Session 11572

Using the z/OS SMB Server to Access z/OS Data from Windows Hands-On Lab

Lab prepared by Jim Showalter and Karl Lavo

© 2012 IBM Corporation

Trademark Information

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- z/OS®

The following are trademarks or registered trademarks of other companies.

- Linux is a registered trademark of Linus Torvalis in the United States, other countries or both
- Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation
- Red Hat is a registered trademark of Red Hat, Inc.
- UNIX is a registered trademark of The Open Group in the United States and other countries
- SUSE is a registered trademark of Novell, Inc. in the United States and other countries

Objective of this Lab:
At the end of this lab, you will be able to …

- Connect to the z/OS SMB server from Windows
- Access z/OS UNIX data from Windows
- Understand what is necessary to configure the z/OS SMB server

Files and directories in the lab

/sharelab – parent directory of home directories
/sharelab/sharaxx – your z/OS UNIX home directory
/sharelab/sharaxx/smblab – your z/OS UNIX working directory

Overview of z/OS DFS SMB (Distributed File Service Server Message Block)

- This support provides a server that makes z/OS UNIX files and traditional z/OS data sets available to SMB clients.
  - Hierarchical File Systems which include HFS,ZFS,TFS. It also can access file system data that is handled through AUTOMOUNT.
  - Includes sequential data sets, partitioned data sets (PDS), partitioned data sets extended (PDSE), and Virtual Storage Access Method (VSAM) data sets. The data set support is typically referred to as Record File System (RFS) support.
- The data set support is typically referred to as Record File System (RFS) support. To access shared directory paths and shared printers, the SMB protocol is supported through the use of TCP/IP on z/OS.
- Personal Computer (PC) clients on the network use the file and print sharing functions that are included in their operating systems.*
- Supported SMB clients include many Windows levels, as well as several Linux levels.
- Using SMB support, PC users can access files that are on a z/OS system. That is, PC users can access files that are not located on their PC computer.
  - Remote files are visible to the PC user on one or more separate drive letters.
  - PC users can "connect" an unused drive letter to a "shared resource" on a remote computer.
- In addition, Windows PC users are able to use remote printers that are attached to a z/OS system.
  - Remote printers are visible as additional printers available to the PC user.
  - Remote printers are installed on PCs using existing commands or install utilities.

* z/OS SMB server supports basic file and print serving. It does not necessarily support all functions that a Windows file server supports. For example, the z/OS SMB server does not support Kerberos authentication or DFS. There might be other functions that are not supported.

© 2011 IBM Corporation

IBM System z Technical University – Vienna , Austria – May 2-6

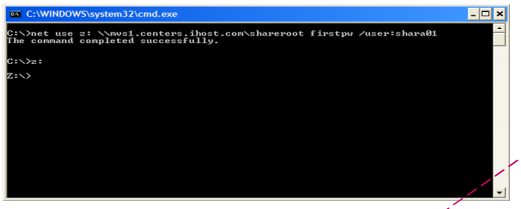
Very Simplified Overview of z/OS DFS SMB Configuration Steps

1. Define SMB administrators
2. Modify /opt/dfslocal/home/dfskern/envar to activate SMB file server (_IOE_PROTOCOL_SMB=ON)
3. Ensure that the DLLs used by the SMB server are APF-authorized.
4. For Windows, ensure that the computer name is the same as your TCP/IP host name in envar (_IOE_SMB_COMPUTER_NAME=SMBSHARE)
5. Check that there are no reserves for ports 137, 138, 139, and 445.
6. Define SMB users by using the **smbidmap** file pointed to in envar (_IOE_SMB_IDMAP=/opt/dfslocal/home/dfskern/smbidmap)
7. Determine what authentication methods you'll use: passthrough authentication, password encryption, and whether to allow guest users.
8. Determine if you want to have dynamic export capability (to support automount).
9. Define shared directories by updating the **smbtab, dfstab,** and **devtab** files in /etc/dfs/var/dfs/.
10. Start the DFS server (**dfskern**). Sample is found in IOE.SIOESAMP(DFS).
11. Configure the PC client workstation to access the SMB server. (Necessary configuration is minimal.)

Complete steps can be found in *z/OS DFS SMB Administration*

© 2011 IBM Corporation

z/OS SMB Configuration for Lab

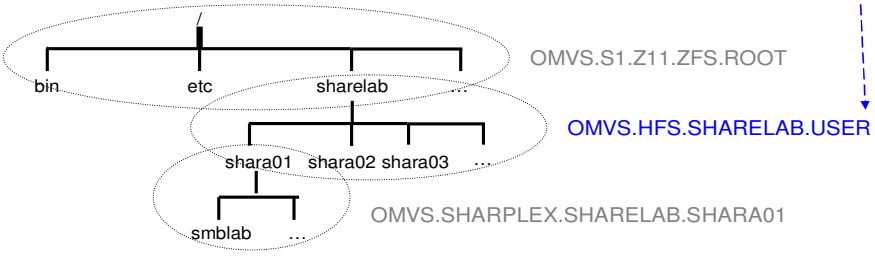


smbtab:
/dev/ufs2 shareroot ufs "Sharelab root" r/w 100 /

dfstab:
/dev/ufs2 hfs2 ufs 2 0,,2

devtab:
define_ufs 2
OMVS.HFS.SHARELAB.USER auto

C:\) net use z: \\mvs1.centers.ihost.com\shareroot firstpw /user:shara01



OMVS.S1.Z11.ZFS.ROOT

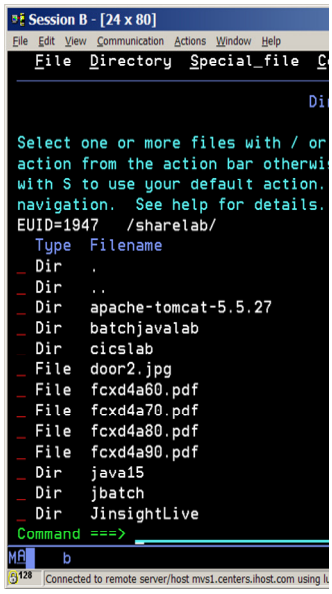
OMVS.HFS.SHARELAB.USER

OMVS.SHARPLEX.SHARELAB.SHARA01

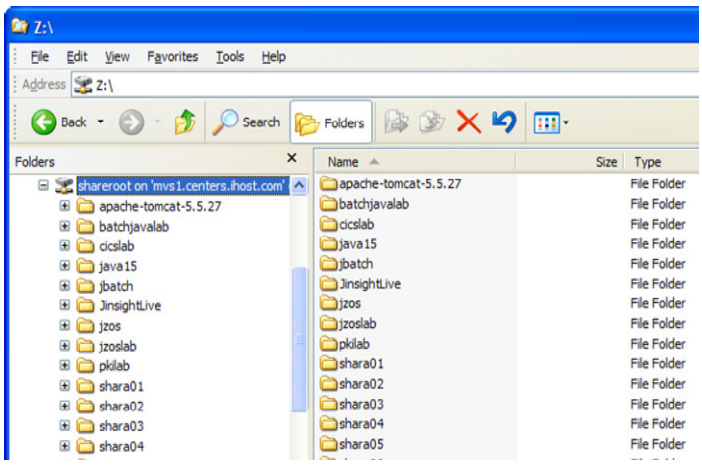
© 2011 IBM Corporation

Lab Example:

z/OS:



Your workstation:



© 2011 IBM Corporation

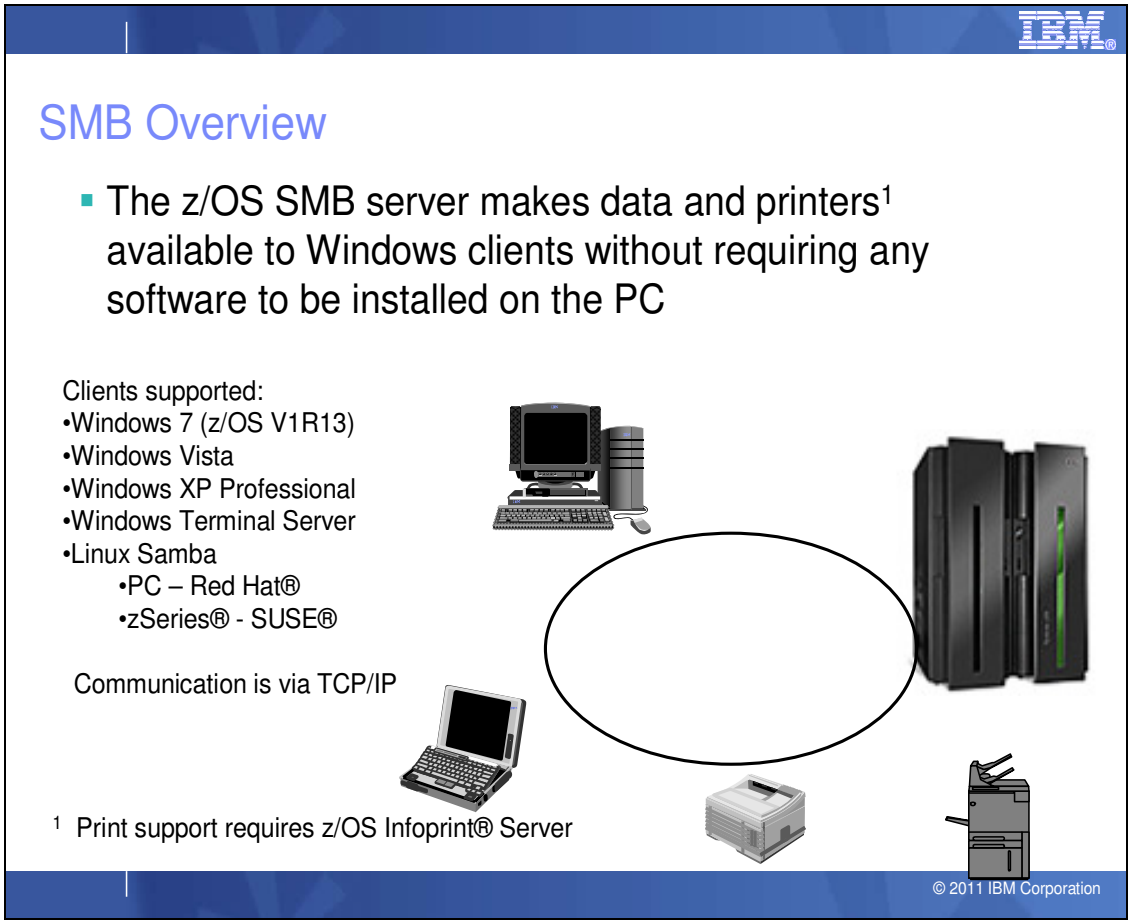
What are we going to do?

In this session, you are going to:

- Get a short, high level overview of what the z/OS SMB server is
- See how a PC user accesses z/OS UNIX data
- See the z/OS SMB server configuration files that control which data is available to PC users
- Learn a few important concepts
  - Dynamic export
  - Different authentication methods used by the z/OS SMB server
- Access z/OS UNIX data from your Windows machine (**beginning of the actual exercise**)
- View the actual z/OS SMB server configuration files
- There are backup slides and the end of this document that:
  - Tell you how to install the z/OS SMB server and specify/modify the z/OS SMB server configuration files
  - Explain the structure of the SMB server
  - Explain which server configuration files to modify
    - ioepdcf in /opt/dfslocal/etc
    - envar in /opt/dfslocal/home/dfskern and /opt/dfslocal/home/dfscntl
    - devtab, dfstab and smbtab in /opt/dfslocal/var/dfs
    - smbidmap in /opt/dfslocal/home/dfskern (Location actually controlled by _IOE_SMB_IDMAP specification in /opt/dfslocal/home/dfskern/envar)

z/OS SMB Server overview

The z/OS SMB server allows PC users (Windows or Linux) to access z/OS data directly. PC users connect to the z/OS SMB server in very much the same way that PC users would connect to a Windows File Server. Once connected to the z/OS SMB server, the PC user can use a drive letter to access the data.

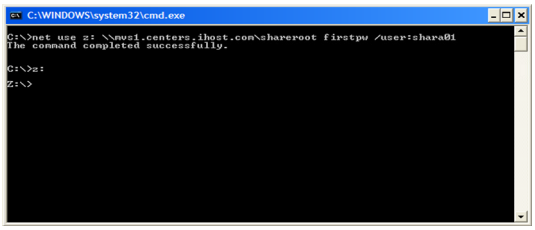


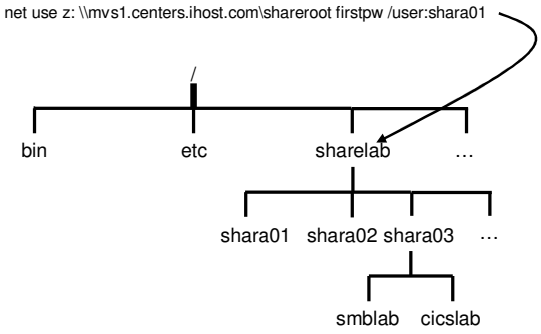
How a PC user accesses z/OS data

To access z/OS data from a PC, you must know the **hostname** of the z/OS server and the **sharename** of the data. In the example below, the hostname is mvs1.centers.ihost.com and the sharename is shareroot. In addition, you may need to specify a userid and password. In the example below, the userid is shara01 and the

password is firstpw. This assigns drive letter z: to the data represented by shareroot.

SMB Access to z/OS UNIX data

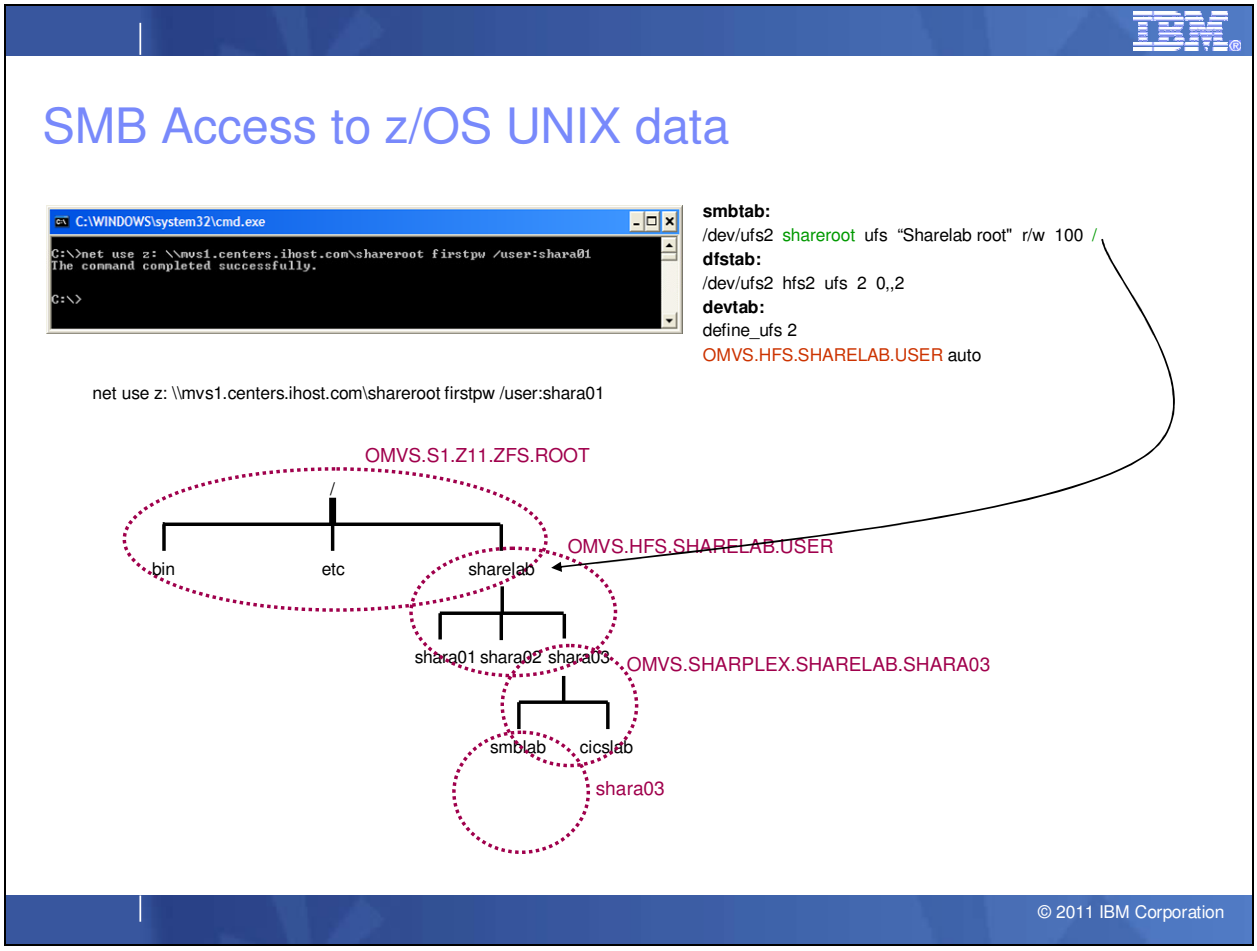




© 2011 IBM Corporation

z/OS SMB server configuration files

Here is a quick view of the z/OS SMB server configuration files that define the sharename and what data it refers to. The **smbtab** defines the sharename and the **devtab** contains the file system name.



Dynamic export

Each file system that is to be accessible to the PC users, must be exported by the z/OS SMB server. The first file system must be explicitly exported at SMB server initialization by having an entry in the **devtab** configuration file. As the PC user uses the **cd** command (or Windows Explorer) to go down the hierarchy (to subdirectories), she may **cd** to a mount point which effectively crosses into the root of the next lower file system. That file system must be exported too. It could be explicitly exported at initialization with another **devtab** entry or you can use the dynamic export capability of the z/OS SMB server. If you specify _IOE_DYNAMIC_EXPORT=ON in the /opt/dfslocal/home/dfskern/envar file, then the SMB server will automatically export any file system it encounters as PC users move around the file system hierarchy.

Static Export/Dynamic Export

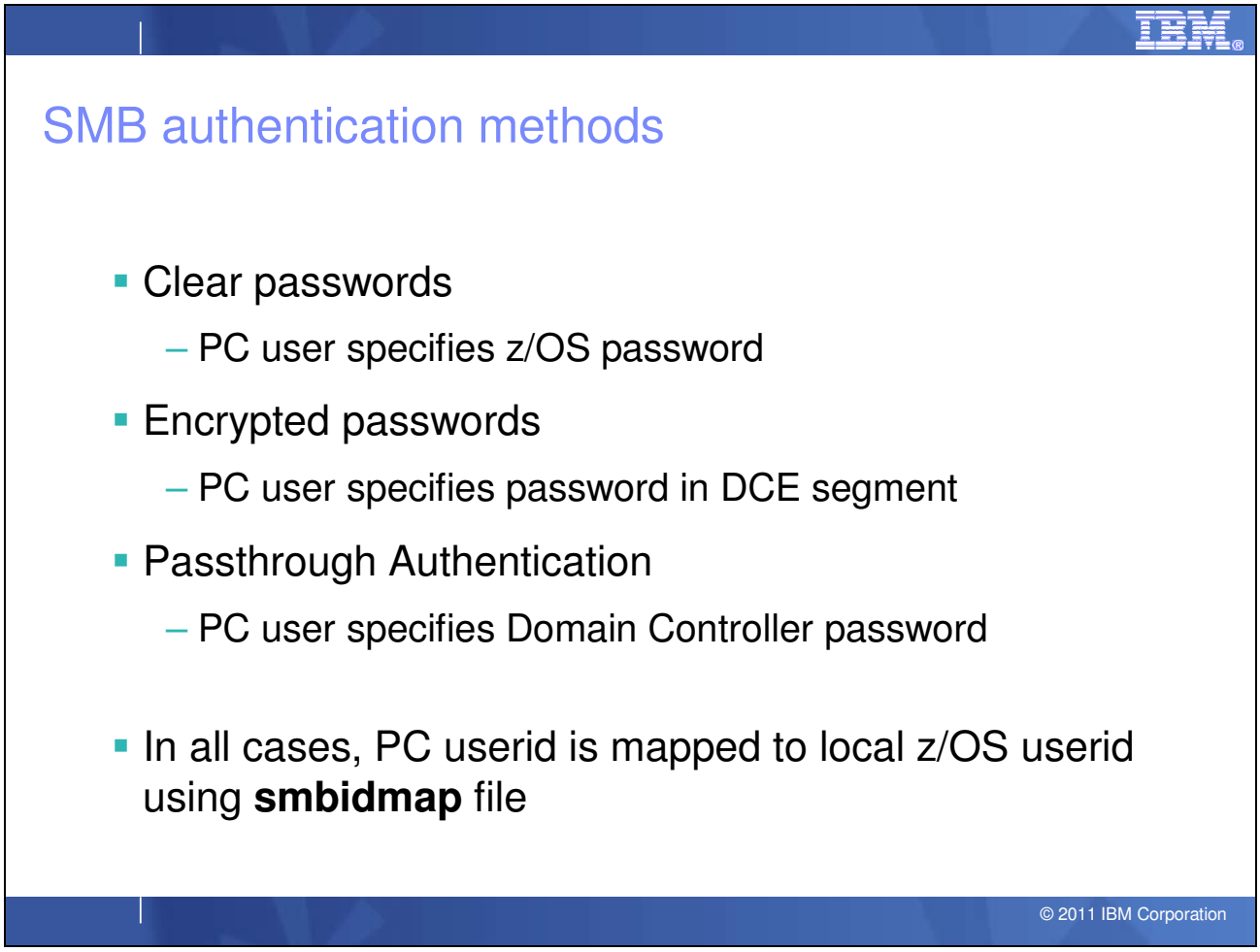
- Normally, each file system to be made available to PCs must be exported by specifying it in a separate **dfstab**/**devtab** entry (static export)
- You can, however, choose to let the SMB server "discover" mounted file systems and dynamically export them (RFS cannot be mounted)
- The SMB server can dynamically export HFS, zFS, TFS and AUTOMNT
- The file system containing the shared directory must still be statically exported

© 2011 IBM Corporation

SMB authentication methods

There are three methods that the z/OS SMB server can use to authenticate a PC user.

1. Use clear passwords. In this case the password you need to specify is your MVS (RACF) password.
2. Use encrypted passwords. In this case the password you need to specify is the password you have stored in your DCE segment using the smbpw command.
3. Use passthrough authentication. In this case the password you need to specify is your PC password as known to your Domain Controller.



