER LIBRARY LIST TO 25

Value
Offset  *...+....1....+....2....+....3....+....4....+....5
   0    'THE EXISTENCE OF THIS DATA AREA LIMITS THE NUMBER '
  50    'OF LIBRARIES IN THE USER PART OF THE LIBRARY SEARC'
 100    'H LIST TO 25 FOR ALL JOBS ON THE SYSTEM.  DELETING'
 150    ' THIS DATA AREA ALLOWS ALL JOBS THAT BECOME ACTIVE'
 200    ' AFTER THE DELETION TO HAVE 250 LIBRARIES IN THE U'
 250    'SER PART OF THE LIBRARY SEARCH LIST.                '
 300    '                                                         '
 350    '                                                         '
 400    '                                                         '
```

# Notes: Increased Maximum Capacities-3

Spin-offs in several areas for the teraspace enablement:

- Database LOBs (BLOB, CLOB, DBLOB) can now grow up to 2 GB
- Journal entry maximum length is 4 GB
- SLIC trace data and error log grow to 4 GB per segment
- Removes another set of potential limits

Maximum number of jobs is now 480 000 (versus 163 520)

- Set via new system value QMAXJOB
- Allows 30 tables of 16 352 jobs each + IPL space

Maximum number of spool files per job increased to 999 999

V5R1 continued with removing the few limits still left in memory addressing by lifting the limit to database large objects (Character Large Objects, Binary Large Objects and Double Byte Large Objects) from its original 15 MG to 2 GB. At the same time, the maximum length for a single journal entry was increased to 4 GB.

It is also possible in V5R1 to use 4 GB segments for SLIC trace data and error log entries.

The maximum number of jobs within a system (i.e. active, on jobqueue or terminated with spool file entries) is increased to a maximum of 490 560), which is actually 30 tables of 16 352 job structure entries. To avoid all entries to be used and thus causing abnormal system terminations, a new system value is introduced, *Maximum number of Jobs* (QMAXJOB), that allows you to specify a maximum of up to 485 000 jobs. Once this threshold is reached, the system will still be able to resume work after an IPL and a clean up of job entries that are no longer needed.

The QMAXJOB system value is shipped with an initial value of 163 520, which is the maximum number of jobs on earlier releases.

The new *Maximum spooled files per job* (QMAXSPLF) system value allows to specify a significantly higher number of spooled files per job. If the number is decreased and if a job has a number which is higher than the new maximum, the spooled files are not deleted.

# Tape Support Enhancements

- The Set Tape Category (SETTAPCGY) command is enhanced to allow multiple categories to be mounted on a tape library with multiple devices. This enables multiple jobs to be using mounted categories on the same tape library.
- Application programs that use tape devices will now be able to use all of the formats and densities supported by OS/400 for the tape device. This enhancement allows applications to create tapes with any tape format or density supported for the tape device.
- If the door to a tape library is opened during a multi-volume save or restore operation, the operation will no longer fail when it is time to load the next cartridge. The save or restore operation will now be suspended for 10 minutes to give the operator time to make the tape library ready by closing the door, and the save or restore operation will not be canceled unless the operator replies with a Cancel to an inquiry message sent after the 10 minute wait. This change reduces the chance that a long running save or restore operation will be ended before it completes.

# Optical Support Enhancements

In V5R1, the following enhancements are provided to support Optical Removable Media (Magneto Optical (MO), Write Once Read Many (WORM), CD-ROM and DVD-RAM):

- Enhanced data caching when processing files on optical volumes formatted with Universal Disk Format (UDF). This will improve performance and functionality while at the same time reducing overhead.
