



IBM Software Group

# **z/OS® V1R9 Communications Server**

## ***TN3270 other enhancements***



@business on demand.

© 2008 IBM Corporation  
Updated January 15, 2008

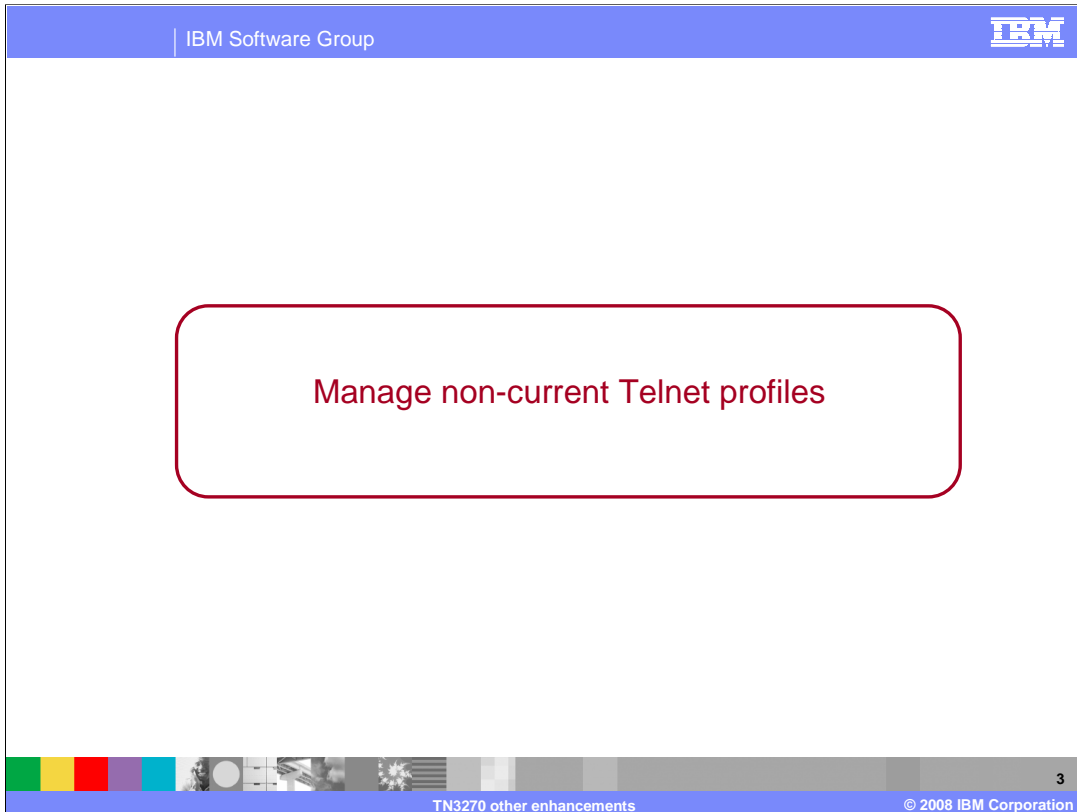
This presentation discusses other TN3270 enhancements in the z/OS V1R9 Communications Server.

## Agenda

- TN3270
  - ▶ Manage non-current Telnet profiles
  - ▶ Telnet uses APPLDATA function
  - ▶ Telnet must run in its own address space



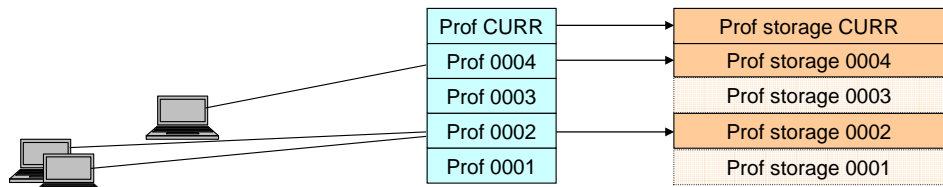
This presentation explains how Telnet realizes storage savings by managing non-current profiles, how Telnet uses the new APPLDATA function, and explain why Telnet must run in its own address space.



Telnet has been enhanced to manage non-Current Telnet profiles.

## Background information - Manage non-current profiles

- V TCPIP, *tnproc*, Obeyfile creates a new profile
  - New storage for all parameters and mapping statements
  - Existing connections remain associated with same profile
- Storage released when:
  - No longer the current profile
  - No connections associated with the profile



4

TN3270 other enhancements

© 2008 IBM Corporation

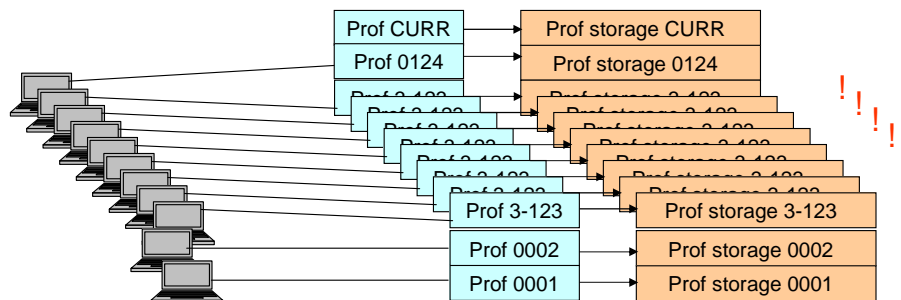
When a V TCPIP, *tnproc*, Obeyfile command is issued, new private storage is obtained from the address space to create completely new profile structures from the Telnet statements in the profile dataset. The new profile becomes the current profile and the previously current profile is considered non-current. Each profile, by port, is assigned a number in ascending order when created starting with 1. The current profile has been assigned a number but is referred to as the CURR profile until it is replaced.