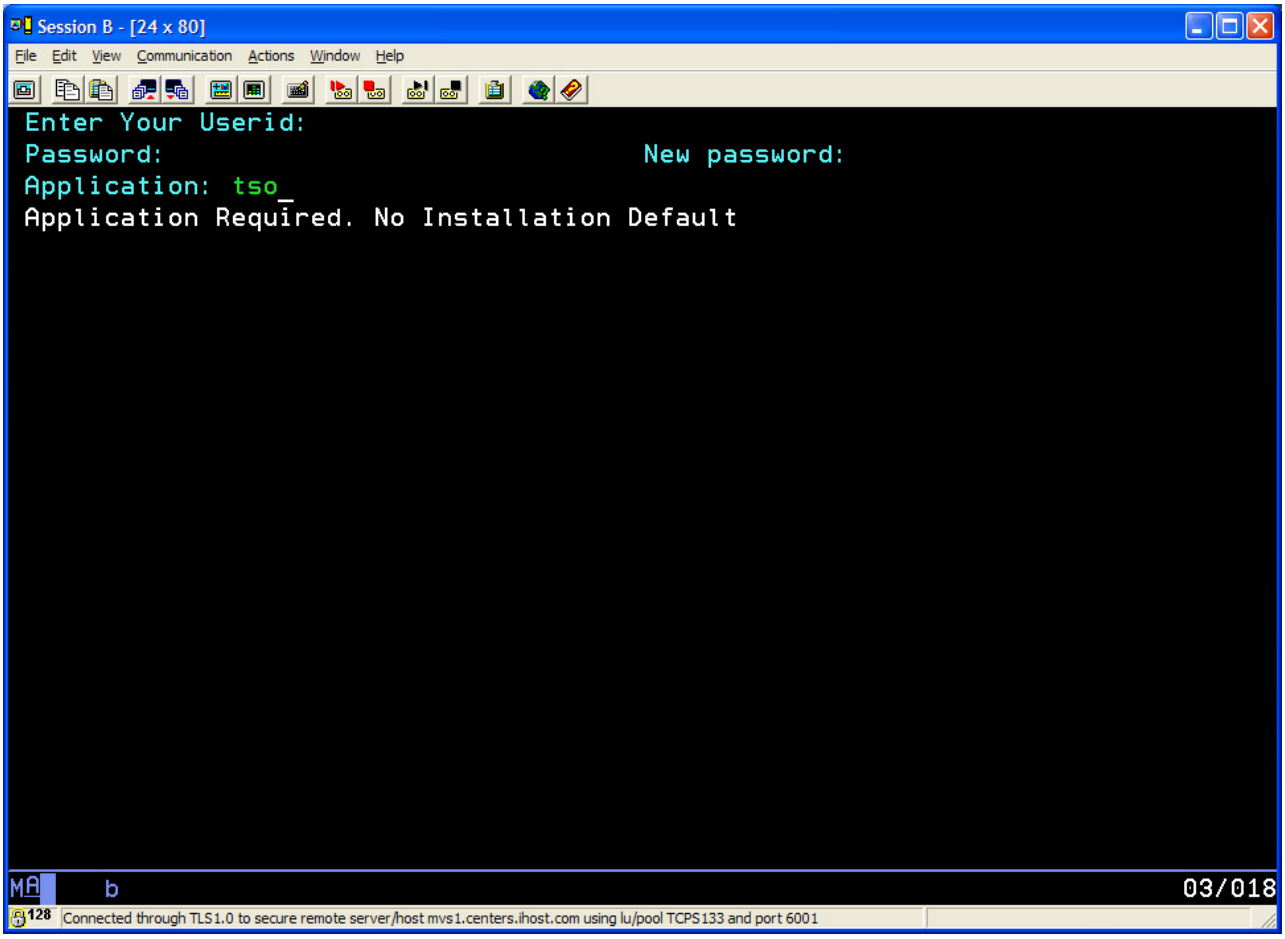
Now, let's start the lab exercises.

To start your TSO session:

Begin the lab exercise by starting your 3270 session to TSO. You will use an MS-DOS session later in the lab exercise.

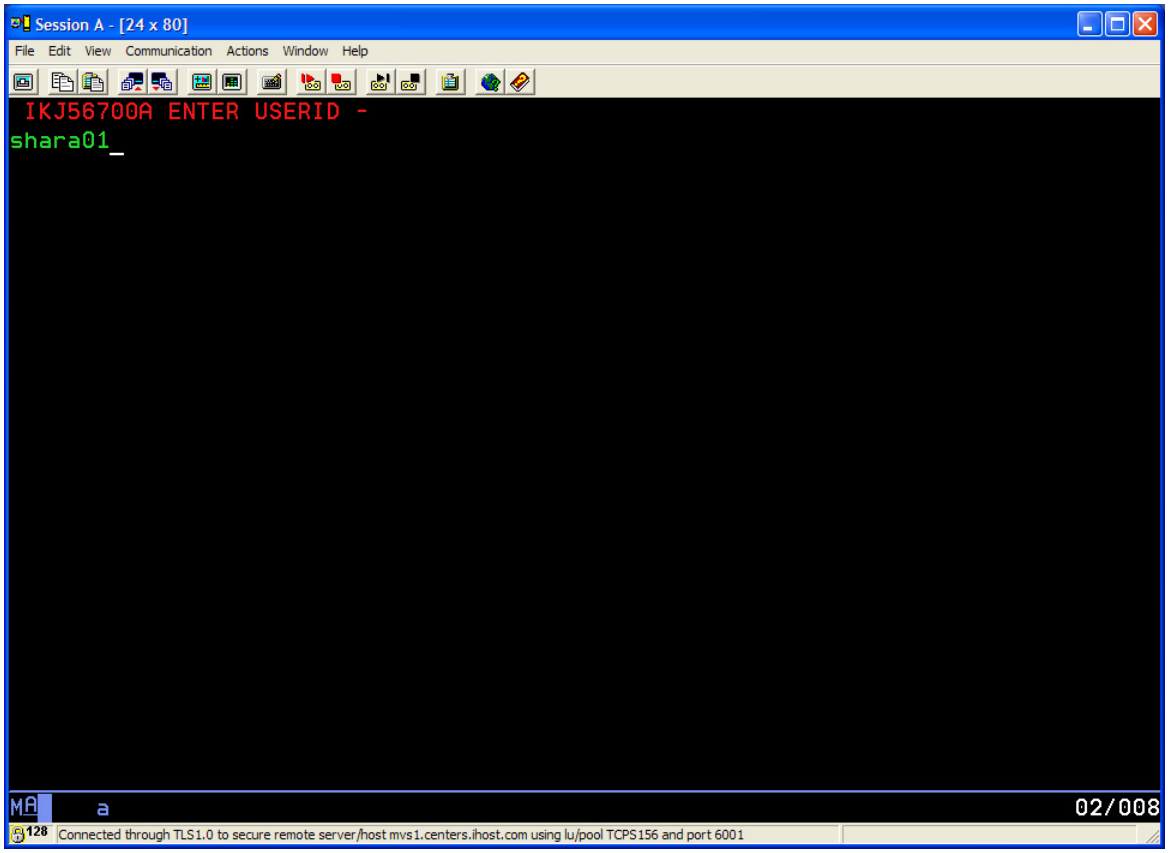
__ 1. Locate the MVS icon on your desktop.

__ 2. Double-click the MVS icon to open the PCOMM window. You should see the VTAM USS logon screen. Enter TSO in the *Application* field:



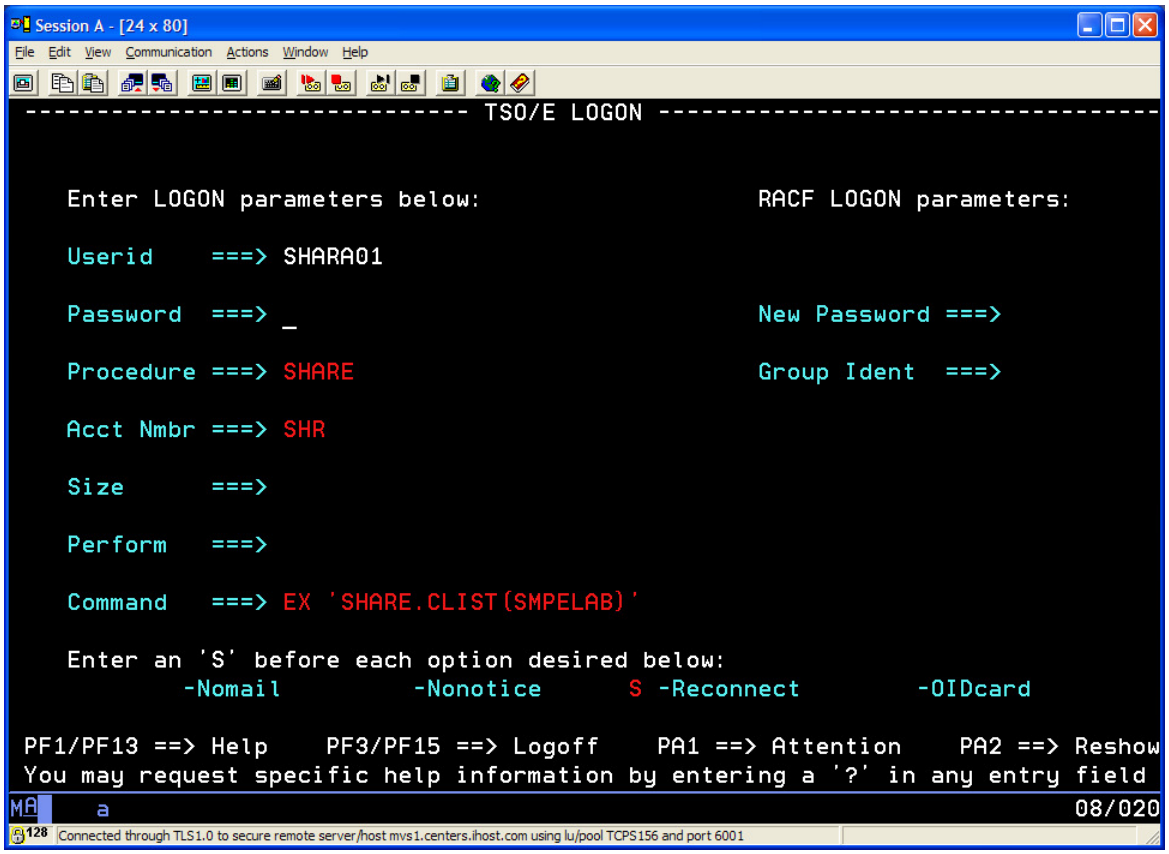
Leave the *Userid* and *Password* fields blank.

__ 3. You should see the TSO logon screen with message IKJ56700A:



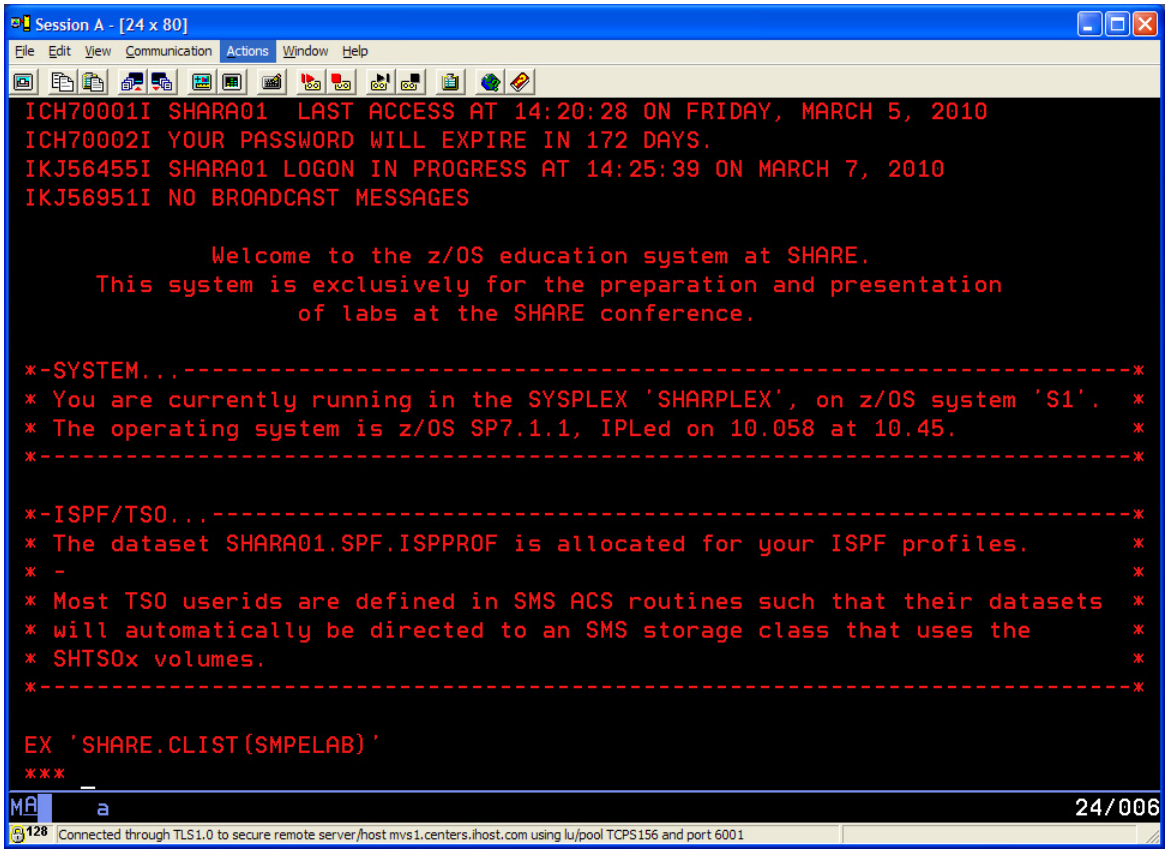
Enter your TSO user ID and password as provided by your instructor. For example, SHARAnn where nn is 01 through 30.

__ 4. You should see the TSO logon panel:

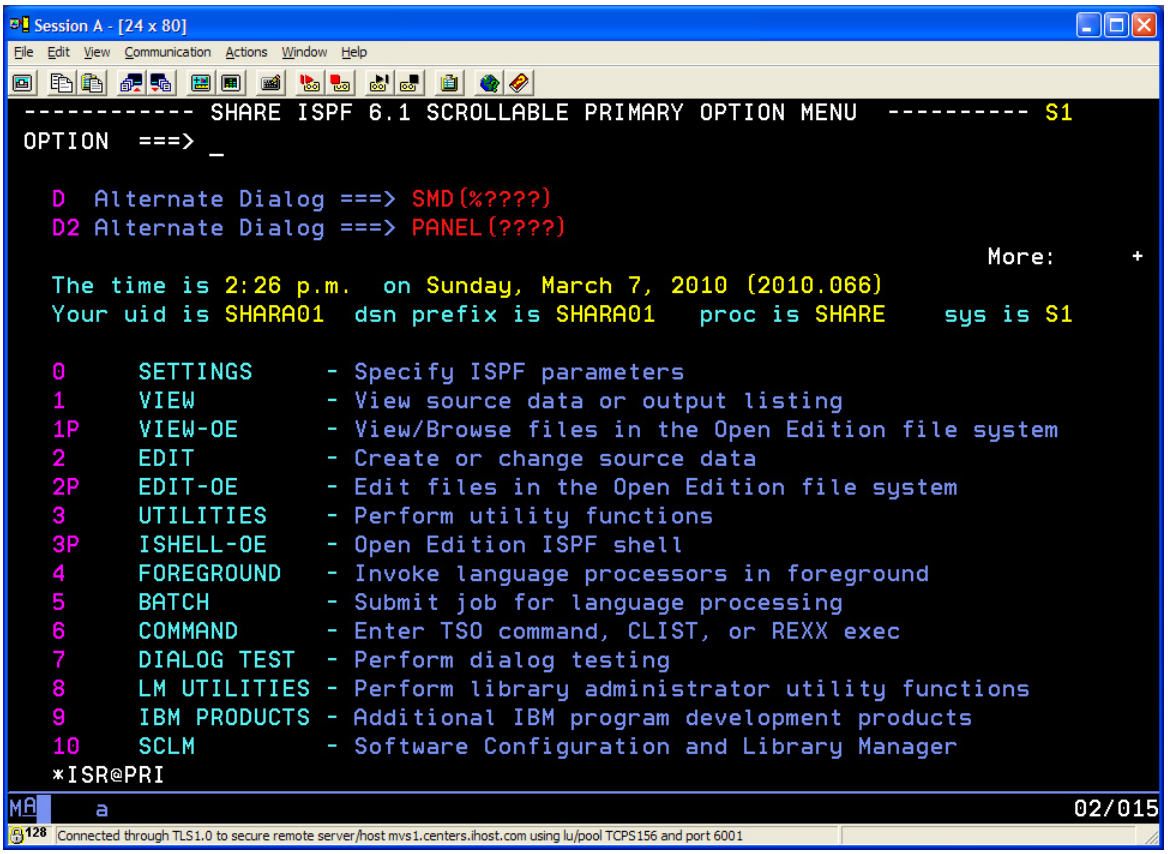


Enter the password provided by your instructor and press Enter

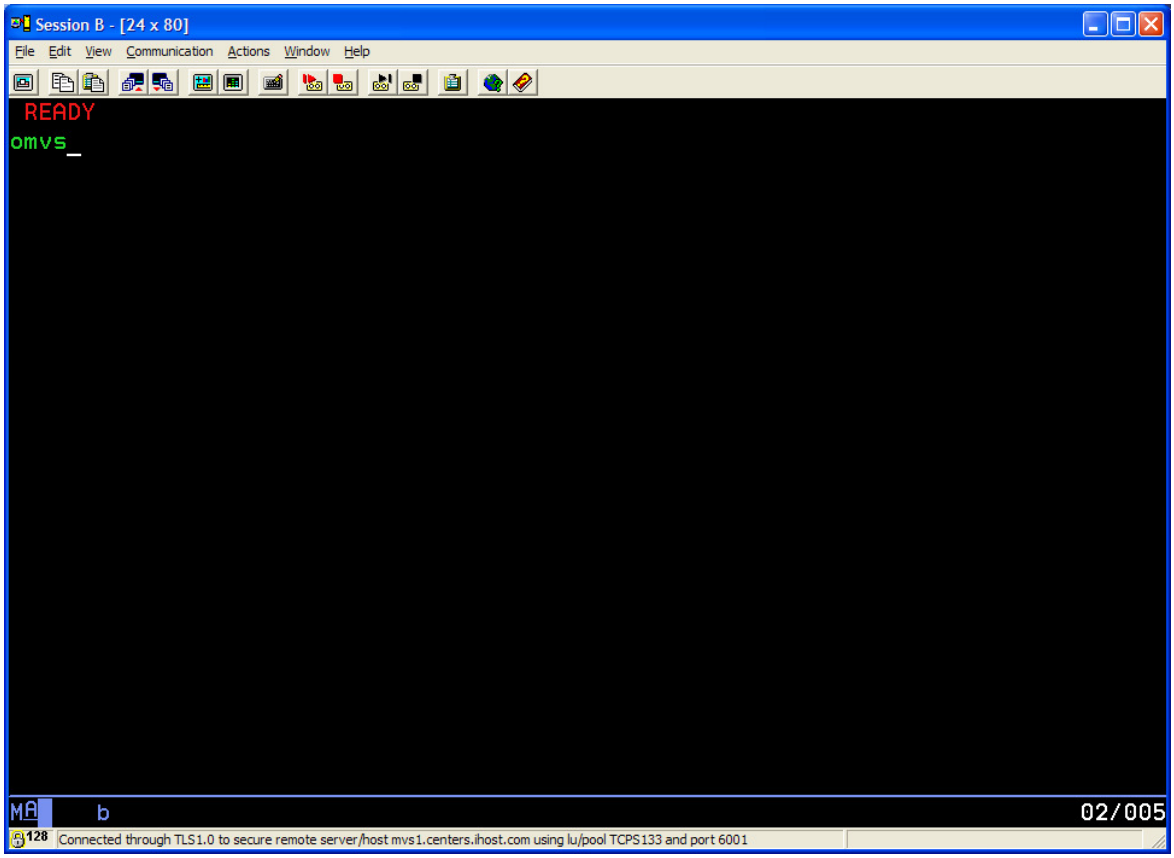
__ 5. You should see a panel similar to this one and be connected to TSO:



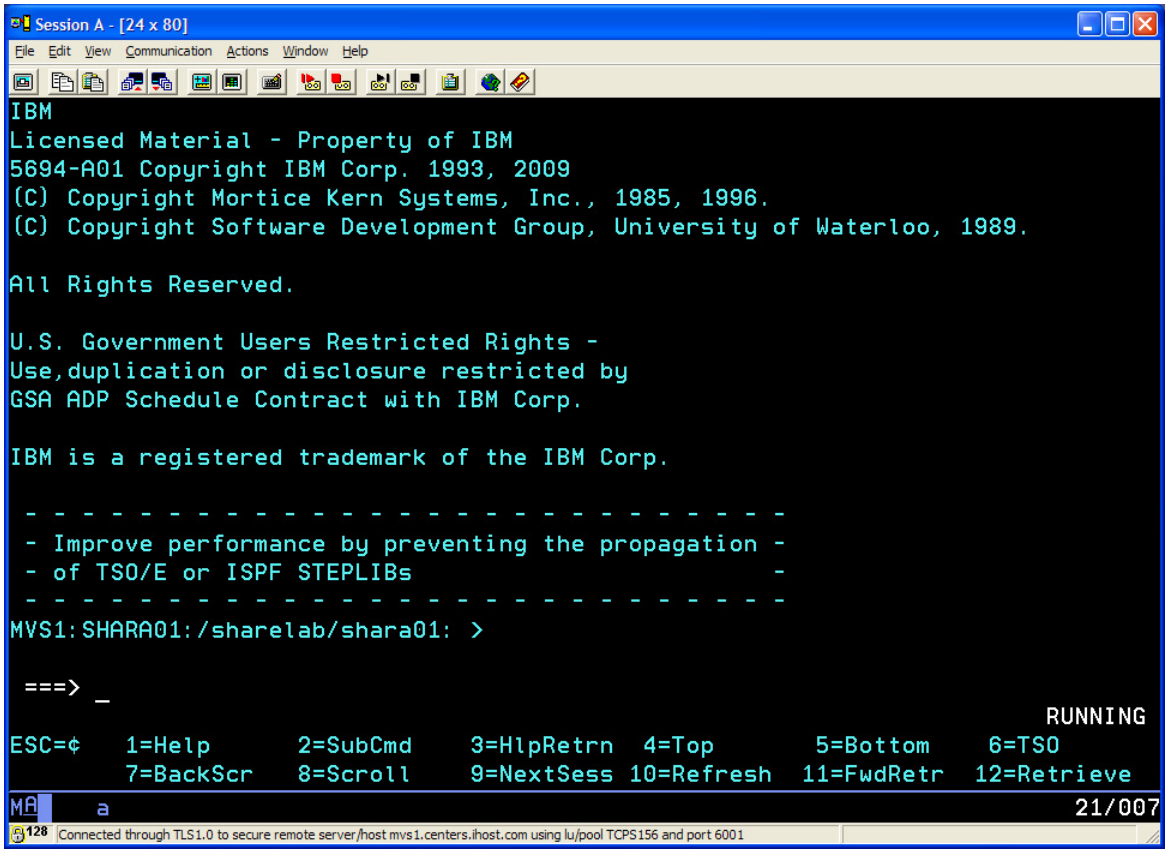
__ 6. Press Enter to access the ISPF Primary Menu:



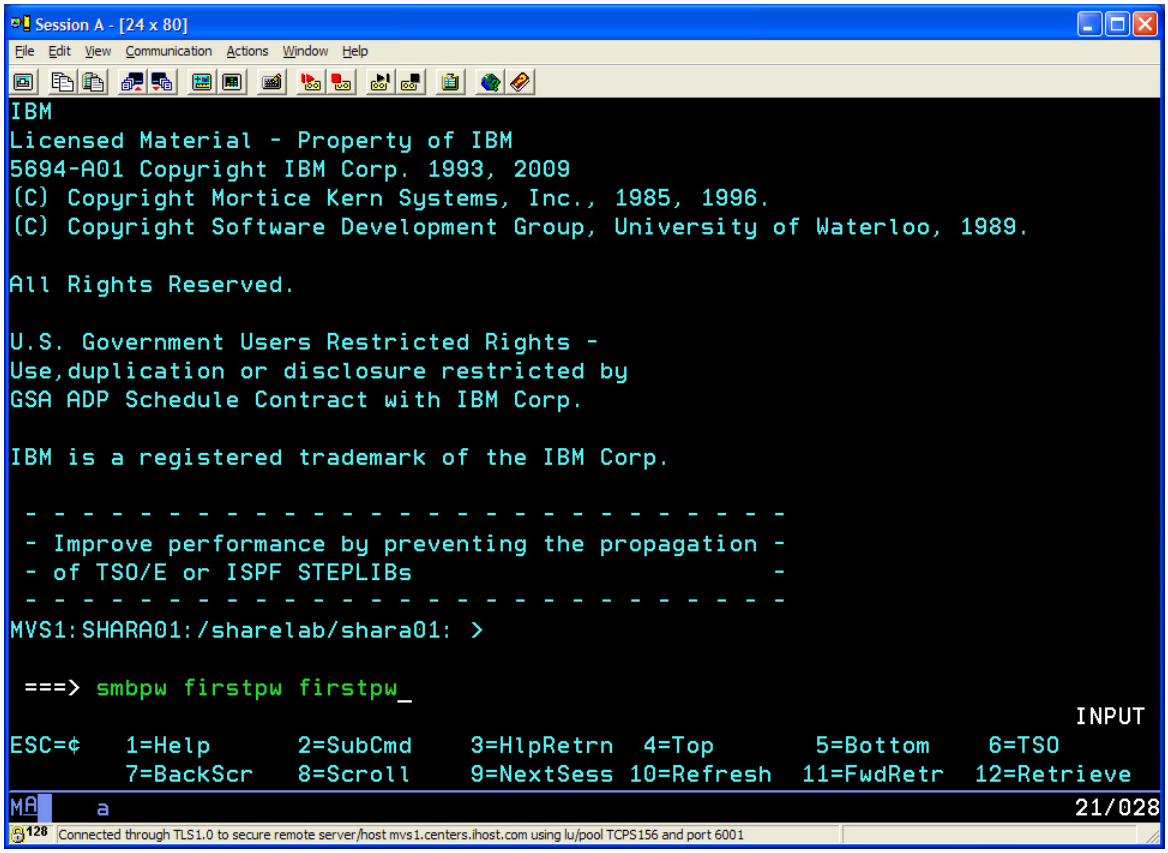
__ 7. Press F3 to exit ISPF (to get to the TSO screen) and enter OMVS and hit Enter:



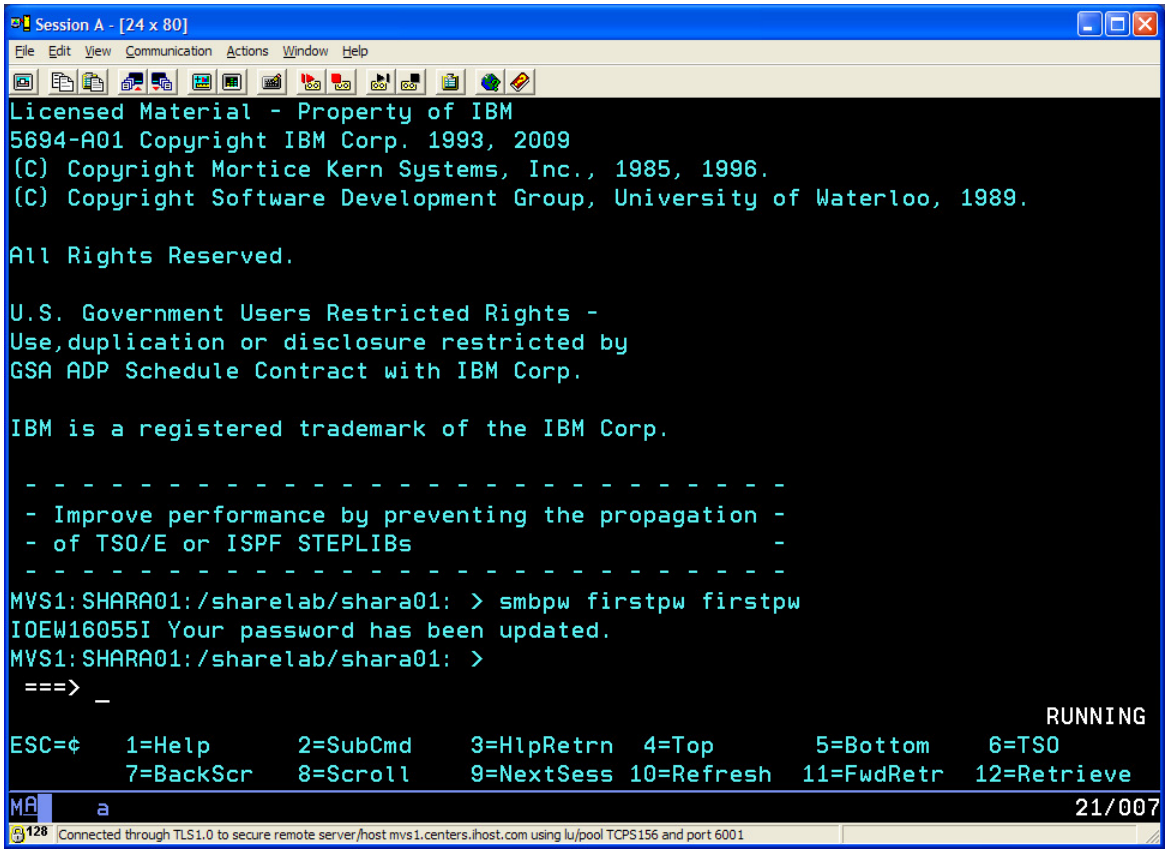
__ 8. You should see the OMVS screen (looks like below – except that the userid will be the one you used to logon to TSO):



__ 9. Next issue the **smbpw** command to tell the SMB server about the password you are going to use when you connect to z/OS from Windows:

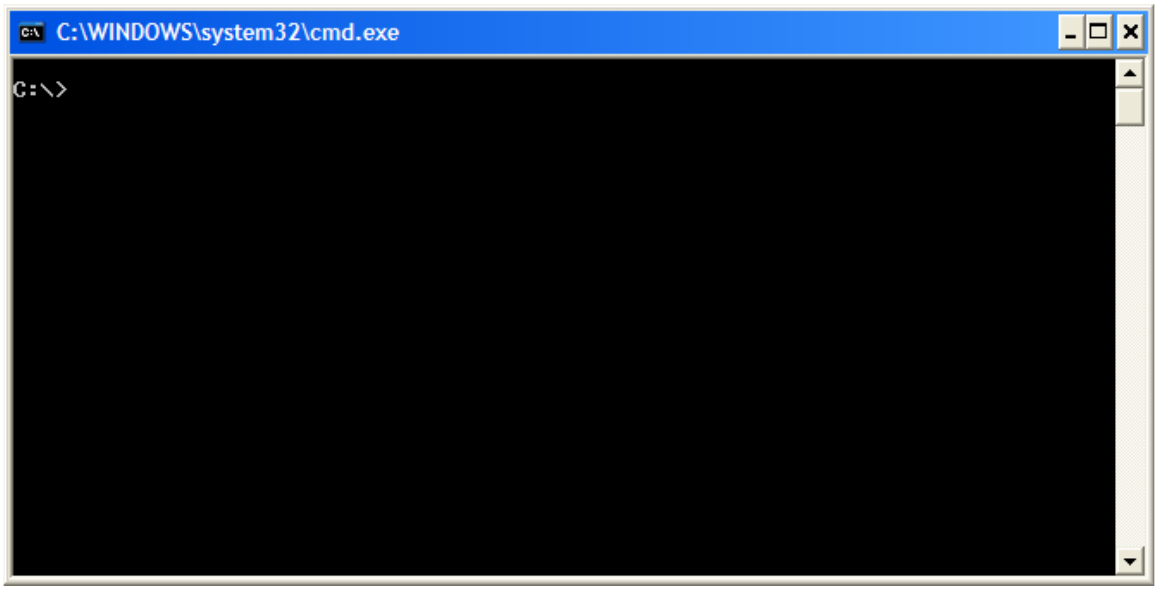


__ **10.** You should see that the password has been updated in z/OS as shown in the following screen:



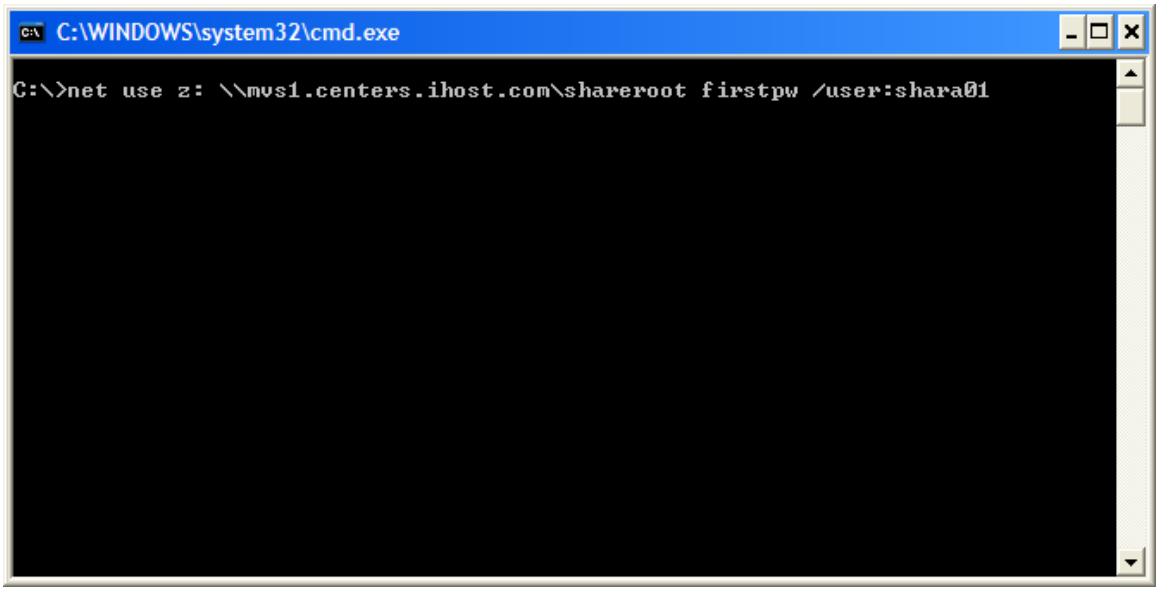
Now you are all set to connect from Windows to z/OS.

__ **11.** Next, find the MS-DOS icon on your desktop and double click it to open an MS-DOS window

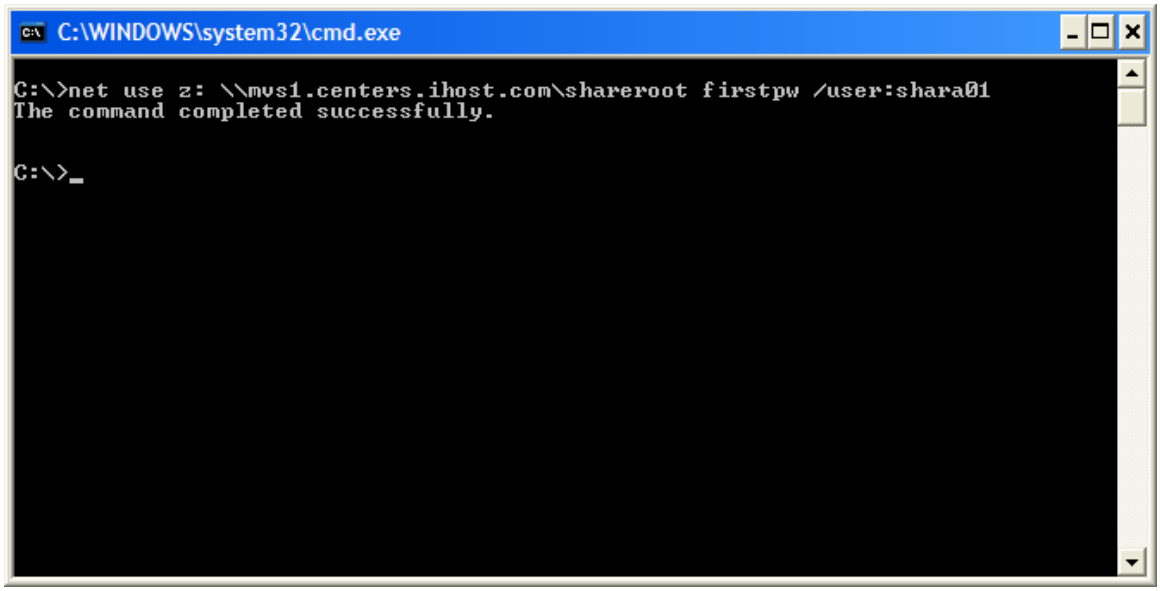


__ **12.** Type in the net use command (one one line and using the sharann userid given to you by the instructor):

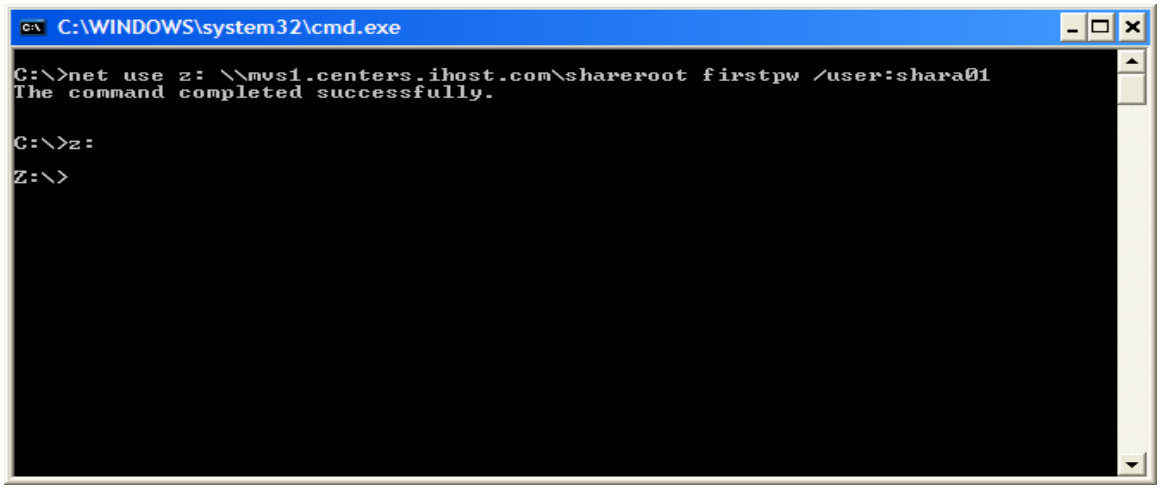
net use z: \\mvs1.centers.ihost.com\shareroot firstpw /user:sharann



__ **13.** You should see the net use command was successful (there may be a delay before it responds – be patient):

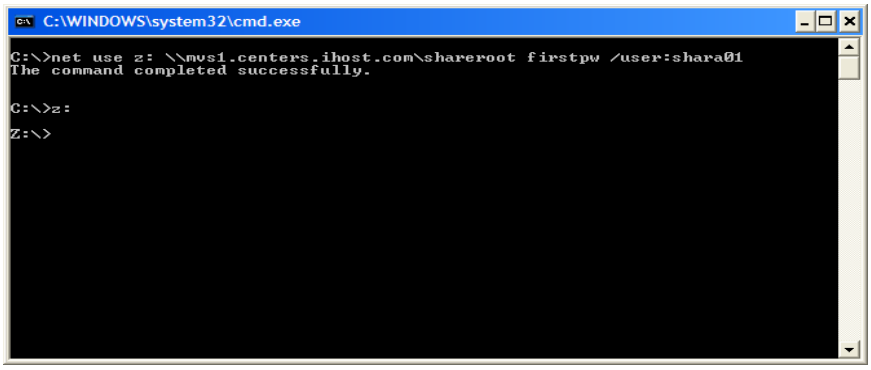


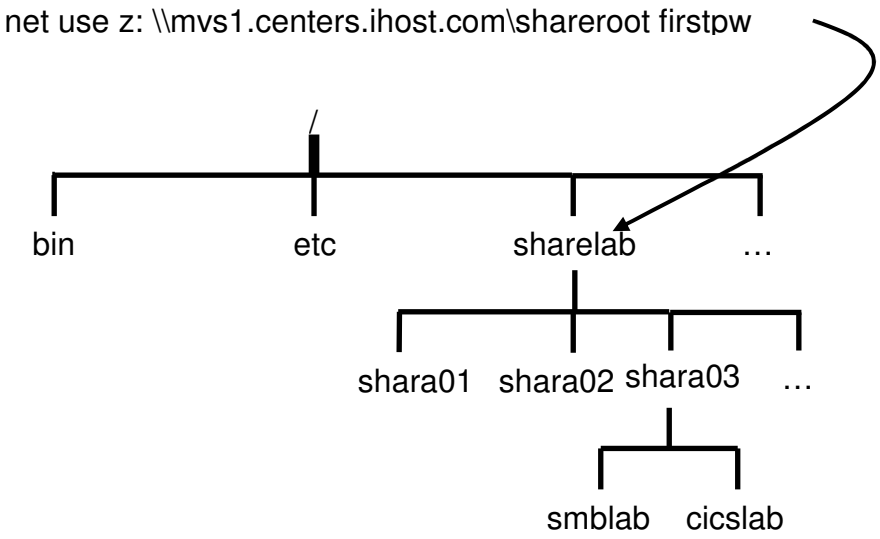
__ **14.** Next, you can change drive letters by specifying the z drive (type **z:** and hit enter)



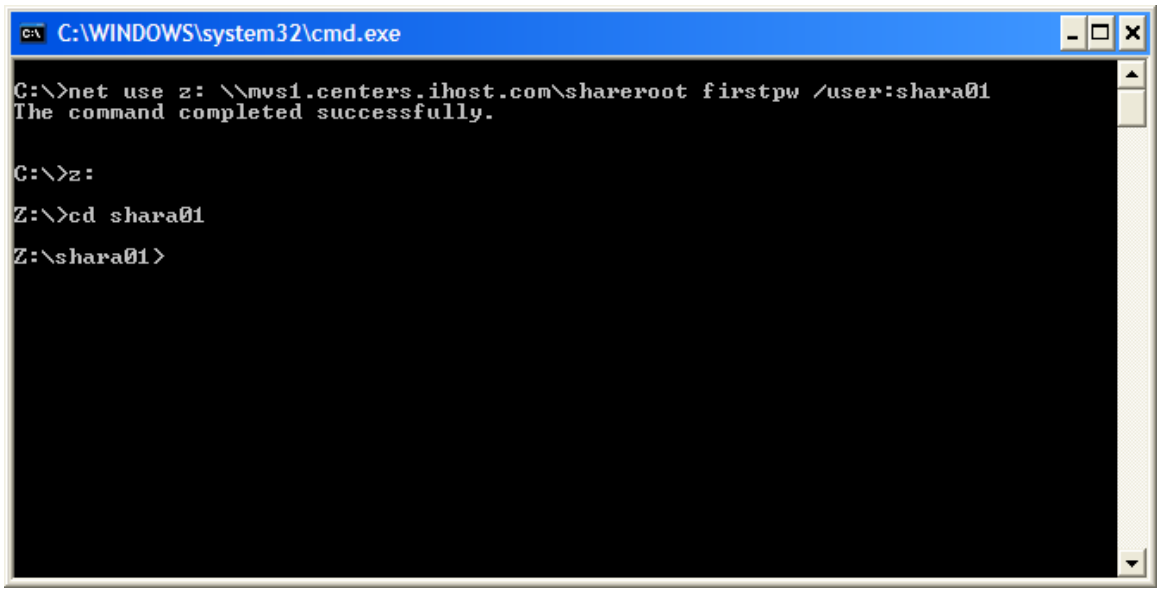
You are now positioned at the root of all the lab home directories.

This is what the z/OS UNIX hierarchy looks like and you are positioned at the /sharelab directory which is the parent of the sharann home directories.

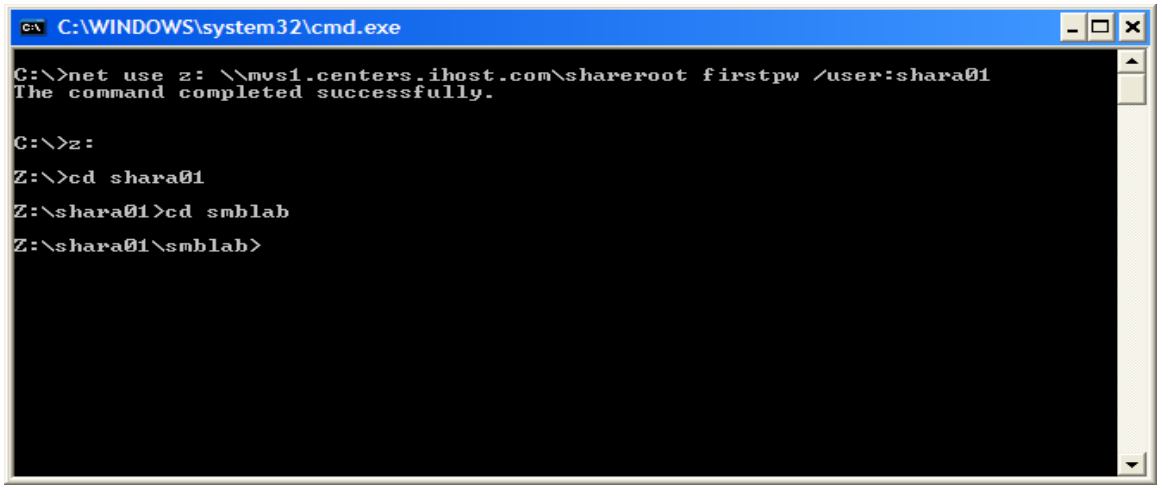




__ **15.** You can change directories (cd) to your home directory. Be sure to use the directory name that is the userid that was given to you by your instructor.

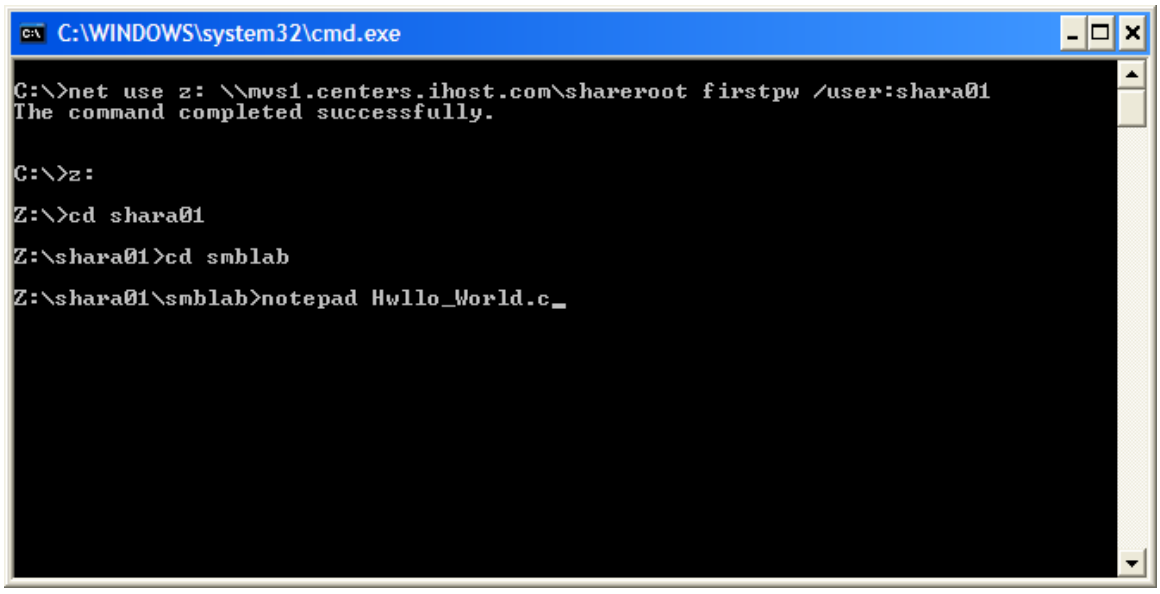


__ **16.** You should now change directories (cd) to the smblab directory. This is the place you will use to create a file.

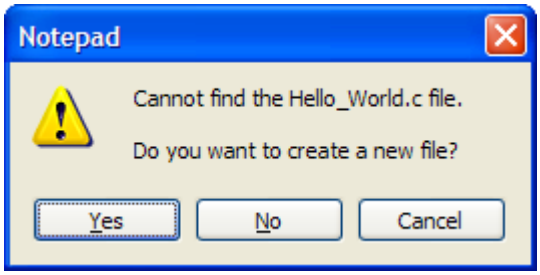


__ **17.** You are going to use Notepad to create a file directly onto z/OS. The file will be called Hello_World.c and it will contain a C program which can be compiled and run on z/OS. (It will never touch the local hard disk on the PC.)