- Improved performance for incremental copies using Copy Optical (CPYOPT)
- Improved reliability and performance of Duplicate Optical (DUPOPT) utility for optical volumes formatted with High Performance Optical File System (HPOFS)
- :Improved file creation performance
- Allow Remove Optical Cartridge (RMVOPTCTG) for optical volumes in CD-ROM and DVD-RAM devices
- Increase the maximum optical file size accessible through HFS or IFS to 4,294,705,152 bytes (4 GB - 256 KB)
- Usability enhancements to Initialize Optical (INZOPT)

A new job trace facility introduces several changes needed in TRCJOB. The new job trace support should be used instead of TRCJOB. The new job trace facility includes four new CL commands (STRTRC, ENDTRC, PRTTRC, and DLTTRC) and includes the following enhancements:

- Significantly improved performance, particularly when tracing ILE programs
- Improved reporting information, including finer time granularity and additional I/O information
- Eliminates the need to STRSRVJOB on jobs to trace
- Can trace multiple jobs with one command, including a generic job set
- Increased buffer size for collecting trace information to 4 GB.

# Program observability and restore

IBM  server iSeries

On V5R1, using the Remove Observability function now retains sufficient information to retranslate that program, if necessary, when restored

**IBM**  server. For the next generation of e-business.

# Key Memo to Users Topics

Read this first chapter (before installing V5R1)

Exchanging journal receivers with V4R4 or V4R5 requires PTFs to V4R4, V4R5

Audit Journal file will contain object path names in the entries

Unicode and CCSID data conversion tips

Communicating with OS/400 V4R4 requires PTFs

Increased upper limit of 250 libraries in job library list may impact applications currently retrieving library list information via commands or APIs

Program observability changes

OS/400 and other licensed IBM programs now come "digitally signed"

Larger maximum password length impacts

Start/End Performance Monitor support removed

Java Development Kit Java release level and default authority changes

TCP/IP IP filtering and Network Address Translation(NAT) enhancements

Required security changes for Dedicated Service Tools and System Service Tool function

# Key Memo to Users Topics -2

\*BASE Network Server Description configuration objects from earlier than V51 cannot be used.

Directory Servis (LDAP) option 32 no longer requires installation. Functions now run in QSYS....

Numerous small changes made to TCP Connectivity Utilities for iSeries.

Client Access changes

# IPL Attributes

## *New Start TCP/IP on IPL Attributes*

```
                                Display IPL Attributes
                                System:      AS25B

System Features:
Restart type . . . . . : *SYS
Keylock position . . . . . : *NORMAL

Hardware Functions:
Hardware diagnostics . . . . . : *MIN