New connections are always associated with the CURR profile. Once a connection is associated with a profile, it stays associated with that profile until the connection is dropped. The connection will never access another profile.

Telnet profile storage is never completely released. A small block is used to anchor the larger parameter and mapping structures. When a profile is no longer current and there are no connections associated with the profile, all parameter and mapping structure storage is released leaving only the small anchor block that is used for profile displays.

## Problem statement - May lead to storage shortage

- Some customers change their profiles frequently
  - ▶ Add new LU names
  - ▶ Add new mapping statements
- End users do not disconnect from logon panel
  - ▶ USSMSG10
  - ▶ Solicitor Panel
  - ▶ No Active SNA sessions associated with the connections



5

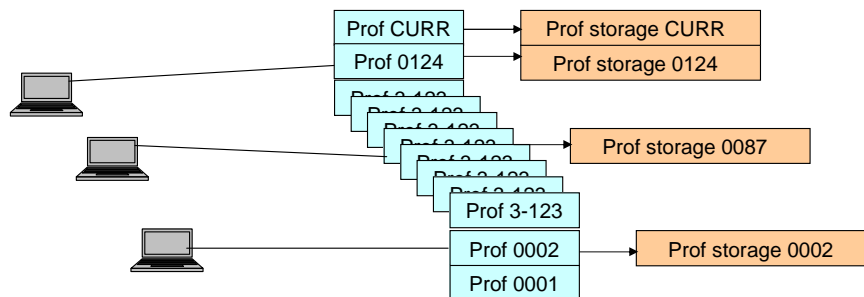
TN3270 other enhancements

© 2008 IBM Corporation

Some customers update their profiles frequently to add new Logical Unit (LU) names or new mapping statements. Each time a new entry is added the V TCPIP, *tnproc*, *Obeyfile* command is issued to activate the new profile. The pre-existing profiles often have connections still associated with them. In many cases these connections do not have active SNA sessions but are sitting at a USSMSG10 or Telnet solicitor prompt waiting for the end user to start a SNA session. The profile structures can not be released because the connection is controlled by that profile until the TCP connection is ended. Frequent profile updates can cause significant storage usage supporting non-current profiles that may have little activity. This storage usage generally shows up as an Auxiliary Storage Shortage message.

## Solution - Manage non-current profiles

- Check connections using non-current profiles
  - ▶ If no SNA session &
  - ▶ No SNA session for at least 'X' amount of time – Drop
- More profiles with no connections – Storage freed
- With auto-reconnect end users re-establish a TCP connection immediately
  - ▶ Connection uses the current profile



6

TN3270 other enhancements

© 2008 IBM Corporation

The solution to this storage problem is to more actively manage the non-Current profiles. Instead of waiting for all connections to end, periodically check all connections associated with the non-current profile. If the connection does not have a Systems Network Architecture, SNA, session and the connection has not been in a SNA session for at least a configurable period of time, drop the connection. Most end users have auto-reconnect specified on their emulators causing the emulator to immediately re-establish a TCP connection with the new current profile. Without auto-reconnect, the end user will need to manually reconnect.

When the connection is dropped, that is one less connection associated with the old profile. When the connection count goes to zero, the profile parameter and mapping structures can be released, freeing potentially large amounts of storage.

A potential storage shortage problem is avoided by cleaning up these unnecessary connections and non-current profiles.

## Telnet configuration

- New Telnet parameter statement
  - ▶ ProfileInactive sec
    - ✓ sec Time in seconds a connection can stay active without being in a SNA session and is associated with an inactive profile.
    - ✓ Telnet is initialized with a value of 1800
    - ✓ 0 will turn off the function (existing behavior)
  - ▶ Set in TelnetGlobals/TelnetParms/Parmsgroup
- If the default is used, connections using non-current profiles will be dropped after being without a SNA session for at least 30 minutes
- To keep your system as-is in z/OS V1R9 you must code 0 to disable the function
  - ▶ ProfileInactive 0
    - ✓ TelnetGlobals block  
– or
    - ✓ All TelnetParms blocks

ProfileInactive is a new parameter used to control how long a connection can stay connected without a SNA session when associated with a non-current profile. The time specified is in seconds. Telnet is initialized with a value of 1800 seconds (or 30 minutes). The function can be turned off by coding a time value of zero. Like most other parameters, ProfileInactive can be specified in TelnetGlobals, TelnetParms, or ParmsGroup depending on the level of granularity required.