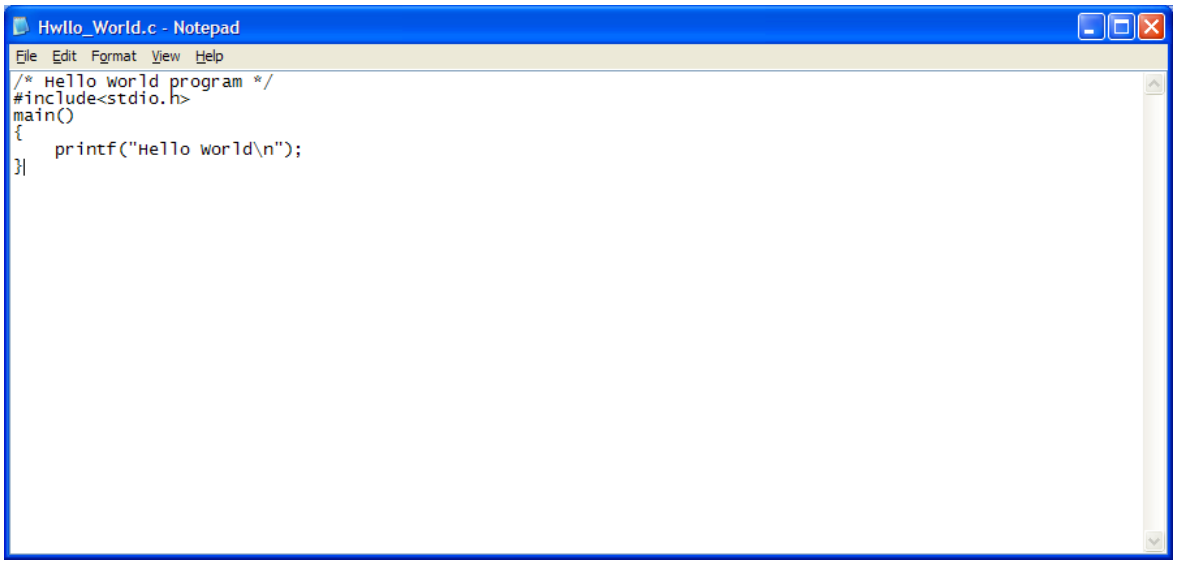
While you are positioned in the z:\shara01\smblab directory, type **notepad Hello_World.c** and hit Enter.



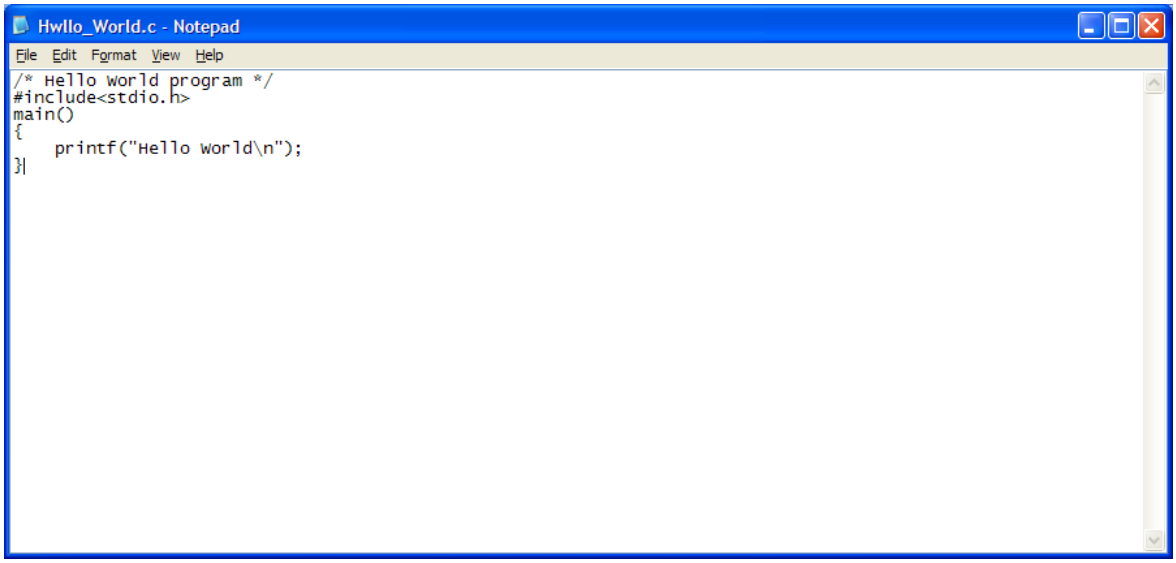
__ **18.** This will cause Windows to display the following window:



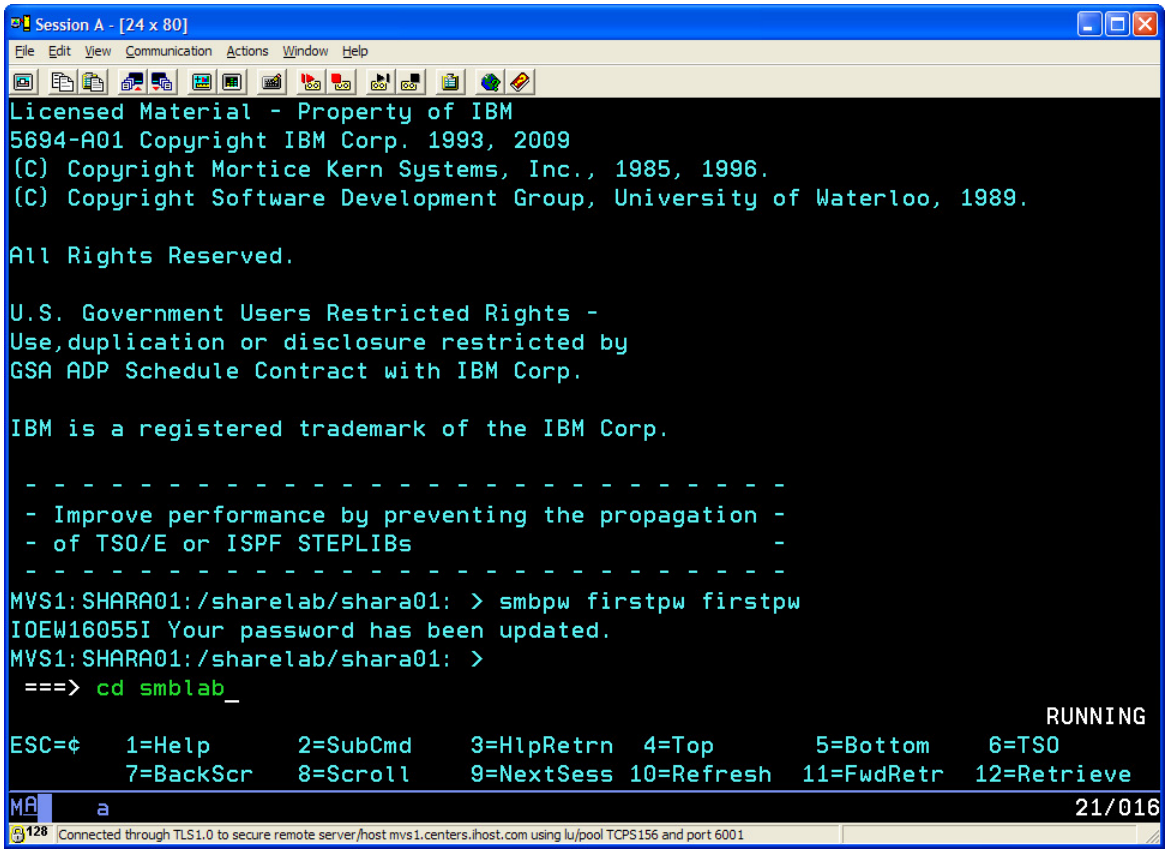
Click yes and you should see the following window (after the program has been typed in):



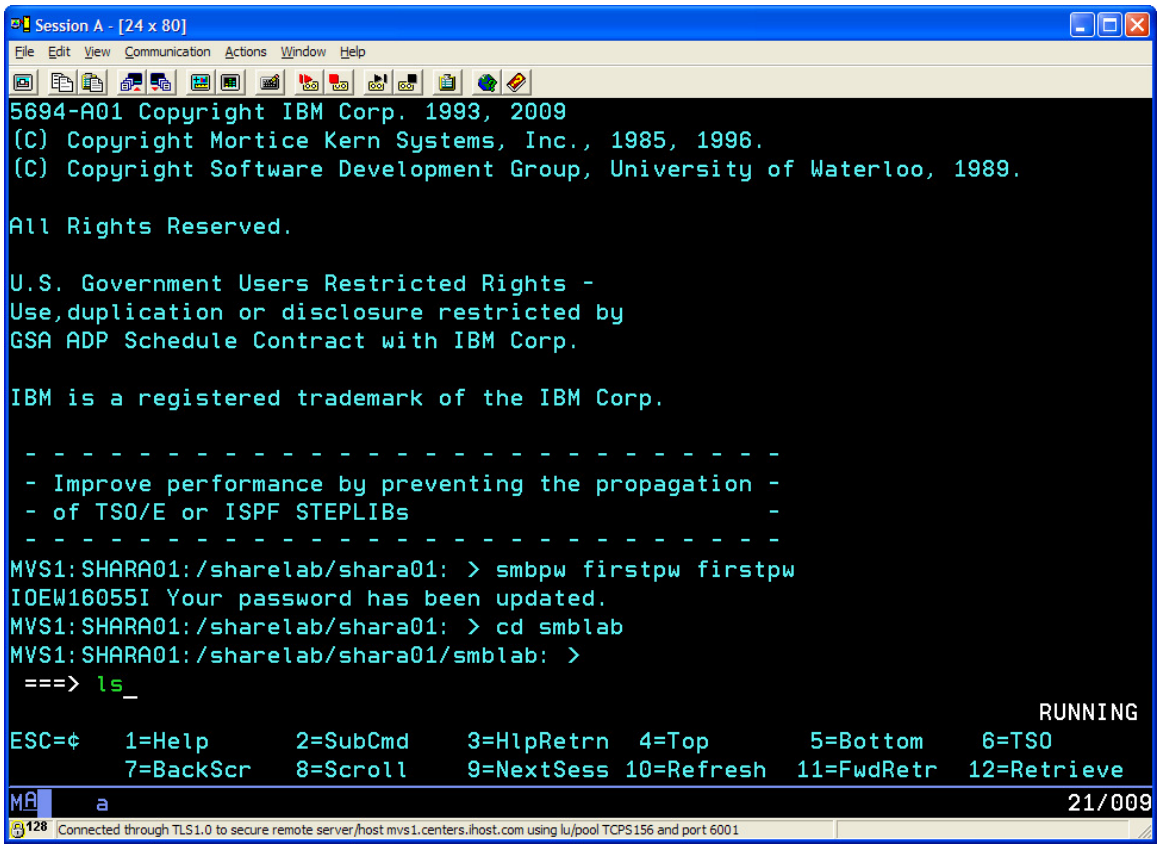
__ **19.** Then Click File, Save.



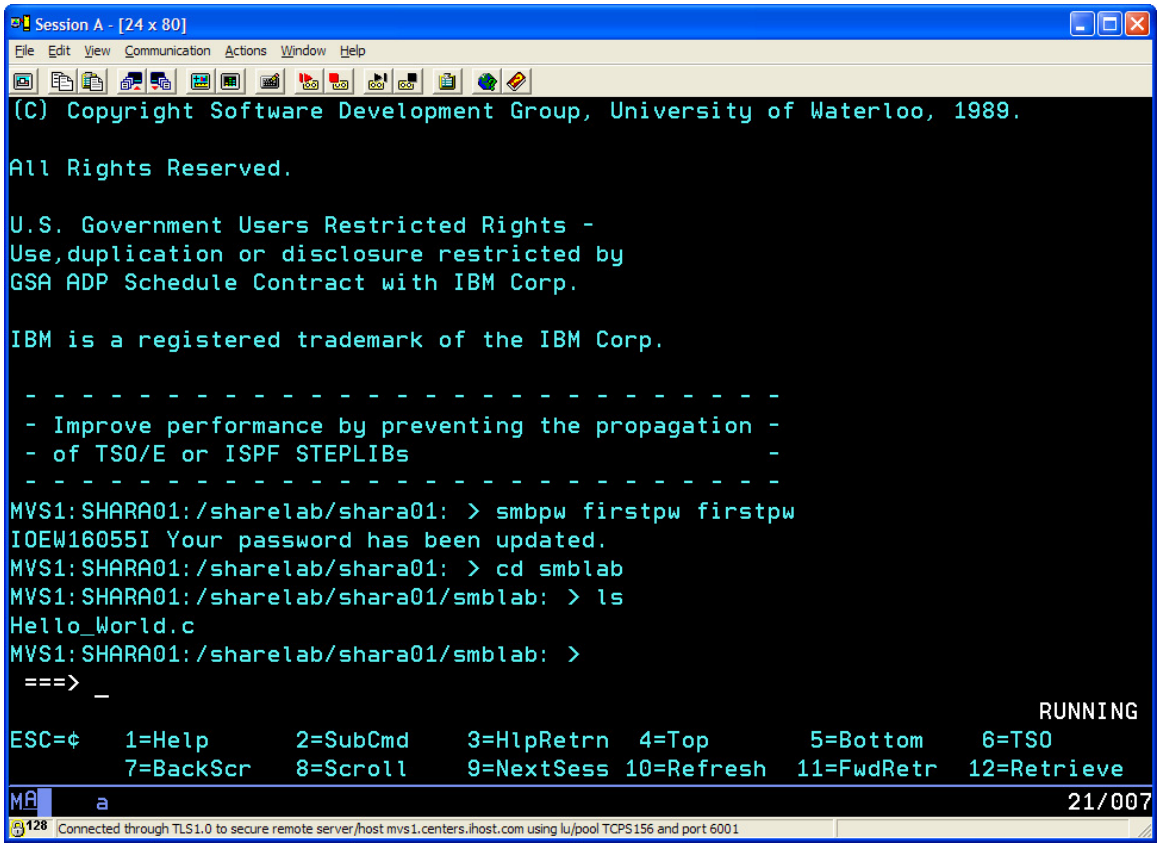
__ **20.** Close that window and go back to your OMVS window.
Change directory to smblab.



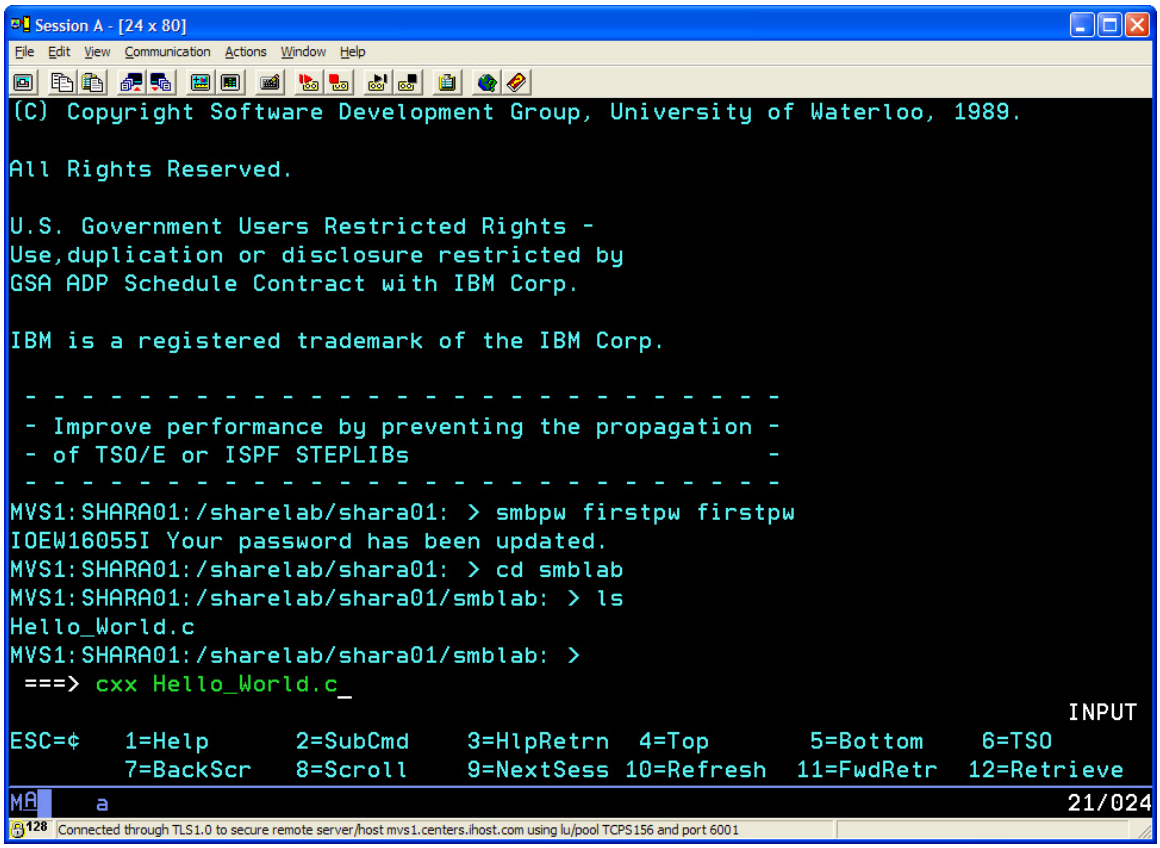
__ **21.** This is where the Hello_World.c source program is located that was created from the PC. Type **ls** to verify that Hello_World.c is in this directory.



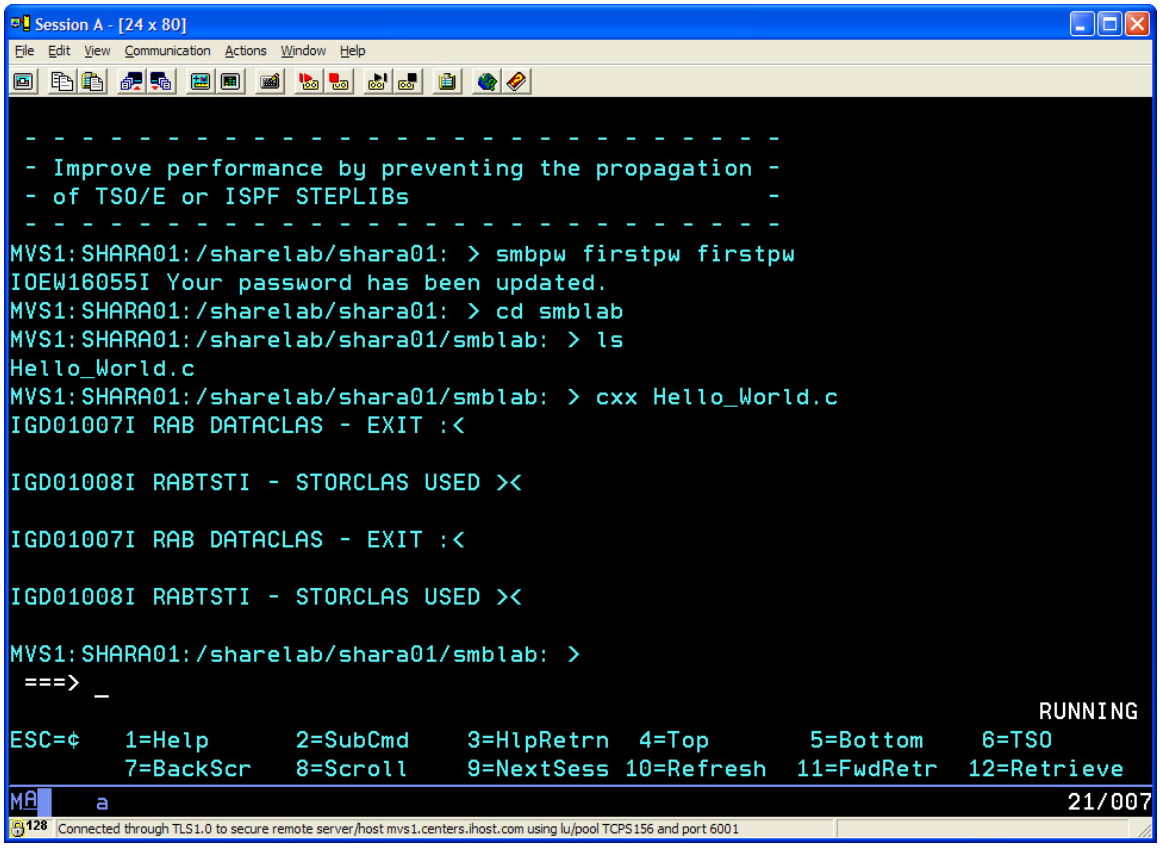
__ **22.** You should see that the name Hello_World.c is displayed.



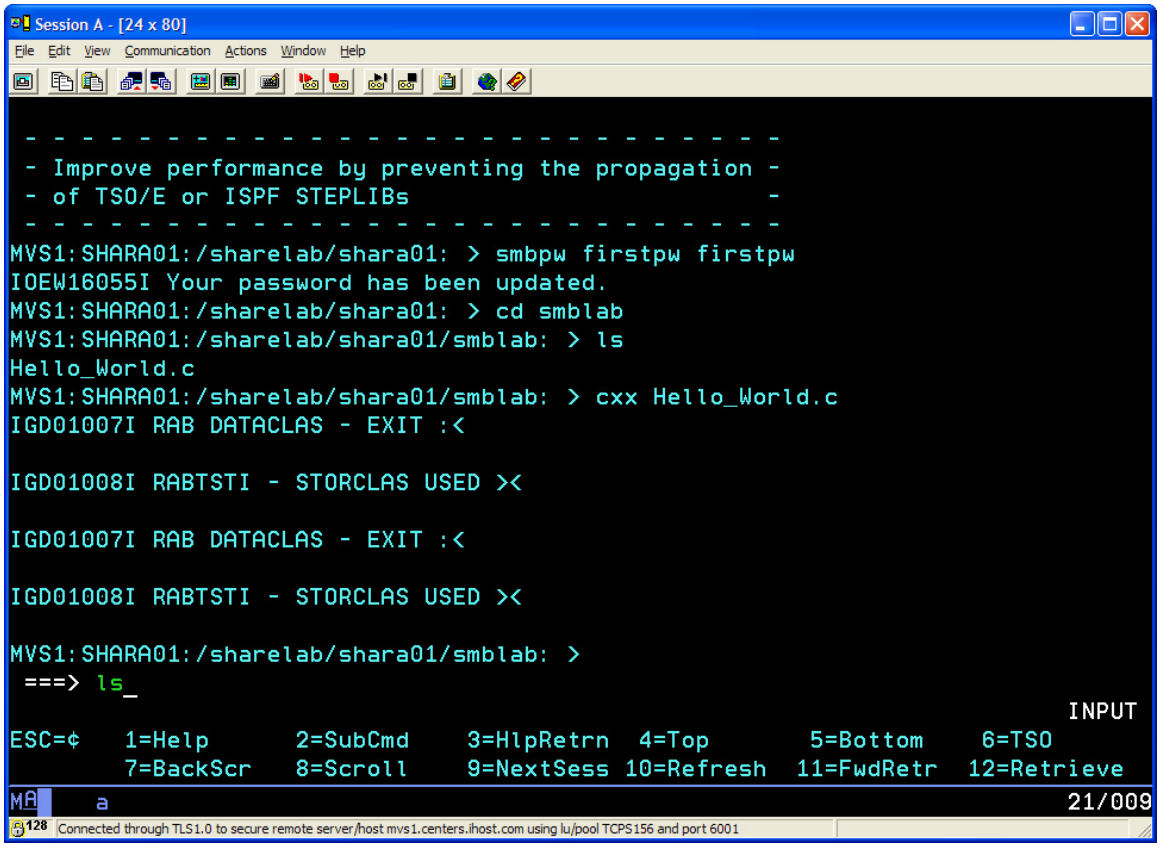
__ **23.** We are now going to use the z/OS C/C++ compiler to compile and link the Hello_World.c program that we entered from the PC. Type **cxx Hello_World.c** to compile and link the program (be careful of upper and lower case letters – z/OS UNIX is case sensitive).