Operating System Functions:
Compress job tables . . . . . : *NONE
Check job tables . . . . . : *ABNORMAL
Rebuild product directory . . . . . : *NONE
Mail Server Framework recovery . . . . . : *NONE
Display status . . . . . : *ALL
Start TCP/IP . . . . . : *YES
```

# General V5R1 Requirement Summary

V5R1 runs on all AS/400 and iSeries systems

V5R1 requires a minimum main memory size of 128 MB and a recommended minimum disk size of 8 GB

OS/400 V5R1 requires more free disk space than previous releases as follows:

- Additional 90 MB for installation than installation of V4R5 required
- Additional 270 MB for installation than installation of V4R4 required
- Total disk space required for OS/400 in the range of 350 MB to 1 GB

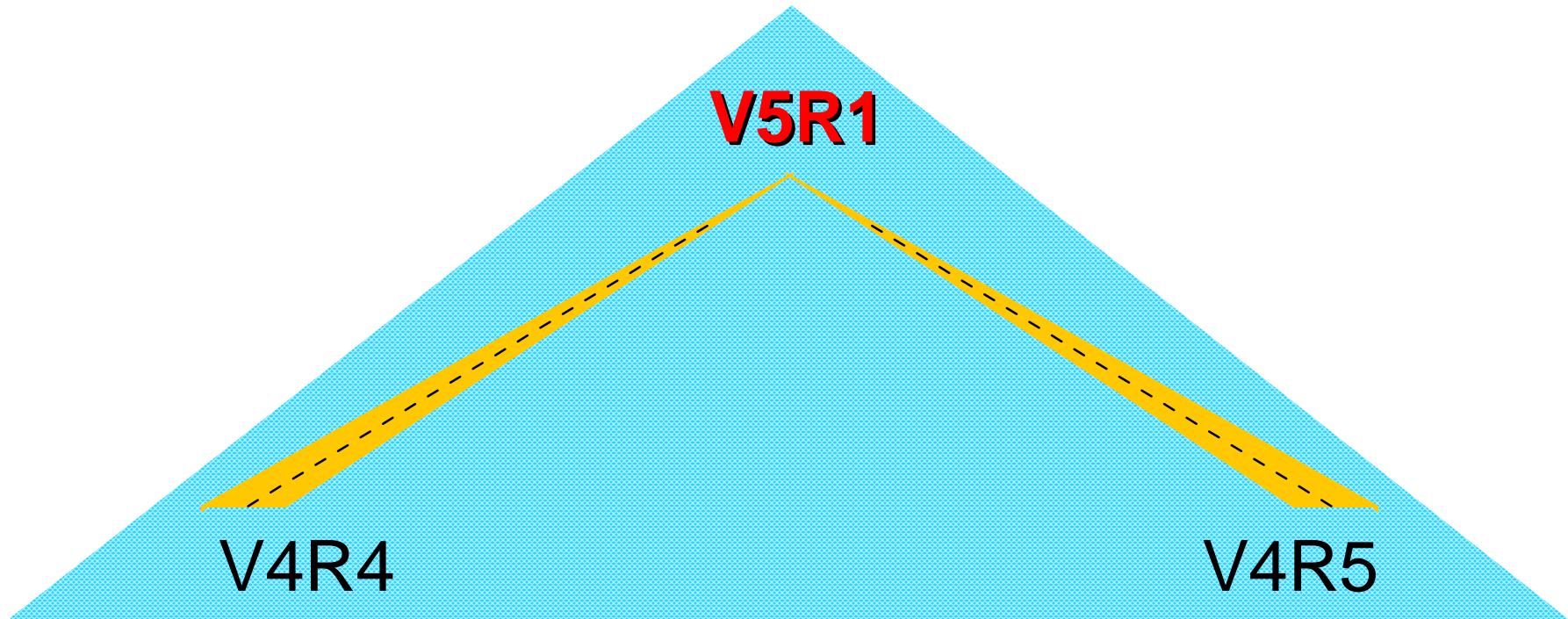
# Installing V5R1

**IBM @server.** For the next generation of e-business.

# Release-to-Release Compatibility

Single Step Upgrade from:

Interoperate, save/restore with:



**IBM**  server. For the next generation of e-business.

## Supported releases:

- V4R4
- V4R5

## Support was added to specify TGTRLS(\*V4R4M0) to save:

- Cluster Resource Groups
- Media Definitions
- Management Collection objects
- SQL User Defined Types

# Notes: Target Release Support

Although upward compatibility is still maintained in V5R1, the target release parameter supported on the SAVE and CREATE commands that support a target release, is now limited to N-2 releases, meaning that objects can be created or saved for V4R4, V4R5 and V5R1. This implies that no support is provided any longer to create or save objects for any CISC release (V3R2M0) as was the case in V4R5.

If you use to *Save Save File Data* (SAVSAVFDTA) command to save save files, created on a system with target release support to versions earlier than V4R4, but residing on a V5R1 system, the output on tape will be determined by the version level used to create the save file, so that it can be distributed to earlier systems.

Check with the *Availability Enhancements* section of this presentation to find out more about the Save/Restore capabilities of the iSeries.

# Migration to V5R1

## Releases allowing slip-install:

- V4R4
- V4R5

## Version level change includes:

- New Product Identifiers for the OS/400 and Licensed Programs (5769 becomes 5722)
- Repackaging of Licensed Products
- List of Licensed Products which are no longer supported
- Software problem identification changes

You can install V5R1 over an existing V4R3, V4R4 or V4R5 Operating System. As always, best practice guidelines for installing an new version/release do apply, so please read carefully the documentation provided with this release (*Read this First, Software Installation Guide*, etc.) before you engage into an upgrade of your current system. We also recommend that performance data must be collected before the upgrade is performed, so that you can compare it with performance data gathered after the upgrade.