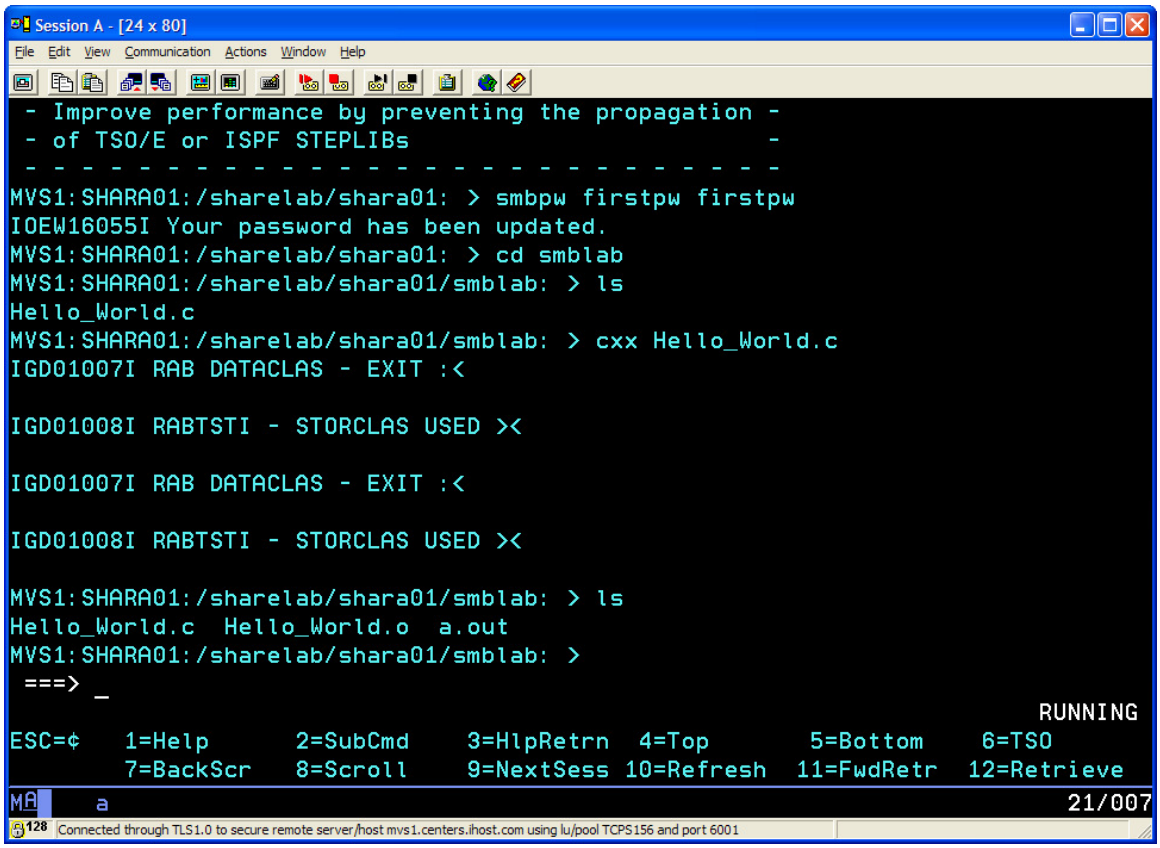
__ **24.** After the compile, you should see a screen something like this.



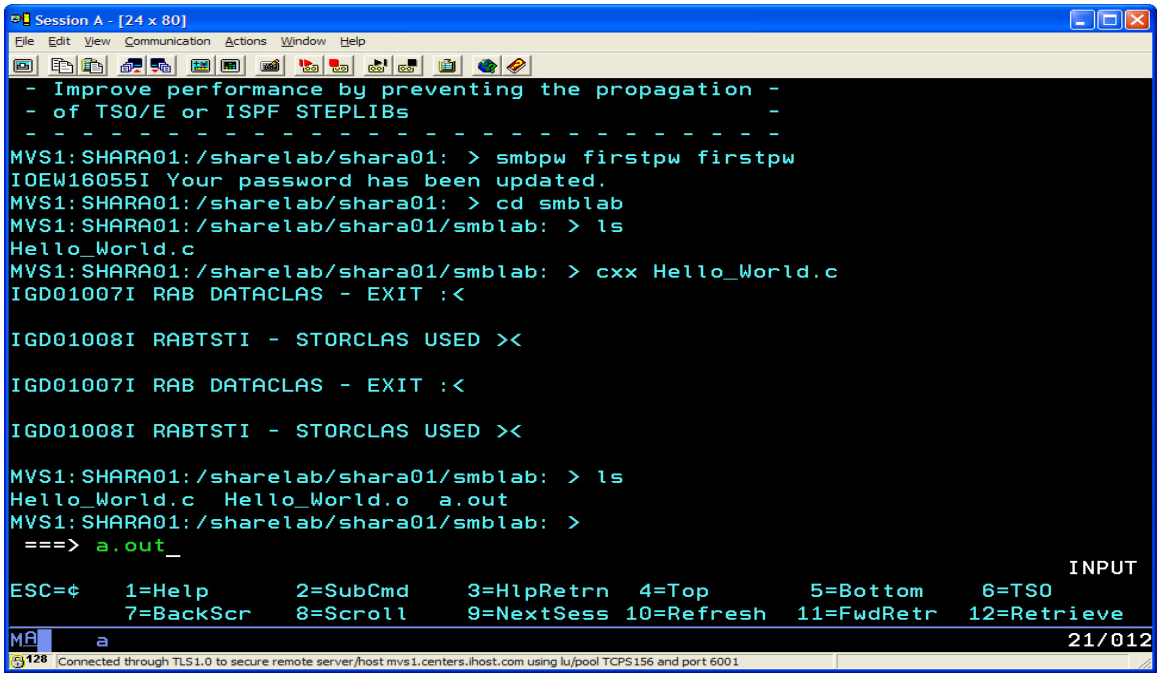
__ **25.** Now let's see what files were created. Type ls



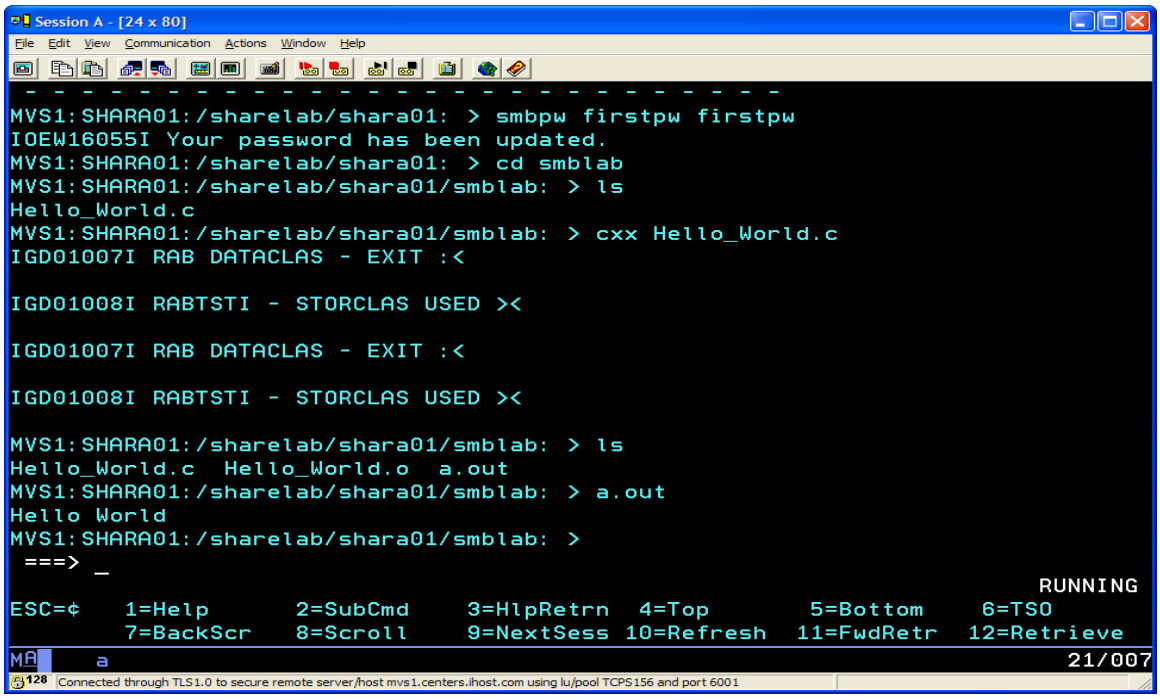
__ **26.** This should show you the files created by the compile command. You should see an **a.out** file. This is the executable created by the compiler.



__ **27.** Let's try executing the Hello_World program's executable (a.out) file. Type **a.out**

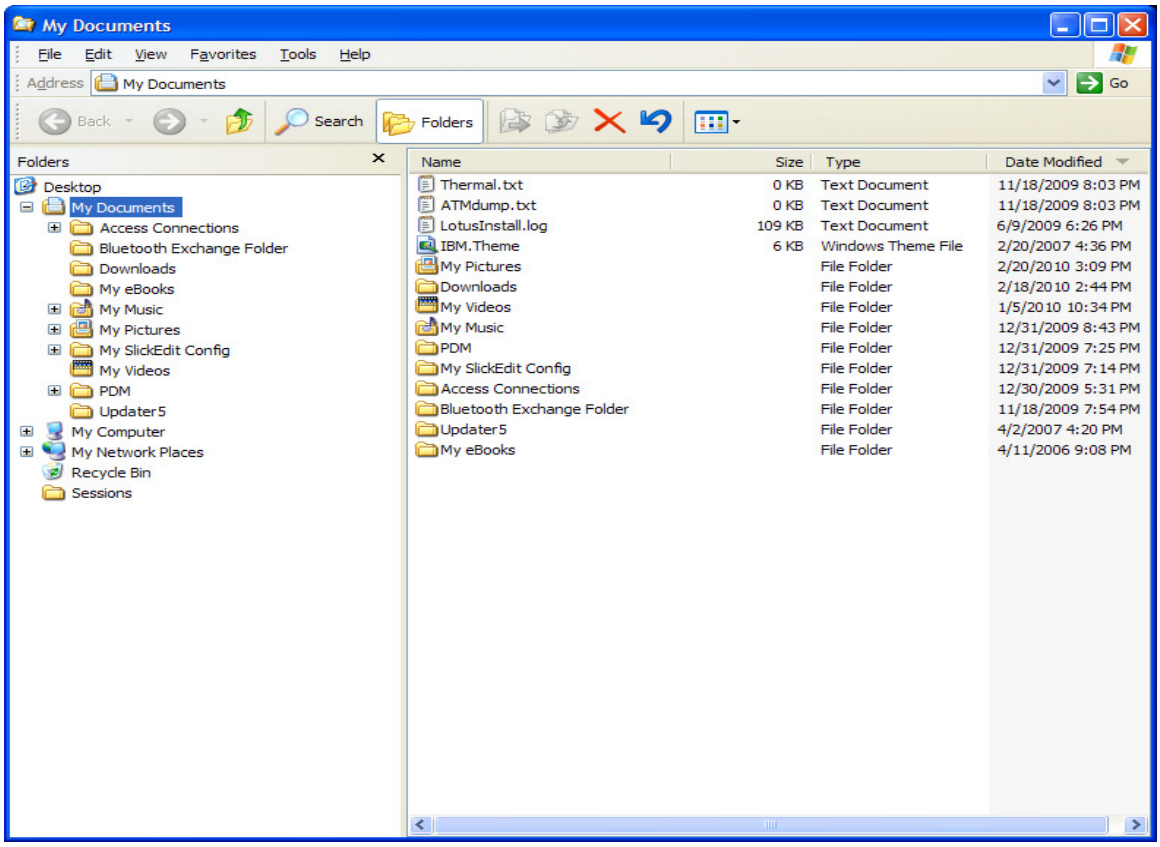


__ **28.** You should see the output of the Hello_World program. (A very impressive display of the words **Hello World**).

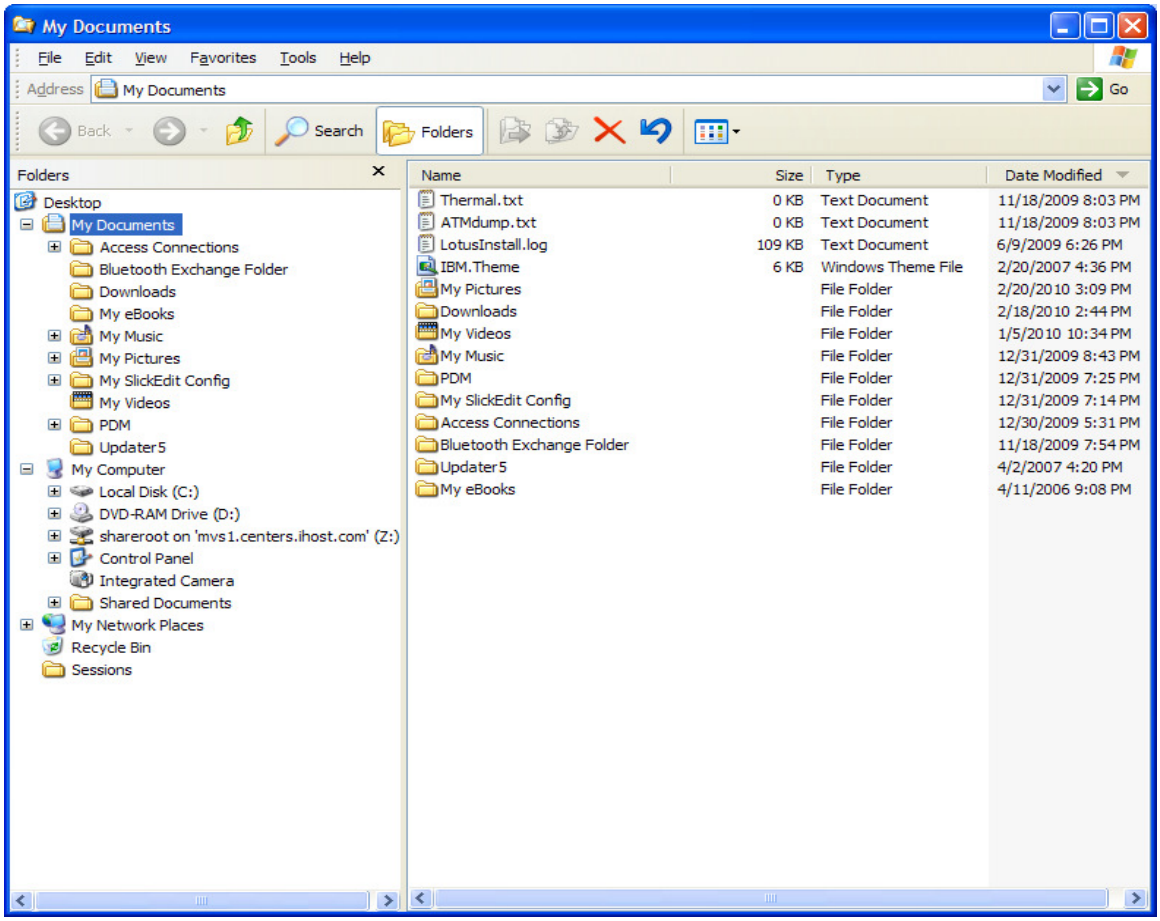


__ **29.** Now, we're going to try another use of the z/OS SMB server. We have a pdf file stored in the z/OS UNIX file system that we are going to directly access and display by simply double clicking on the file name from Windows Explorer.

Bring up a Windows Explorer screen.

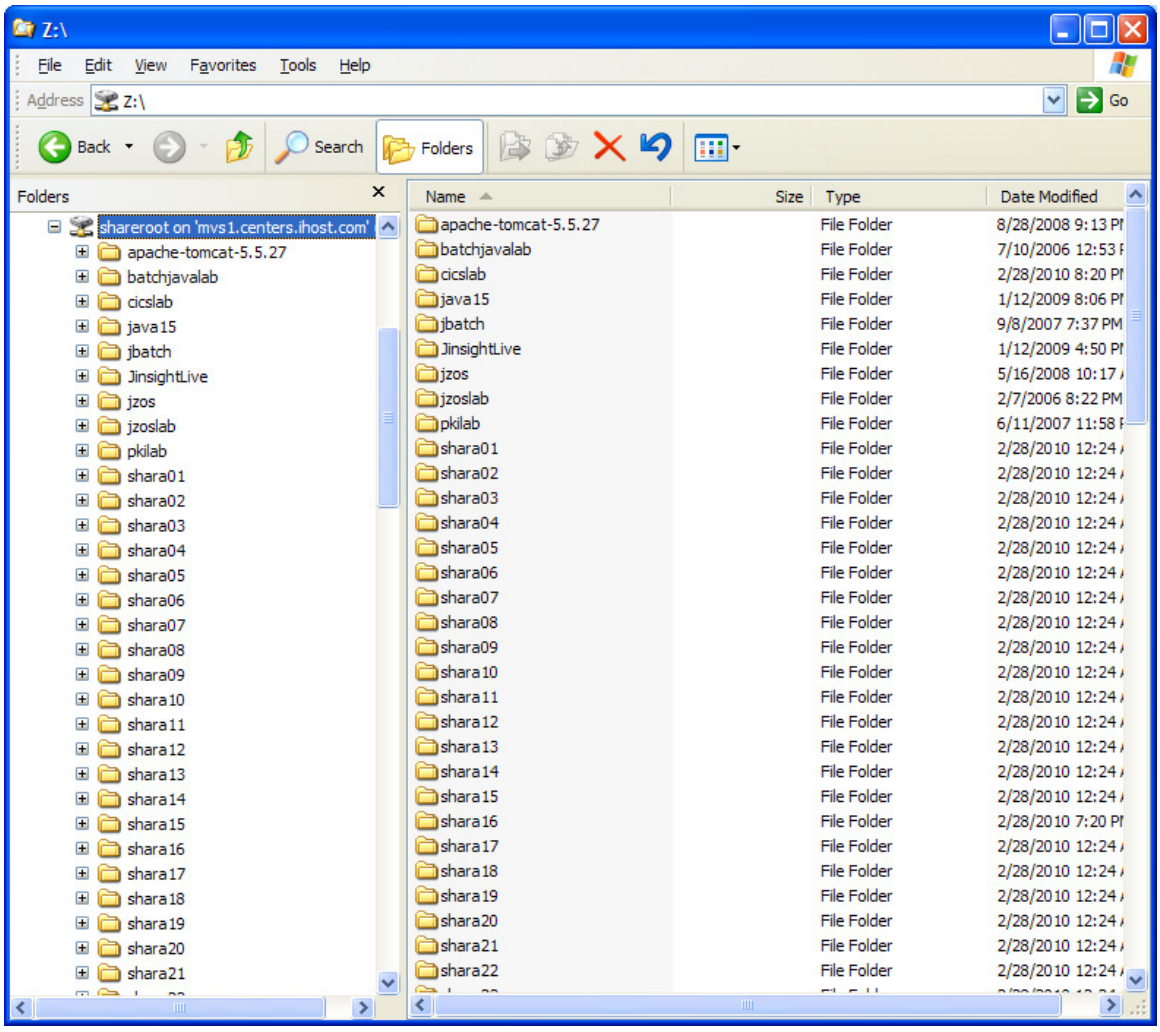


__ **30.** Now click on the + in front of My Computer on the left pane to expand what is in My Computer.

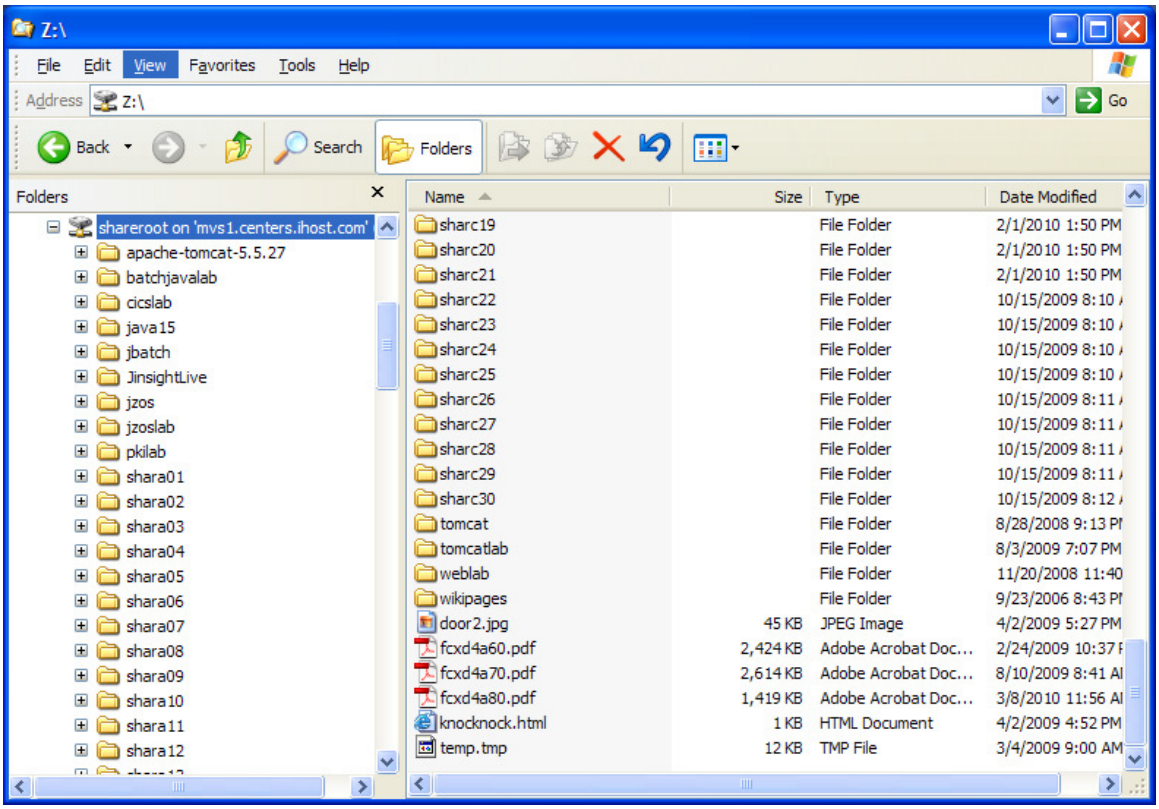


__ **31.** Now click on the entry in the left pane that says shareroot on mvs1.centers.ihost.com'(Z:)

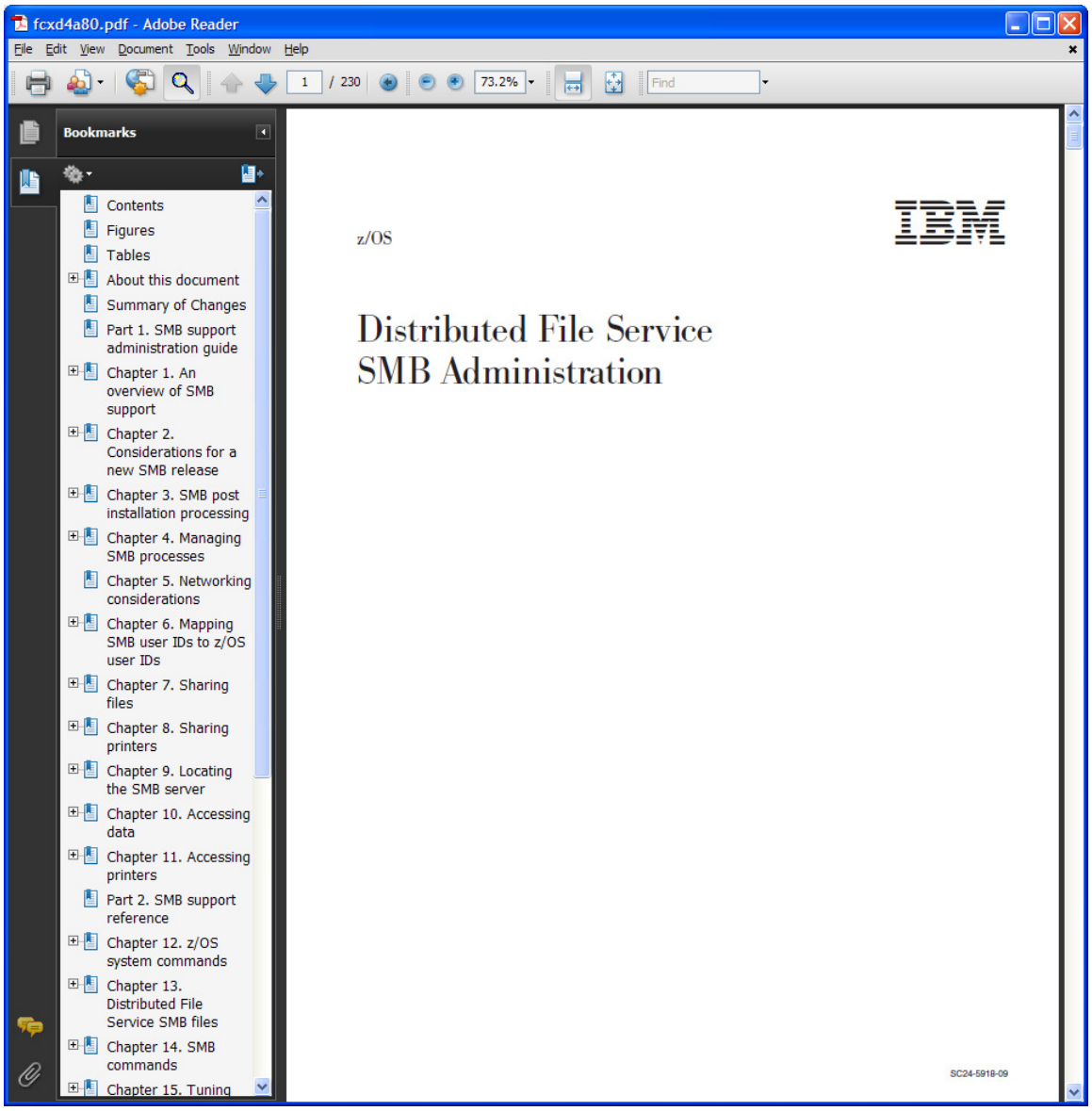
This may take a while because there are many directories to display – be patient.



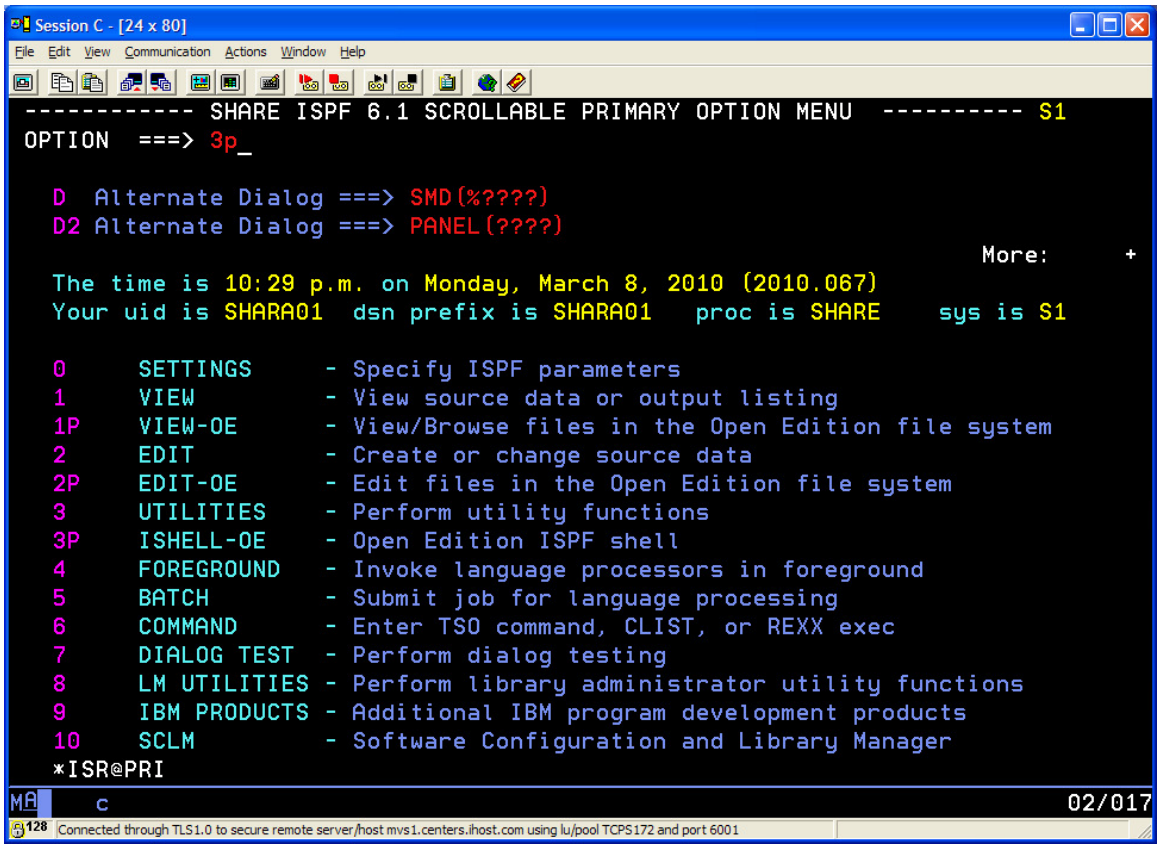
__ **32.** Scroll down using the scroll bar on the very right of the panel to the bottom of the list. You should see a file named fcxd4a80.pdf – Double click on this file.



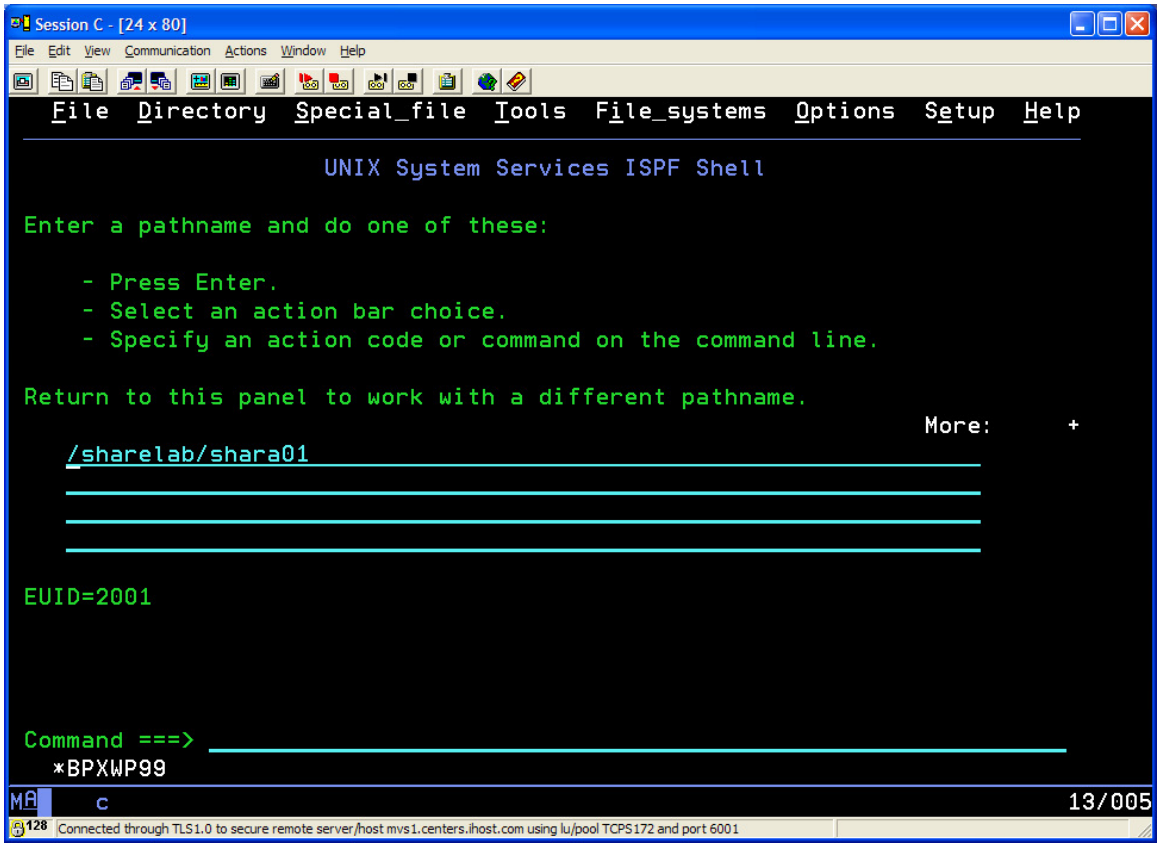
__ **33.** You should see the pdf displayed. (This is the z/OS Distributed File Service SMB Administration book – we are using the SMB server to retrieve its own book.)



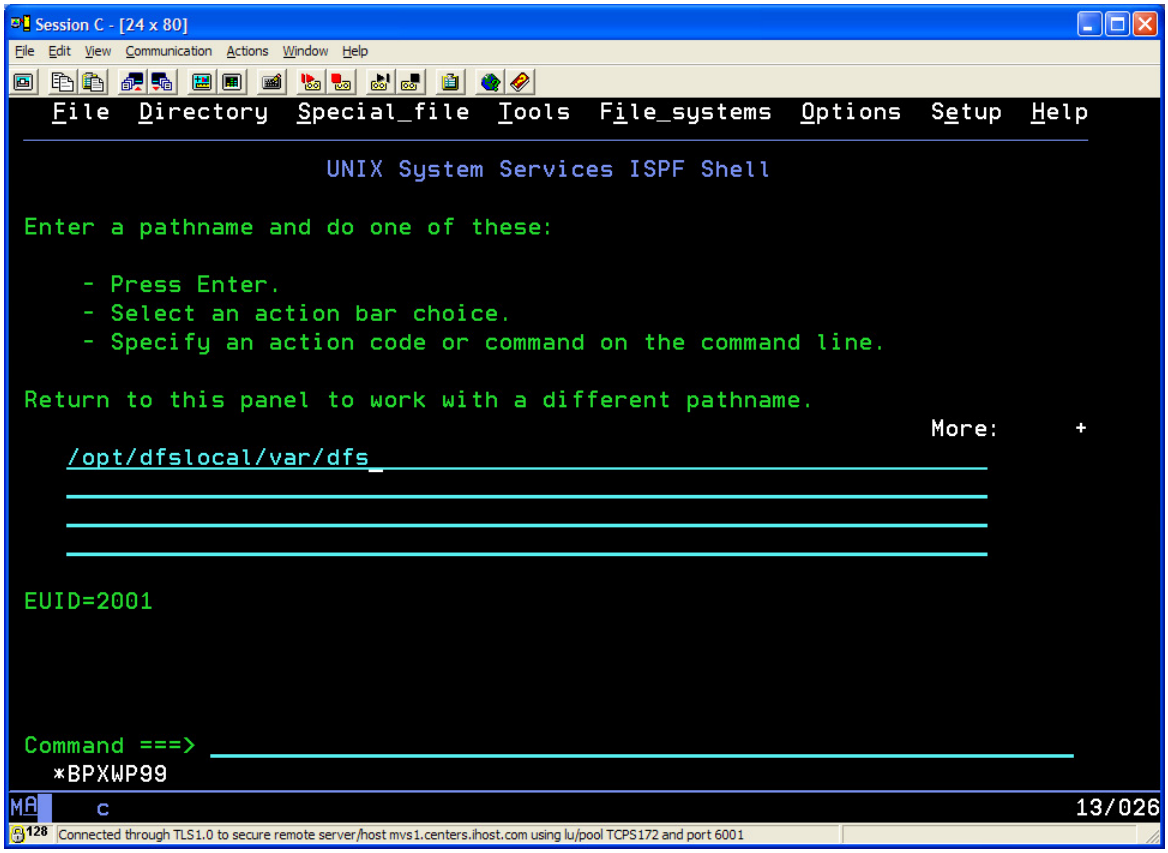
__ **34.** Now, let's take a look at some of the configuration files for the z/OS SMB server. Go back to your TSO session. If you are still in OMVS, type exit to get out of OMVS. Then type ISPF to get into ISPF. Once in ISPF, enter **3p** to get into the IShell.



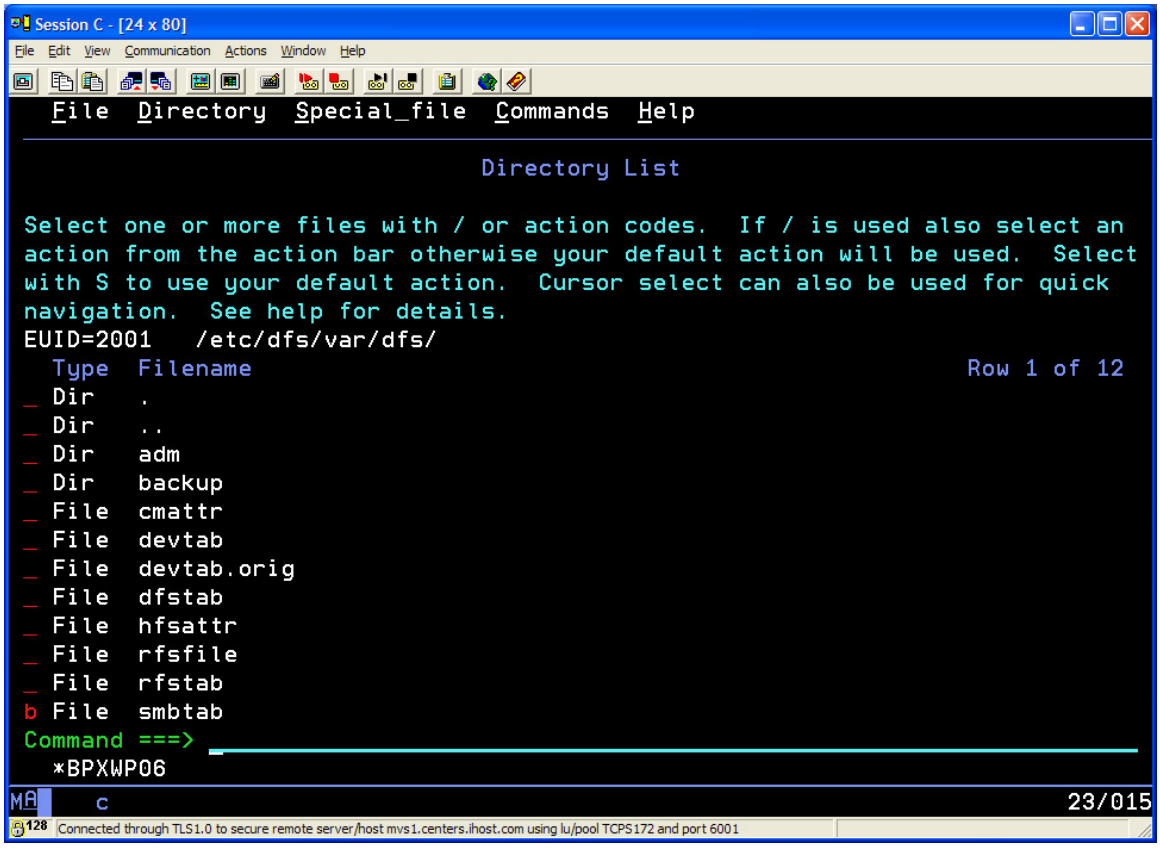
__ **35.** You will see the first IShell screen with the pathname filled in with your home directory.



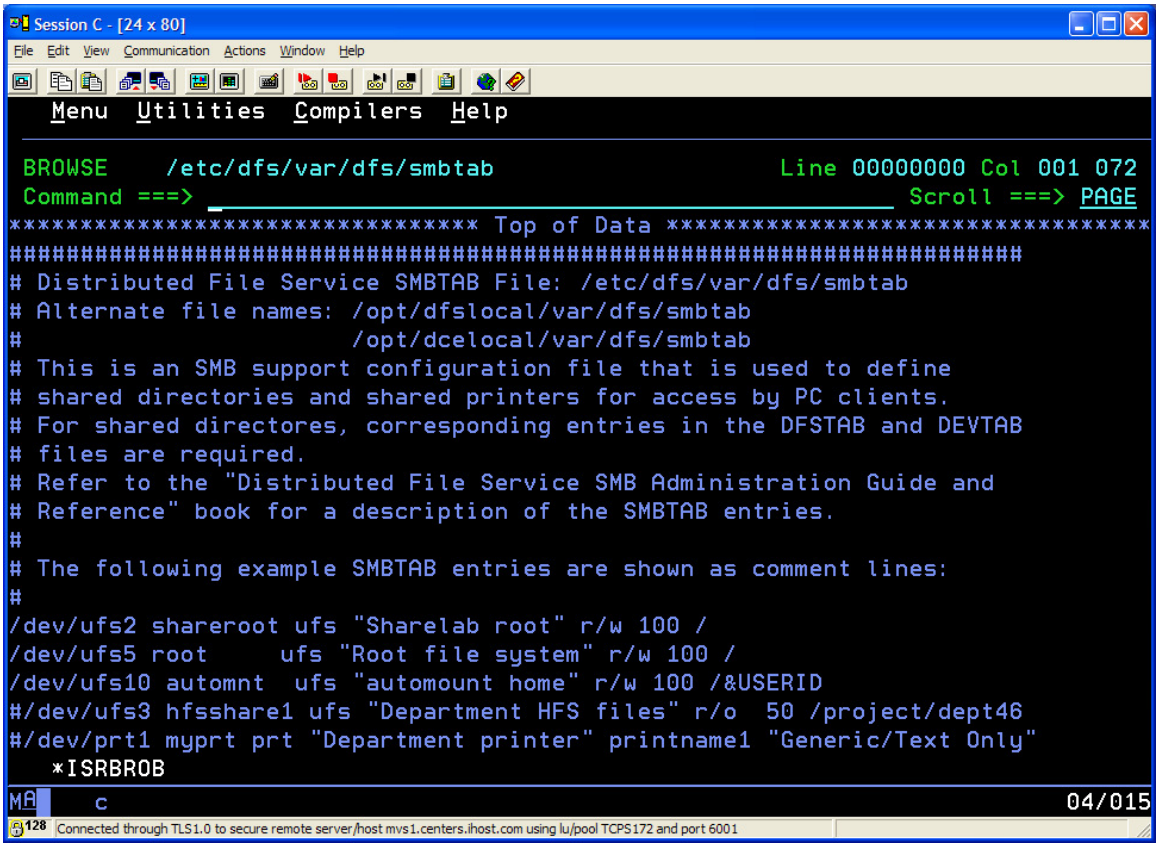
__ **36.** We are going to look at the z/OS SMB configuration files that control the data that is made available to PC users – the **smbtab**, the **dfstab** and the **devtab**. Enter the following path and hit enter - /**opt/dfslocal/var/dfs**



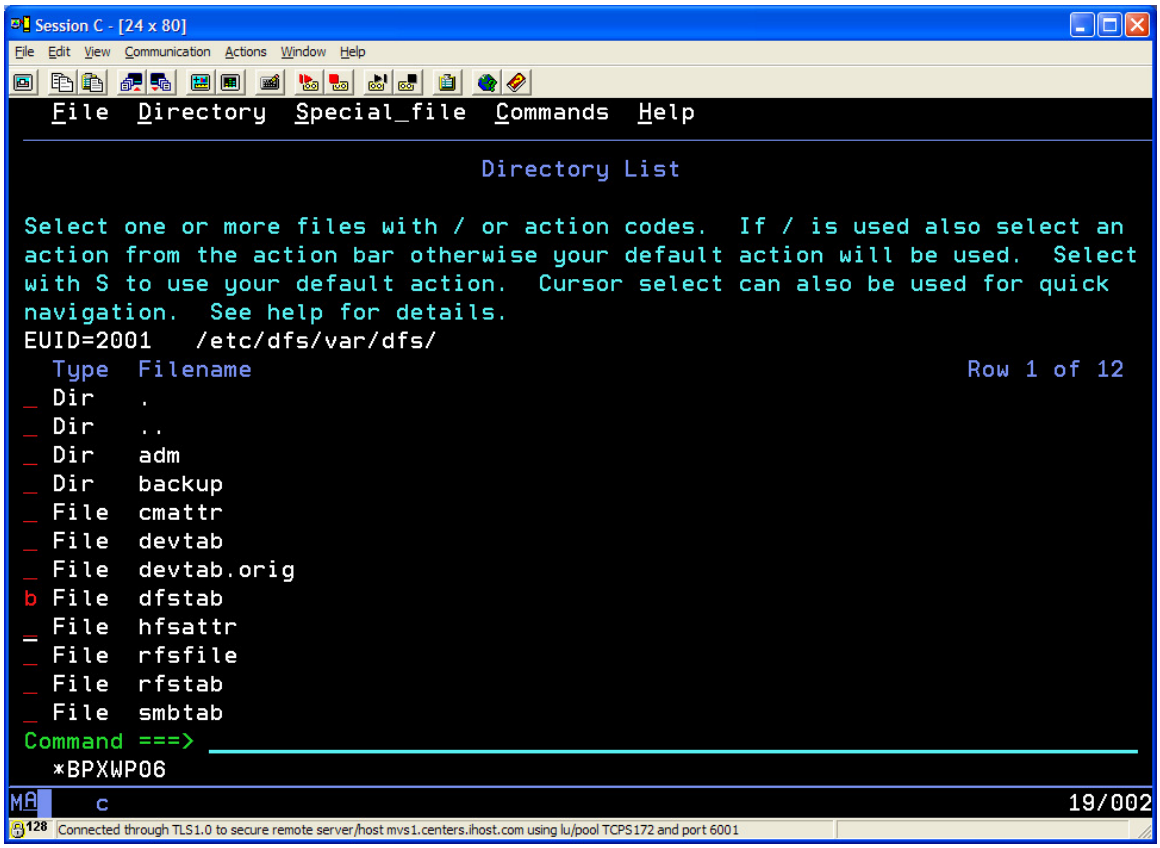
__ **37.** You will see the list of files in that directory. Place a b (for browse) by the smbtab and hit enter.



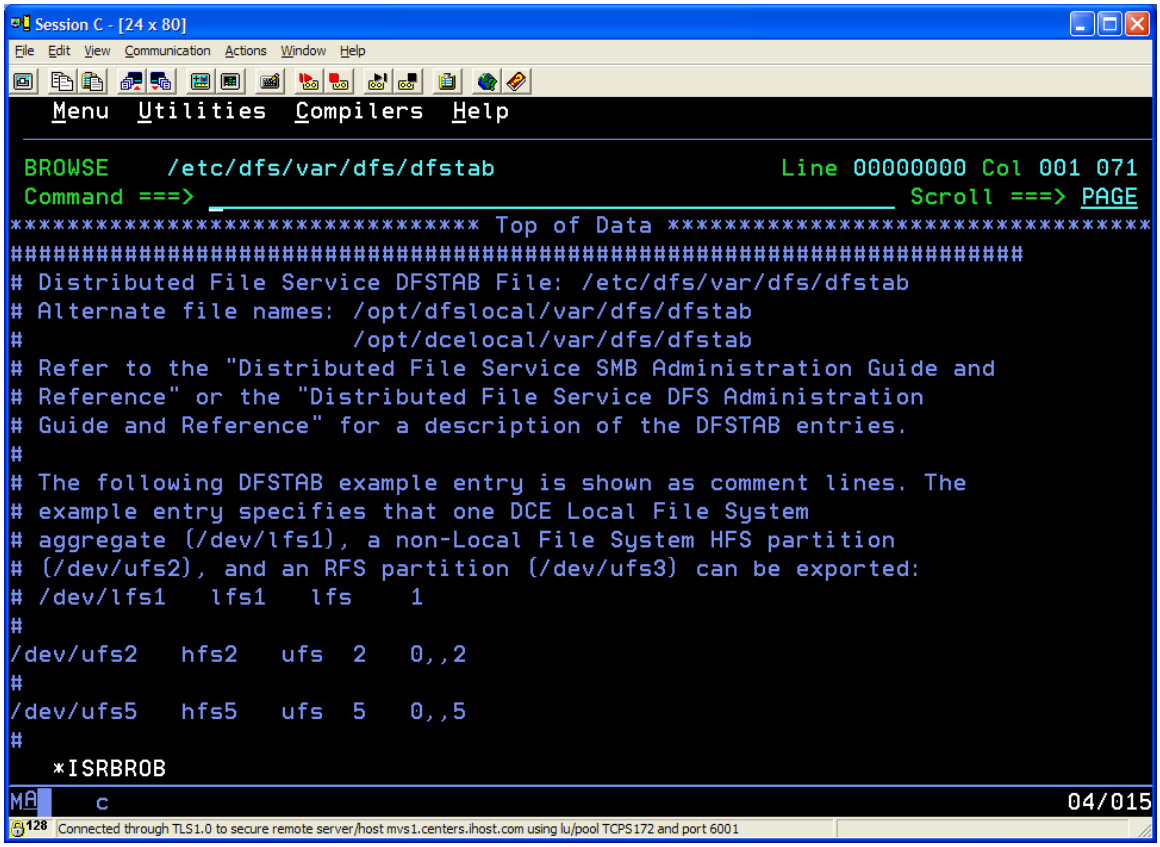
__ **38.** This is **smbtab** file. Notice the entry for shareroot. This is the sharename that you connected to. Also, notice the first field on that line (/dev/ufs2). The 2 is important. It is used to match the corresponding entries in the **dfstab** and the **devtab**. Now hit PF3 to go back to the file list.