This new version repackaged and renamed both OS/400 and all refreshed or new Licensed Programs from the 5769 to the 5722 prefix. Licensed Products that are "skip ship" (i.e. no changes since the last release) maintain the 5769 prefix. On the next pages you will find a full overview of all product changes and/or additions from V4R5 to V5R1. This list applies only to those products which are the base Licensed Products as shipped with the stamped media.

For V5R1 software problems:

- Software APAR prefix changed from SA to SE.
- Software fix prefix changes from SF to SI
- No changes to MAccc SLIC APARS or MFccc SLIC PTFs.
- Reserved PTF numbers (SF99nnn, SF97nnn, SF96nnn, and SF95nnn) remain unchanged.

# Notes 2: Migration to V5R1

Product	Option	Status	Release	Description
5722-SS1	Base	RFR	V5R1M0	OS/400 - base
5722-SS1	1	RFR	V5R1M0	OS/400 - Extended Base Support
5722-SS1	2	RFR	V5R1M0	OS/400 - Online Information
5722-SS1	3	RFR	V5R1M0	OS/400 - Extended Base Directory Support
5722-SS1	4	RFR	V5R1M0	OS/400 - S/36 and S/38 Migration
5722-SS1	5	RFR	V5R1M0	OS/400 - System/36 Environment
5722-SS1	6	RFR	V5R1M0	OS/400 - System/38 Environment
5722-SS1	7	RFR	V5R1M0	OS/400 - Example Tools Library
5722-SS1	8	RFR	V5R1M0	OS/400 - AFP Compatibility Fonts
5722-SS1	9	RFR	V5R1M0	OS/400 - *PRV CL Compiler Support
5722-SS1	11	RFR	V5R1M0	OS/400 - S/36 Migration Assistant
5722-SS1	12	RFR	V5R1M0	OS/400 - Host Servers
5722-SS1	13	RFR	V5R1M0	OS/400 - System Openness Includes
5722-SS1	14	RFR	V5R1M0	OS/400 - GDDM
5769-SS1	15	DLT		OS/400 - Common Programming APIs Toolkit
5722-SS1	16	RFR	V5R1M0	OS/400 - Ultimedia System Facilities
5769-SS1	17	DLT		OS/400 - PSF/400 Fax Support
5722-SS1	18	RFR	V5R1M0	OS/400 - Media and Storage Extensions
5722-SS1	21	RFR	V5R1M0	OS/400 - Extended NLS Support

RFR = Refresh DLT = Deleted

**IBM**  server. For the next generation of e-business.

# Notes 3: Migration to V5R1

Product	Option	Status	Release	Description
5722-SS1	22	RFR	V5R1M0	OS/400 - ObjectConnect
5722-SS1	23	RFR	V5R1M0	OS/400 - OptiConnect
5722-SS1	25	RFR	V5R1M0	OS/400 - NetWare Enhanced Integration
5722-SS1	26	RFR	V5R1M0	OS/400 - DB2 Symmetric Multiprocessing
5722-SS1	27	RFR	V5R1M0	OS/400 - DB2 Multisystem
5722-SS1	30	RFR	V5R1M0	OS/400 - Qshell Interpreter
5722-SS1	31	RFR	V5R1M0	OS/400 - Domain Name System
5722-SS1	32	RFR	V5R1M0	OS/400 - Directory Services
5722-SS1	33	RFR	V5R1M0	OS/400 - Portable Application Solutions Environment
5722-SS1	34	RFR	V5R1M0	OS/400 - Digital Certificate Manager
5722-SS1	35	RFR	V5R1M0	OS/400 - Cryptographic Service Provider
5722-SS1	36	RFR	V5R1M0	OS/400 - PSF/400 1-20 IPM Printer Support
5722-SS1	37	RFR	V5R1M0	OS/400 - PSF/400 1-45 IPM Printer Support
5722-SS1	38	RFR	V5R1M0	OS/400 - PSF/400 Any Speed Printer Support
5722-SS1	39	NEW	V5R1M0	OS/400 - International Components for Unicode
5722-SS1	40	NEW	V5R1M0	OS/400 - PSF/400 One Printer Only, 1-45 IPM
5722-SS1	41	NEW	V5R1M0	OS/400 - HA Switchable Resource
5722-AC3	Base	RFR	V5R1M0	Crypto Access Provider 128-bit for AS/400
5769-AS1	Base	DLT	V5R1M0	AFP UtilitiesWebSphere Application Server for AS/400

RFR = Refresh DLT = Deleted