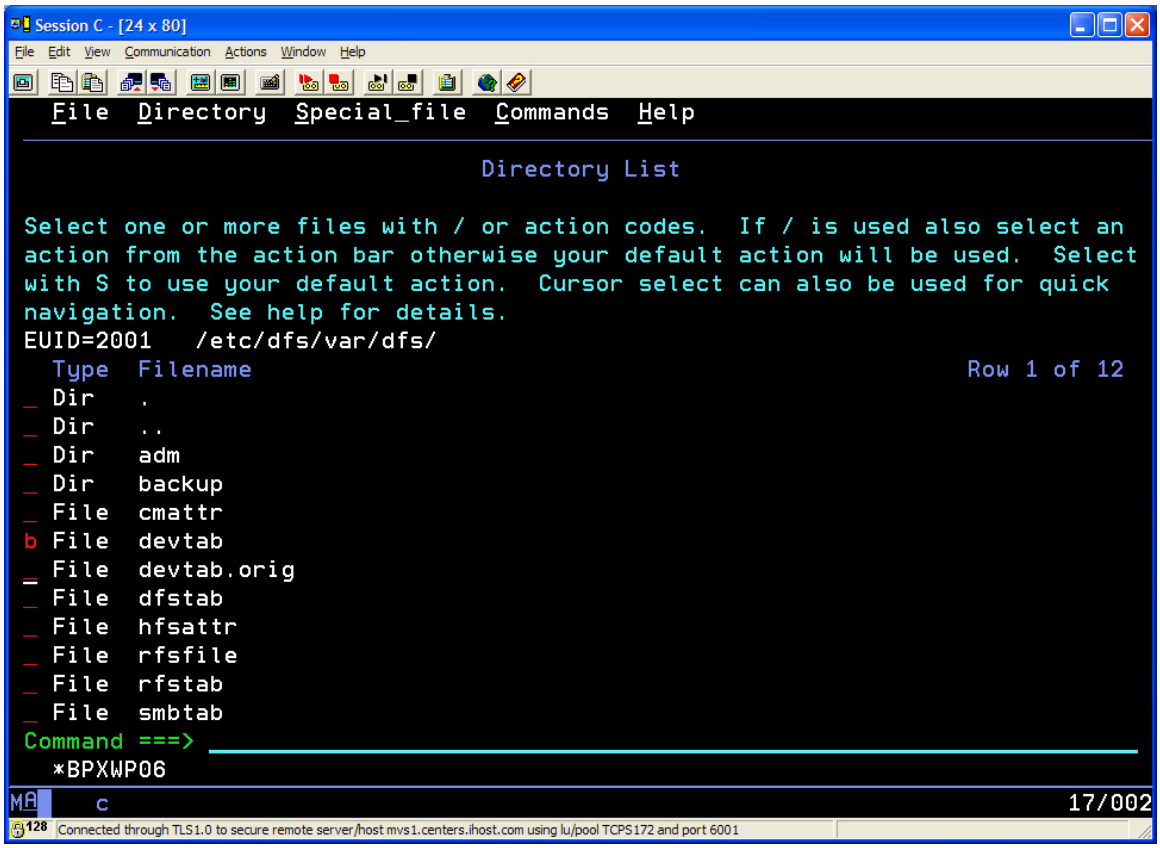
__ **39.** Place a b by the dfstab and hit enter.



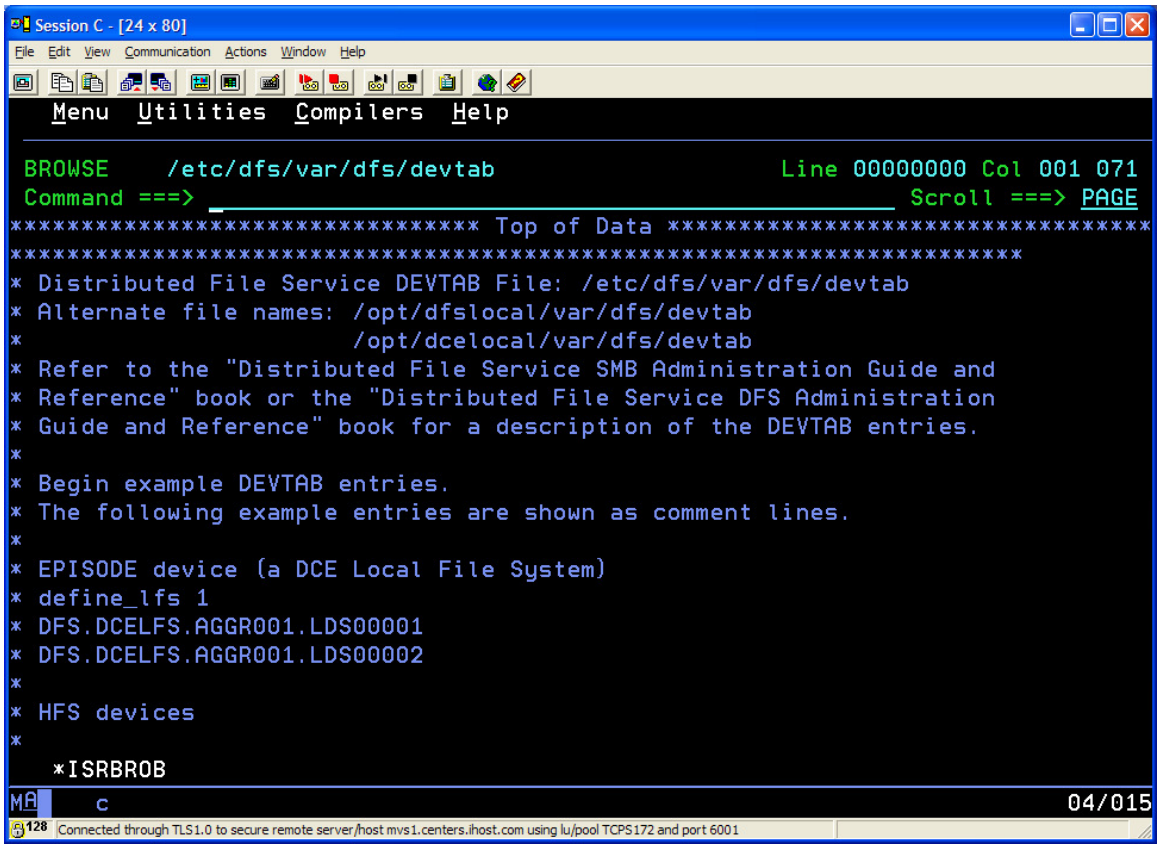
__ **40.** This is the **dfstab**. The entry for /dev/ufs2 is the corresponding entry to the shareroot entry in the smbtab shown previously. Now hit PF3 to go back to the file list again.



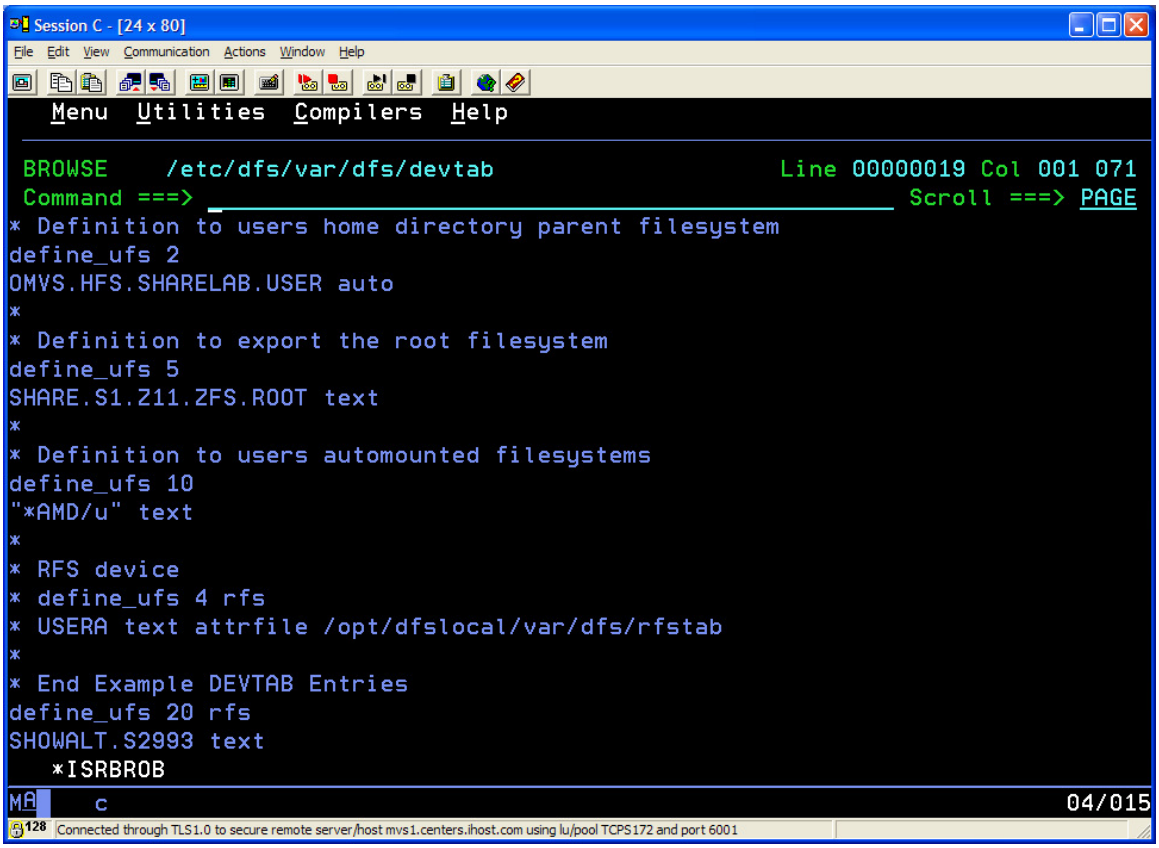
__ **41.** Place a b by the **devtab** and hit enter.



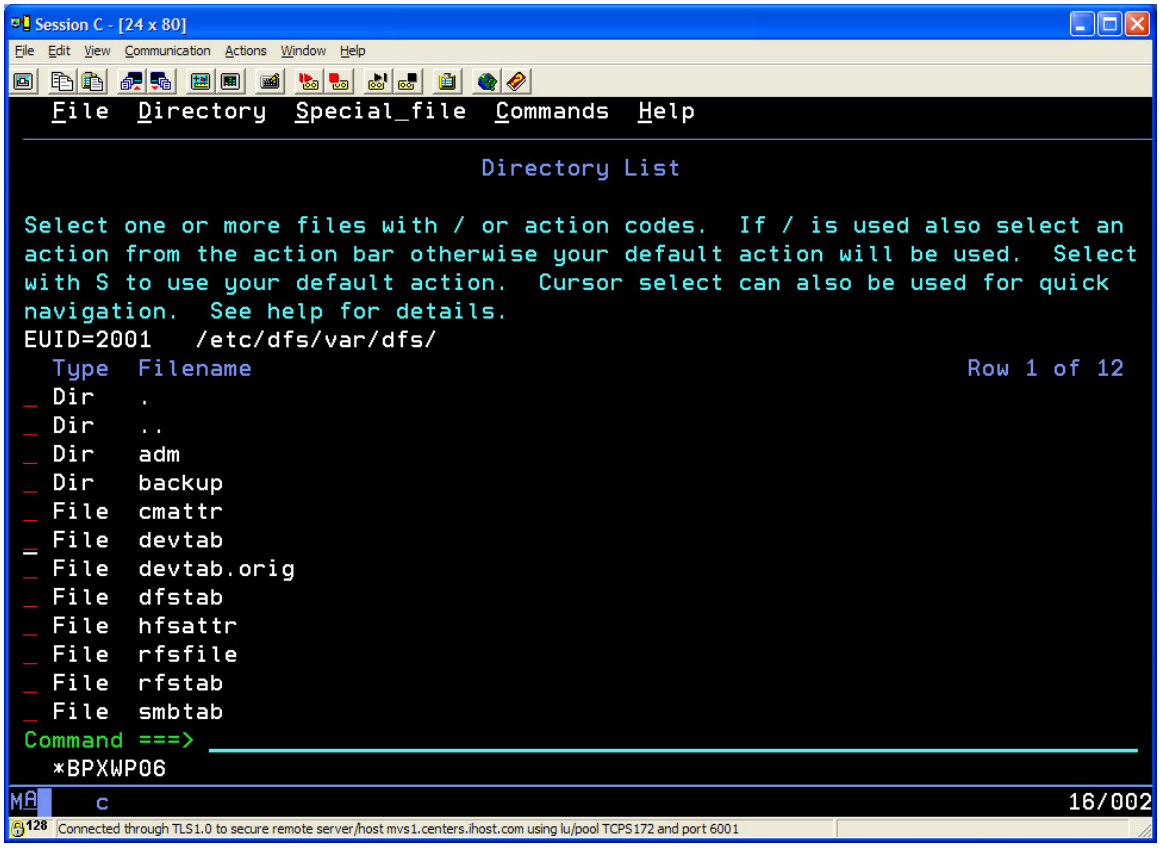
__ **42.** This is the **devtab**. Scroll to the next page using PF8.



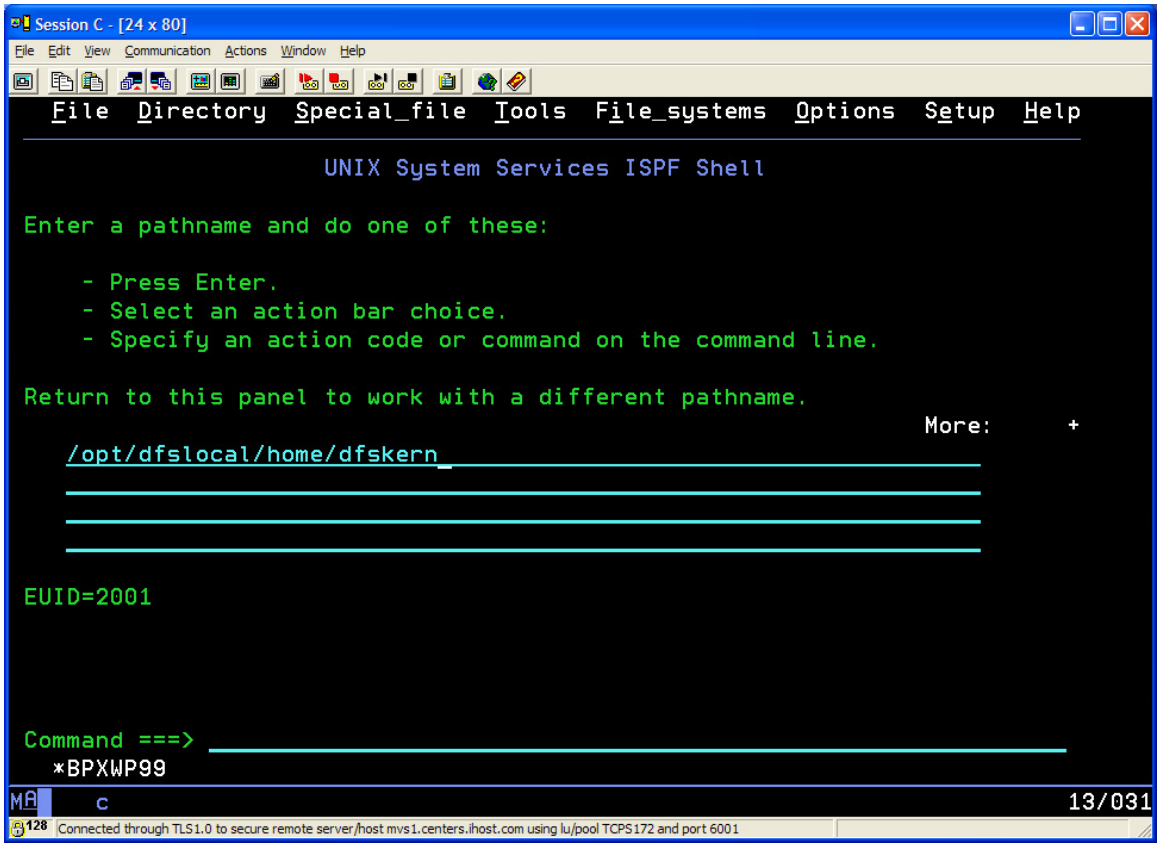
__ **43.** The define_ufs 2 entry in the **devtab** is on this page. It corresponds to the smbtab /dev/ufs2 entry. This devtab entry has the file system name to be exported on the next line (OMVS.HFS.SHARELAB.USER). On the same line, it has auto. This is the translation option for data in this file system. Press PF3 to go back to the file list.



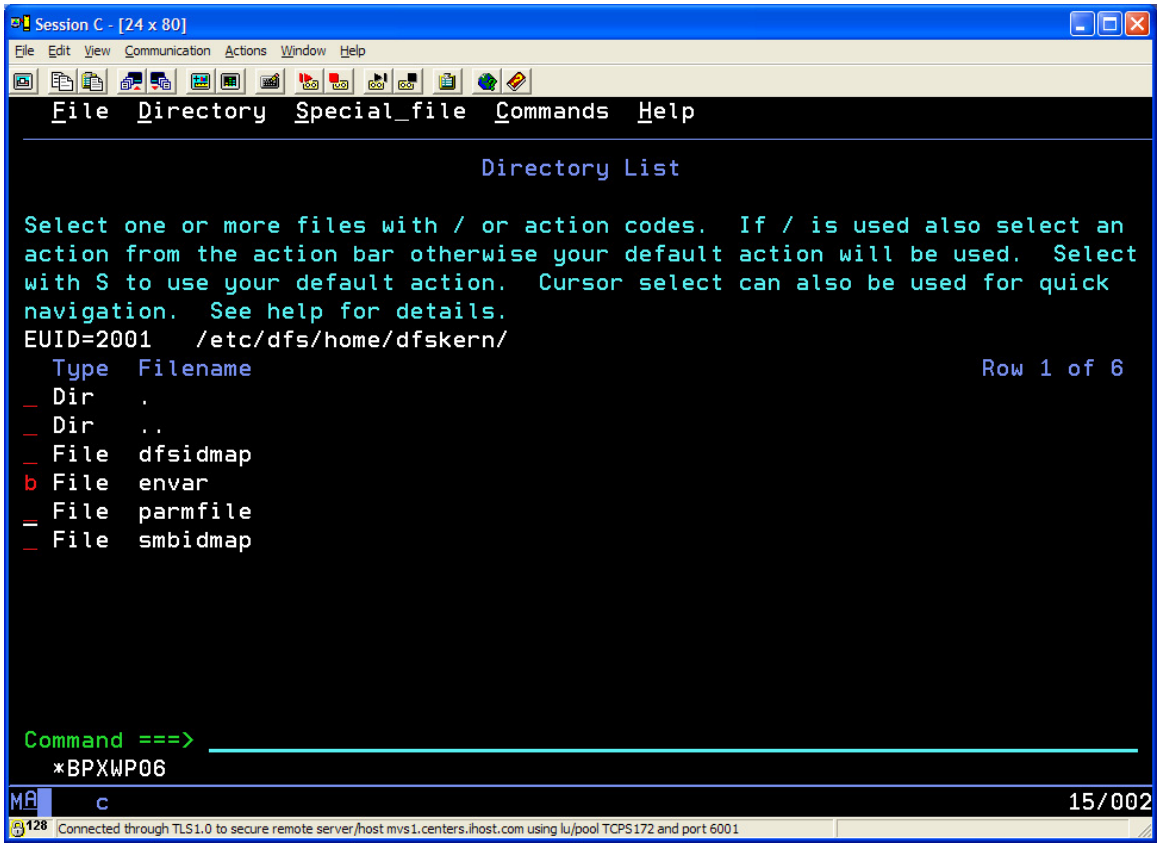
__ **44.** This is the file list screen again. Press PF3 again to go back to the main IShell screen.



__ **45.** This is the main IShell panel again. Type the following pathname - /opt/dfslocal/home/dfskern and hit enter.



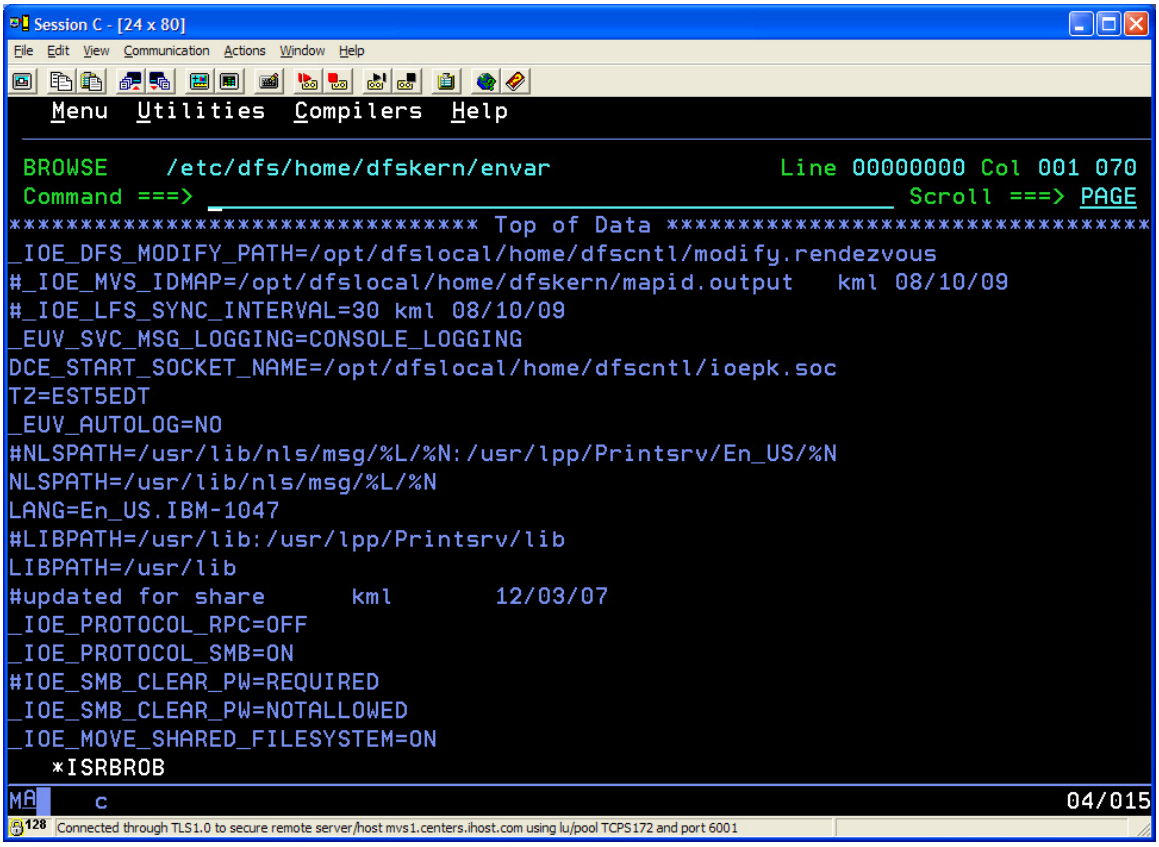
__ **46.** This directory contains the other main SMB server configuration file named **envar**. Place a b by the envar file and hit enter.