**IBM**  server. For the next generation of e-business.

# Notes 4: Migration to V5R1

Product	Option	Status	Release	Description
5722-AP1	Base	RFR	V5R1M0	Advanced DBCS Printer Support for AS/400
5722-AP1	1	RFR	V5R1M0	Adv DBCS Printer Support for AS/400 - IPDS
5722-BR1	Base	RFR	V5R1M0	Backup Recovery and Media Services
5722-BR1	1	RFR	V5R1M0	BRMS/400 - Network Feature
5722-BR1	2	RFR	V5R1M0	BRMS/400 - Advanced Function Feature
5722-CE2	Base	RFR	V5R1M0	Client Encryption 56-bit
5722-CE1	Base	RFR	V5R1M0	Client Encryption 128-bit
5722-CM1	Base	RFR	V5R1M0	Communications Utilities
5722-DE1	Base	New	V5R1M0	DB2 UDB Extenders
5722-DE1	1	New	V5R1M0	DB2 UDB Text Extenders
5722-DE1	2	New	V5R1M0	DB2 UDB XML Extenders
5722-DE1	3	New	V5R1M0	Text Search Engine
5722-DG1	Base	RFR	V5R1M0	IBM HTTP Server
5722-DG1	1	New	V5R1M0	Triggered Cache Manager
5769-DP3	Base	Skip	V5R1M0	DB2 DataPropagator 7.1
5722-IP1	Base	New	V5R1M0	IBM Infoprint Server
5722-JC1	Base	RFR	V5R1M0	Toolbox for Java
5722-JS1	Base	RFR	V5R1M0	Job Scheduler
5722-JV1	Base	RFR	V5R1M0	Developer Kit for Java

RFR = Refresh DLT = Deleted

**IBM**  server. For the next generation of e-business.

# Notes 5: Migration to V5R1

Product	Option	Status	Release	Description
5722-JV1	3	RFR	V5R1M0	Java Developer Kit 1.2
5722-JV1	4	RFR	V5R1M0	Java Developer Kit 1.1.8
5722-JV1	5	New	V5R1M0	Java Developer Kit 1.3
5722-PT1	Base	RFR	V5R1M0	Performance Tools
5722-PT1	1	RFR	V5R1M0	Performance Tools - Manager Feature
5722-PT1	2	RFR	V5R1M0	Performance Tools - Agent Feature
5722-CE1	Base	RFR	V5R1M0	Client Encryption 128-bit
5722-QU1	Base	RFR	V5R1M0	Query
5722-SM1	Base	RFR	V5R1M0	System Manager for AS/400
5722-ST1	Base	RFR	V5R1M0	DB2 Query Mgr and SQL DevKit
5722-TC1	Base	RFR	V5R1M0	TCP/IP Connectivity Utilities
5722-WDS	Base	New	V5R1M0	WebSphere Development ToolSet
5722-WDS	21	New	V5R1M0	Tools - Application Development
5722-WDS	31	New	V5R1M0	Compiler - ILE RPG IV
5769-WDS	32	New	V5R1M0	Compiler - System/36 Compatible RPG II
5722-WDS	33	New	V5R1M0	Compiler - System/38 Compatible RPG III
5722-WDS	34	New	V5R1M0	Compiler - RPG/400
5722-WDS	35	New	V5R1M0	Compiler - ILE RPG IV *PRV
5722-WDS	41	New	V5R1M0	Compiler - ILE COBOL

RFR = Refresh DLT = Deleted