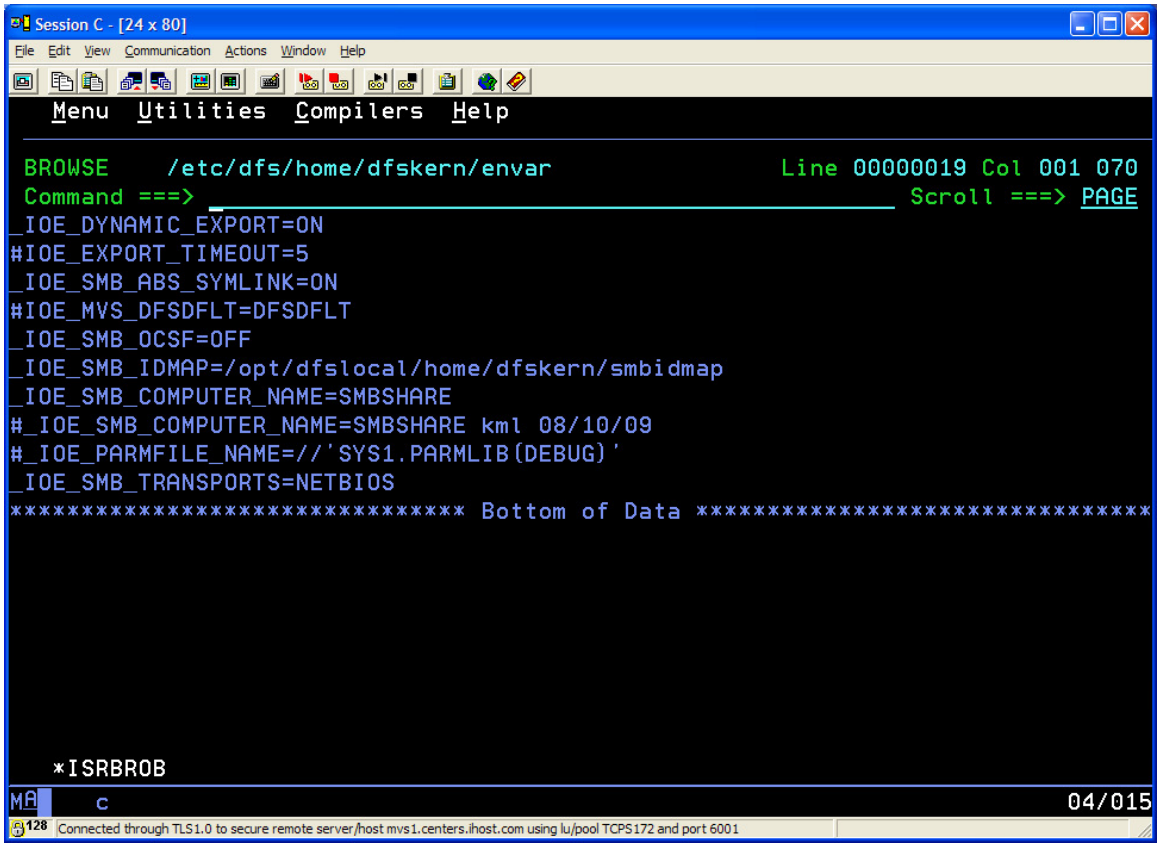
__ **47.** The envar contain configuration settings for the SMB server. Two important settings that are required to use the SMB server are

_IOE_PROTOCOL_RPC=OFF
_IOE_PROTOCOL_SMB=ON

The next setting (_IOE_SMB_CLEAR_PW) controls whether the SMB server uses clear passwords (REQUIRED) or encrypted passwords (NOTALLOWED). This SMB server is using encrypted passwords. Scroll to the next page using PF8.



__ **48.** This screen contains the _IOE_DYNAMIC_EXPORT=ON setting.



__ **49.** End of lab exercises. You can use PF3 to get all the way out of ISPF and then logoff.

z/OS Distributed File Service SMB server Overview

The Distributed File Service Server Message Block (SMB) server provides support that makes Hierarchical File Systems (HFS/zFS) files and data sets available to SMB clients. Throughout this presentation, references are made to HFS. Unless otherwise stated, HFS is a generic reference that includes HFS, zFS, TFS, and AUTOMNT file system data.

The data sets supported include sequential data sets, partitioned data sets (PDS), partitioned data sets extended (PDSE) and Virtual Storage Access Method (VSAM) data sets. The data set support is usually referred to as Record File System (RFS) support.

The SMB protocol is supported through the use of TCP/IP on z/OS. This communication protocol allows clients to access shared directory paths and shared printers.

Personal Computer (PC) clients on the network use the file and print sharing functions that are included in their operating systems.

The z/OS SMB server runs in its own server address space and there can be only one instance running per z/OS system. Due to system constraints, only one instance of the server will be running for this lab exercise. You will not be configuring a server of your own today, but the review of the configuration should be adequate to allow you to configure the server at your installation.

SMB Access to z/OS UNIX data

z/OS UNIX data is made available through the z/OS SMB server to PC clients by exporting file systems and creating shares.

File systems are exported by specifying them in the dfstab and devtab files. (They are located in /opt/dfslocal/var/dfs.) These files are read during the z/OS SMB server initialization. They are also read as a result of executing the dfsexport command. The main thing specified in dfstab and devtab is the name of the file system you want to make available to PC users.

The other specification needed is the share. The share is specified in the smbtab file (located in the same directory). The smbtab entry specifies the share name and a directory within a file system that the share name represents. The share name is used on the PC to connect the PC drive letter to the directory in the file system.

Static Export/Dynamic Export

Dynamic export allows the SMB server to discover a “lower” z/OS UNIX file system even though it has not be specified in a dfstab/devtab file (that is, it has not been statically exported).

Determine whether you intend to use the dynamic export capability. It is controlled by the _IOE_DYNAMIC_EXPORT environment variable of dfskern. The default is OFF meaning that dynamic export is not enabled. Dynamic export allows the SMB server to support file systems mounted by using the z/OS Automount Facility.

If you attempt to cross into a lower file system (for example, by issuing the cd command on the PC to the mount point directory) and that lower file system has not been exported (neither statically with a dfstab/devtab specification, nor allowing dynamic export to occur), you will see the files and directories that existed in the directory before it became a mount point (that is, you will see “under” the mount point).

The _IOE_SMB_ABS_SYMLINK environment variable of dfskern is also related to dynamic export. If you enable the ability for the SMB server to follow absolute symbolic links (ones that begin with /), you probably want to enable dynamic export to ensure that the file system of the target of the symbolic link is exported.

SMB authentication methods

Define SMB users by modifying the smbidmap file identified by the _IOE_SMB_IDMAP environment variable of dfskern. This file will contain a mapping of SMB users to local z/OS users.

Follow the applicable instructions in Chapter 6, “Mapping SMB user IDs to z/OS user IDs”. In addition, z/OS users that do not use DCE, should put the following line in their HFS .profile file in their home directory: export _EUV_AUTOLOG=NO. Alternatively, if no z/OS users are using DCE, the above line can be placed in /etc/profile. It is then set for all z/OS UNIX users.

Determine which authentication method you plan to use. The available methods are:

- ClearText - Password must match the password for the MVS user id that the PC user is mapped to. (_IOE_SMB_CLEAR_PW=REQUIRED)
- Encrypted - Password must match the password stored in the DCE segment of the MVS user id that the PC user is mapped to. An MVS user can store a password in their DCE segment from OMVS using the smbpw command. (_IOE_SMB_CLEAR_PW=NOTALLOWED)
- Passthrough Authentication – Password must match the password of the PC user that is stored in the Windows Domain Controller. Users that are not in the domain and that fail the domain authentication will additionally attempt local (encrypted password) authentication.
  _IOE_SMB_AUTH_SERVER=n.n.n.n
  _IOE_SMB_DOMAIN_NAME=
  _IOE_SMB_AUTH_SERVER_COMPUTER_NAME=

Determine whether you intend to allow guest users. Guest users are PC users that have (limited) access to files and printers on the SMB server without identifying themselves. Guest users are allowed when the _IOE_MVS_DFSDFLT environment variable in the dfskern process is set to a valid z/OS user ID. Guest users can access any data or files that z/OS user ID can access. If guest users are allowed, users that specify an incorrect password or no password become the

guest user ID. It is better to disallow guest users until you are certain you need this capability and that it meets your security guidelines.

The view from the PC

The Windows net use command can be issued from a Command Prompt window. The following net use command will get you connected:

```
net use * \\mvs1.centers.ihost.com\shareroot
firstpw /user:shara01 /persistent:no
```

- net use is the command
* means assign an available drive letter
- \\mvs1.centers.ihost.com is the host name of the z/OS system
- shareroot is the share name of the share
- firstpw is the password for the userid
- shara01 is the PC userid to be mapped to an MVS userid (use your assigned number)
- persistent:no means that connections are not saved over logons

Hint: If you use PC Help and search for “command-line”, you should be able to find the **Command-line reference A-Z** and then the **Net service commands** and then the syntax of the **net use** command. Alternatively, you can type **net help use** in a Command Prompt window.

Once you connect and the drive letter is assigned by Windows, you can use that drive letter to access z/OS data.

Other issues:

- ASCII (ISO8859-1) vs EBCDIC (IBM-1047)
  controlled by dfskern envar _IOE_HFS_TRANSLATION
- end of line characters for text files
  controlled by the application
  Windows uses carriage return/line feed (two characters)
  z/OS UNIX uses newline (one character)
- filename suffix
  helps Windows understand the type of file

iconv on z/OS UNIX can be used to convert a file if it is accidentally translated when it shouldn’t be or is not translated when it should be.

To convert an ASCII file to EBCDIC
iconv –f ISO8859-1 –t IBM-1047 sourcefile_ascii > targetfile_ebcdic

To convert an EBCDIC file to ASCII
iconv –f IBM-1047 –t ISO8859-1 sourcefile_ebcdic > targetfile_ascii

z/OS SMB server install and configuration steps

To install, configure, and access the Distributed File Service server (dfskern) for SMB File/Print Server operation, you must perform the following administrative steps:

Install and perform post-installation of the Distributed File Service by following the applicable instructions in the z/OS Program Directory for FMID H0H23x0 (R7 is H0H2370; R8 is H0H2380; R9 is H0H2390; R10 is H0H23A0, R11 is H0H23B0) in Wave 1G.

Note that Distributed File Service is a base element of z/OS that provides three different services:

- DCE DFS client and server support
- SMB file/print server support
- zSeries File System (zFS) support

Ensure that the target and distribution libraries for the Distributed File Service are available.

Run the prefix.SIOESAMP(IOEISMKD) job from UID 0 to create the symbolic links used by the Distributed File Service. This job reads the member prefix.SIOESAMP(IOEMKDIR) to delete and create the symbolic links.

Ensure that the DDDEFs for the Distributed File Service are defined by running the prefix.SIOESAMP(IOEISDDD) job.

Install the Load Library for the Distributed File Service. The Load Library (hlq.SIOELMOD) must be APF authorized and must be in link list.

Install the samples (hlq.SIOESAMP).

If you plan to use encrypted passwords (recommended) and optionally, you want to exploit OCSF and hardware encryption, you must ensure that the proper authorizations have been given to the

DFS server user ID to use OCSF services. Refer to the "Cryptographic Services OCSF Customization Considerations" section in the z/OS Program Directory, GI10-0670 and the "Configuring and Getting Started" section in the z/OS Open Cryptographic Services Facility Application Programming, SC24-5899 for information on this topic.

If you are using ICSF, you may need to PERMIT the user ID DFS READ access to the profiles in the CSFSERV general resource class. Refer to the z/OS Cryptographic Services ICSF Administrator's Guide, SA22-7521 for more information on the CSFSERV resource class.

The SMB server uses an Event Notification Facility (ENF) exit for event code 51 (allocation contention). When this event occurs, an SRB is scheduled to queue a request to the SMB server. If the SMB server is at too low a dispatching priority, the requests may become backed up and the system may eventually run out of resources. The SMB server should have a dispatching priority that is sufficiently high to allow these requests to be processed in a timely manner.

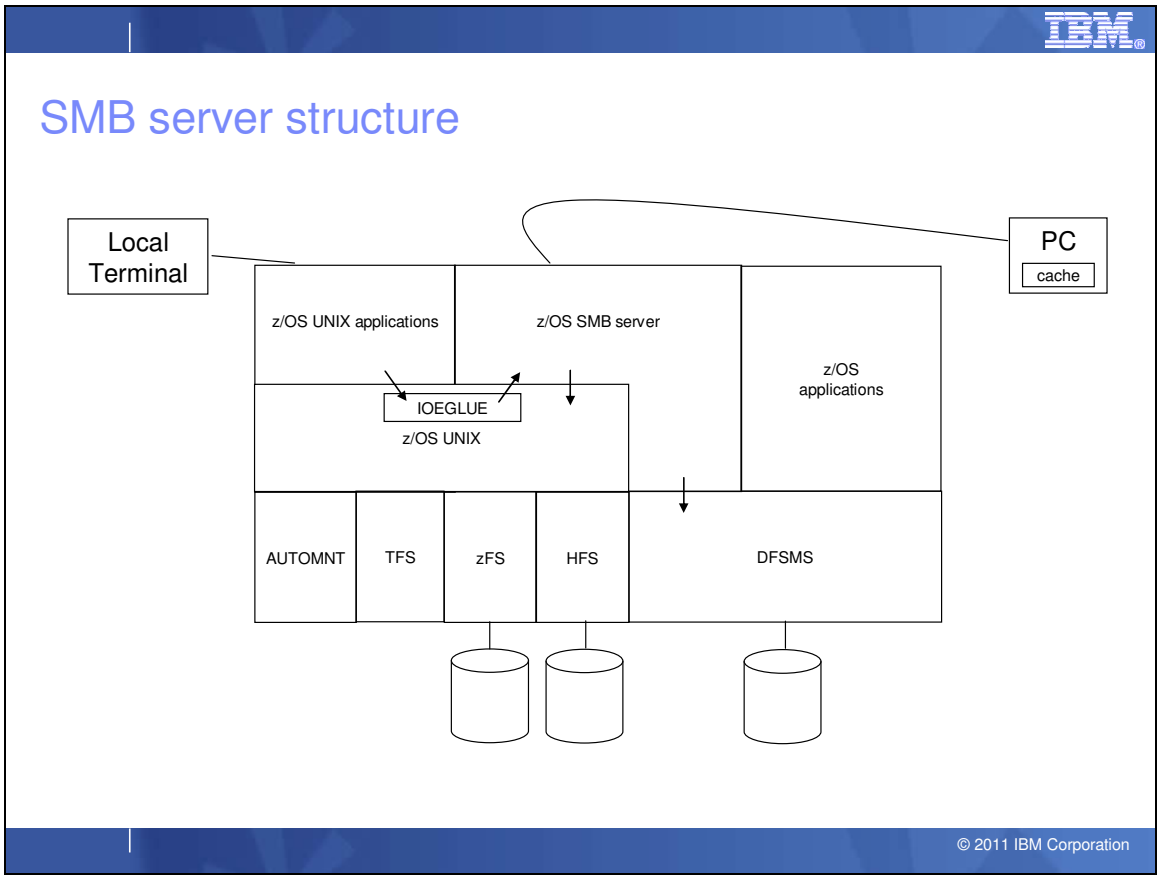
Define administrators on the host system following the applicable instructions in “Defining SMB administrators”. Refer to the zOS Distributed File Service SMB Administration Guide.

Create the default DFS configuration files using the /opt/dfsglobal/scripts/dfs_cpfiles shell script, if they were not created during the installation process.

These configuration files, required by SMB File/Print Server, are usually created before the Distributed File Service installation is verified by the dfs_cpfiles shell script, as indicated in the z/OS Program Directory.

Refer to the zOS Distributed File Service SMB Administration Guide for further details regarding the dfs_cpfiles script.

SMB server structure



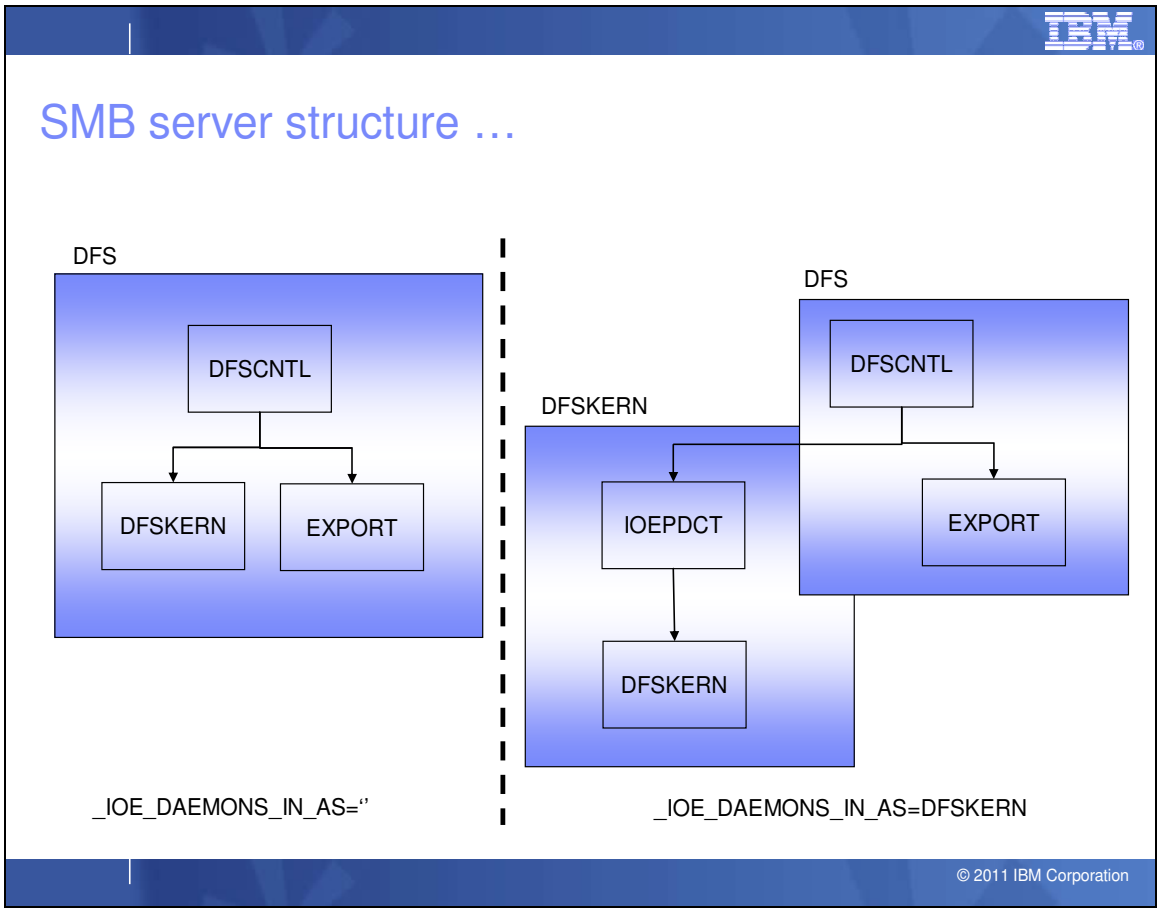
The SMB server runs as an authorized started task in its own address space. It accesses z/OS UNIX files on behalf of PC users. It can also access MVS data sets on behalf of PC users.

As part of its initialization, it asks z/OS UNIX to load the IOEGLUE module into the z/OS UNIX address space. IOEGLUE is part of the z/OS SMB server that runs in the z/OS UNIX address space. IOEGLUE allows the SMB server to know about “local” access of z/OS UNIX files. Why does the SMB server need to know about local access of z/OS UNIX files? Consider the following scenario:

When a PC user reads a z/OS UNIX file via the z/OS SMB server, the data is sent through z/OS UNIX, to the SMB server and it is sent to the remote PC via TCP/IP. The remote PC can cache the data. That is, the PC keeps a temporary copy of it. If the data is read again at the PC, there is no need to communicate to the z/OS SMB server.

The data can be returned to the PC application from the cache. Now, let's say that a local user accesses the same z/OS UNIX file to update it. When the update request comes into z/OS UNIX, z/OS UNIX recognizes that the request is for a file that is contained in a file system that is exported by the z/OS SMB server. Because of this, z/OS UNIX calls the IOEGLUE module which in turn calls the z/OS SMB server. The z/OS SMB server sends a revoke request to the PC which tells it to throw away the data in the PC cache. The next time the PC application attempts to read that data, the PC will need to communicate with the z/OS SMB server and it will access the updated data. In this way, the z/OS SMB server keeps the PC cache consistent (it sees the latest data).

SMB server structure …



The DFSKERN process (which is the main server process) can run in the DFS address space (the default - shown on the left) or in its own address space (shown on the right). IBM recommends that you run DFSKERN in its own address space. This is accomplished by specifying _IOE_DAEMONS_IN_AS=DFSKERN in the /opt/dfslocal/home/dfscntl/envar environment variable. The DFSKERN JCL must also be available.

The IOEPDCF configuration file controls which processes are started by the DFSCNTL process. This configuration file does not need to be modified.

The sample JCL for the DFS PROC and the DFSKERN PROC are in the IOE.SIOESAMP data set.

DFSKERN Envar file

Modify the /opt/dfslocal/home/dfskern/envar file to activate SMB File/Print Server by setting the environment variable (envar) _IOE_PROTOCOL_SMB=ON.

Assuming that you do not want to also use DCE DFS file serving, you should also set the environment variable _IOE_PROTOCOL_RPC=OFF in the file /opt/dfslocal/home/dfskern/envar.

Dynamic export is controlled by the _IOE_DYNAMIC_EXPORT envar. We talked about this in slide 6.

SMB authentication is controlled by the _IOE_SMB_CLEAR_PW envar and the _IOE_SMB_IDMAP envar. We talked about these in a prior slide.

DFSKERN Envar file …

If you are using OCSF, ensure that the /opt/dfslocal/home/dfskern/envar file has a LIBPATH that adds the directory that contains the OCSF DLLs. Be sure that the directory added is the directory indicated in the z/OS Open Cryptographic Services Facility Application Programming.

If you are using the print capability of the SMB File/Print Server, ensure that the Infoprint Server is installed and customized by following the applicable instructions in the z/OS Program Directory. In addition, ensure that the /opt/dfslocal/home/dfskern/envar file has a LIBPATH entry that adds the directory that contains the Infoprint Server DLLs. Be sure that the directory added is the directory indicated in the “Infoprint Server Customization Considerations” section of the z/OS Program Directory.

For example, a LIBPATH that specifies both the OCSF DLL directory and the Infoprint Server DLL directory might look like :

LIBPATH=/usr/lib:/usr/lpp/Printsrv/lib.

There is a relationship between number of threads specified for the SMB server and the maximum number of threads that z/OS UNIX allows in a process. The following DFSKERN envars have an effect on the number of threads created for the SMB server.

_IOE_RFS_WORKER_THREADS
_IOE_SMB_CALLBACK_POOL
_IOE_SMB_MAIN_POOL
_IOE_TKMGLUE_SERVER_THREADS

There are also a number of dynamically created DFSKERN threads (approximately 25). The total of the DFSKERN threads must be less than the z/OS UNIX MAXTHREADS specification in the BPXPRMxx. If this is not the case, DFSKERN can abend during thread creation. The number of z/OS UNIX MAXTHREADS can be increased using the SETOMVS MAXTHREADS=nn operator command. The number of z/OS UNIX MAXTHREADS can be displayed using the D OMVS,O

operator command. See *z/OS MVS System Commands* for additional information on these operator commands.

Since the SMB File/Print Server runs as an APF-authorized server, you must ensure that any DLLs that are used by the SMB File/Print Server are APF-authorized. This can be accomplished by using the OMVS extattr +a command.

If you are using the Infoprint Server or OCSF, see the “Infoprint Server Customization Considerations” section in the z/OS Program Directory and the “Cryptographic Services OCSF Customization Considerations” section in the z/OS Open Cryptographic Services Facility Application Programming for information on the location of the DLLs and setting the APF-authorized extended attribute. The DFS load library is called hlq.SIOELMOD.

PC clients must be able to find the server on the network in order to use the shares that the SMB server makes available. If you are using Windows XP, you should ensure that your computer name (specified in the _IOE_SMB_COMPUTER_NAME environment variable in the /opt/dfslocal/home/dfskern/envar) file is the same as your TCP/IP hostname.

SMB server in a sysplex

There are some limitations when running the SMB server in a shared file system (sysplex) environment. In this case, it is best to run a single instance of the SMB server on one member of the sysplex. The reason is that any particular z/OS UNIX file system can only be exported by one SMB server at a time. If you run an SMB server on each member (or on more than one member) of the sysplex, then two SMB servers may try to export the same file system, and one of them will fail.

Another limitation is that an SMB server can only export a z/OS UNIX file system if the file system is owned on the same system that is running the SMB server. There is a configuration option that can be specified in the /opt/dfslocal/home/dfskern/envar file called _IOE_MOVE_SHARED_FILESYSTEM. When specified as ON, the SMB server will attempt to move ownership of the file system to the system where the SMB server is running if it has failed to export that file system because it is owned on a different system. This is independent of the AUTOMOVE specification on the file system.

SMB Access to z/OS data sets

MVS data sets can also be made available to PC clients. They are exported as a data set “file system”. The set of data sets included in the data set “file system” is specified by the prefix of the data set names. We call this a Record File System (RFS). There are many restrictions when using RFS due to the fact that MVS data sets are not byte stream nor hierarchical files. They have maximum record lengths, file naming restrictions and many other limitations. RFS access from a PC acts best when used in a read-only fashion.

For RFS, the DFS server user ID (usually DFS) must have RACF ALTER authority to the data sets that are made available to PC users. Alternatively, you can give the DFS server user ID the OPERATIONS attribute. If you specify a single level prefix in the devtab, you must use the OPERATIONS attribute since you cannot create a data set profile that covers a single level prefix.