**IBM**  server. For the next generation of e-business.

# Notes 6: Migration to V5R1

Product	Option	Status	Release	Description
5722-WDS	42	New	V5R1M0	Compiler - System/36 Compatible COBOL
5722-WDS	43	New	V5R1M0	Compiler - System/38 Compatible COBOL
5722-WDS	44	New	V5R1M0	Compiler - OPM COBOL
5722-WDS	45	New	V5R1M0	Compiler - ILE COBOL *PRV
5722-WDS	51	New	V5R1M0	Compiler - ILE C
5722-WDS	52	New	V5R1M0	Compiler - ILE C++
5722-WDS	53	New	V5R1M0	Compiler - ILE C *PRV
5722-WDS	54	New	V5R1M0	Compiler - ILE C++ *PRV
5722-WDS	55	New	V5R1M0	IBM Open Class - source and samples
5722-WDS	60	New	V5R1M0	Workstation Tools - Base
5722-WDS	61	New	V5R1M0	Workstation Tools - WebFacing, CODE
5722-WDS	62	New	V5R1M0	Workstation Tools - VisualAge RPG
5722-WDS	63	New	V5R1M0	Workstation Tools - WebSphere Studio
5722-WDS	64	New	V5R1M0	Workstation Tools - VisualAge for Java
5769-WSV	Base	RFR	V5R1M0	Integration for Windows Server
5722-WSV	1	RFR	V5R1M0	Integration for Windows NT 4.0
5722-WSV	2	RFR	V5R1M0	Integration for Windows 2000
5722-XE1	Base	RFR	V5R1M0	Client Access/400 Express for Windows
5722-XW1	Base	RFR	V5R1M0	Client Access Family for Windows
5722-XW1	1	New	V5R1M0	Client Access Enablement Support
1TMELCF	Base	Skip	V5R1M0	Tivoli Management Agent

RFR = Refresh DLT = Deleted

**IBM**  server. For the next generation of e-business.

## Specify Install Options

Type options, press Enter.

Restore option . . . . .	1	1=Restore programs and language objects from current media set 2=Do not restore programs or language objects 3=Restore only language objects from current media set 4=Restore only language objects from a different media set
Job and output queues option . . . . .	2	1=Clear, 2=Keep
Distribute OS/400 on available disk units . .	2	1=Yes, 2=No



The *Distribute OS/400 on available disk units* option specifies whether to use available disk units when restoring OS/400 objects.

If you specify **Yes**, the install function will distribute OS/400 objects on all available disk units in the system auxiliary storage pool (ASP) during the installation process. This option will do the following:

- Create new OS/400 objects from the install media
- Copy attributes from the existing object to the new object
- Delete the existing object from the system

This option will create new objects on the system during the install and thus will "spread" the operating system on all disk units configured in the system ASP. Security information, etc. will be copied from the existing system object to the new object and the system object will then be deleted. This option would typically be used in recovery situations where damaged objects exist on the system and the install will not complete successfully. This in essence will "scratch" OS/400 objects on the system without losing user data, etc.

This option will take more time and should only be used in specific recovery situations as directed by your service representative.

If you specify **No**, the install process will restore the objects from media over the existing objects on the system.

The default for this option will be **2=No** except for certain system determined situations. If an install is being done on a system on which newly configured DASD exists in the system ASP, this value will be set to **1=Yes** by default. This will cause the OS/400 objects to be "spread" on all units including the newly configured ones.

# Trademarks & Disclaimers

© Copyright International Business Machines Corporation 2001

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both

AIX	Application Development	AS/400
AS/400e	DB2	Domino
IBM	OfficeVision	OS/400
Integrated Language Environment	Net.Commerce	Net.Data
PowerPC	PowerPC AS	SanFrancisco
Host on Demand	Screen Publisher	Host Publisher
PCOM	WebSphere Commerce Suite	Payment Manager
WebSphere	WebSphere Standard Edition	WebSphere Advanced Edition
MQSeries	MQSeries Integrator	Host Integration Series
WebSphere Development Tools for AS/400	VisualAge for Java	VisualAge for RPG
CODE/400	DB2 UDB for AS/400	HTTP Server for AS/400
iSeries		

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET 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.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.


Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

**IBM  server. For the next generation of e-business.**