If the default is used, connections associated with non-current profiles will be dropped after being without a SNA session for at least 30 minutes. Because the timer is shared with Inactive,PrtnInactive, and KeepInactive, the connection will be dropped sometime soon after 30 minutes, but probably not at precisely 30 minutes.

Remember that the default in the z/OS V1R9 Communications Server is for this function to be active with a time of 1800 seconds. You can turn off the new function by coding a value of zero in either the TelnetGlobals block or in every TelnetParms block.

## Display command example

- Display TCPIP,tnproc,Telnet,PROF,DETail,PROF=0001 to see ProfileInactive time

```

EZZ6080I TELNET PROFILE DISPLAY
  PERSIS  FUNCTION      DIA SECURITY  TIMERS  MISC
(LMTGCAK) (OATSKTQSWHRT) (DRF) (PKLECXN2) (IPKPSTS) (SMLT)
-----
***** **TSBTQ**RT  EC*  BB**D**** *P**STS *DD* *DEFAULT
-M--C-- -----DC-  -B--*----- P--T*  --- *TGLOBAL
L-TTC-- --*-----H-- -B--*----- P--T*  --- *TEARMS
LMTTC** **SBTQ**HRT  DC*  BB***** *P**ST* *DDT(0001)
PERSISTENCE
LUSESSIONPEND
. . .
TIMERS
INACTIVE 0 (OFF)
PROFILEINACTIVE 2
KEEPINACTIVE 0 (OFF)
PRTinACTIVE 0 (OFF)
SCANINTERVAL 40
TIMEMARK 40
SSLTIMEOUT **N/A**
MISCELLANEOUS

```

The Telnet profile detail display shows the value of the variable being used by Telnet. A “P” is present if a value other than zero is specified. An asterisk (\*) is present if zero is specified, turning off the function.



## Diagnosis


- A new connection drop reason for EZZ6034I
  - ▶ INACT-PF

```
EZZ6034I TELNET CONN 00000183 LU **N/A** CONN DROP INACT-PF  
IP..PORT: ::FFFF:9.65.224.165..1643
```


- Use the display profile detail command to verify time
- Use the display connection detail command to verify a ParmsGroup value has not overridden a port value

A new Telnet closing reason, INACT-PF, is created to indicate a connection is being dropped because of an idle connection associated with a non-current profile. This close reason is similar to the other inactivity close reasons.

If connections are not dropping when you think they should, issue the Telnet display profile detail command to verify the time specified at the port level. If that time is correct, issue a Telnet connection detail display command to see if a ParmsGroup is mapped to the connection and if that ParmsGroup has a different ProfileInactive value specified.

IBM Software Group 

**Telnet uses APPLDATA**

 10

TN3270 other enhancements © 2008 IBM Corporation

Telnet is using the new APPLDATA function to pass Telnet security information to the TCP connection.

## Background information - Telnet APPLDATA

- Several customers and network management applications are interested in Telnet connection security options
  - ▶ Conntype ANY port allows both secure and non-secure
    - ✓ Used as a migration path to an all-secure network
  - ▶ Verify correct protocols/ciphers are being used
- Network management tools are used to monitor Telnet connections
  - ▶ TCP connection termination SMF records
  - ▶ Network Management Interface (NMI)
  - ▶ Various netstat commands
- Telnet passes data to the TCP connection:
  - ▶ LU name
  - ▶ Application name
  - ▶ Logmode
  - ▶ User ID (User ID for RestrictAppl – not application.)
  - ▶ Status
    - ✓ If definite response is requested
    - ✓ If the connection is being monitored
    - ✓ Telnet connection mode (TN3270E/TN3270/Linemode)

As security becomes more prevalent, many administrators are using the Conntype ANY option in Telnet to slowly migrate their users to secure connections. With Conntype ANY, a single port can be used for both secure and non-secure connections depending on the client connection mode choice. These administrators are looking for a way to verify connections are secure and identify the non-secure connections so they can be made secure.

Other administrators are working to migrate their end users from older protocols such as SSLv2 to newer protocols such as TLSv1. These administrators are looking for a way to verify the protocol or ciphers used and identify the end users who need to upgrade.

This type of connection verification is typically done using network management tools such as TCP connection termination System Management Facilities, SMF, records. Real time queries can be done with the NMI interface or various netstat commands.

Since OS/390 V2R5, Telnet has saved some information in the TCP connection control block. This information includes LU name, SNA application name, the SNA logmode used, and the user ID used by the end user to gain access past the Telnet RestrictAppl function. The user ID may be the one eventually used to log onto the SNA application but does not need to be. RestrictAppl is not used very often leaving the user ID field blank in the TCP connection Telnet section. Telnet also reports if the connection supports definite response, if the connection is being monitored for performance data, and what connection mode is used.

## Problem statement - No security information

- Security information is not passed to TCP.
  - ▶ Do not know which connections are secure
  - ▶ Do not know which connections adhere to cipher requirements.

Security information is not saved in the TCP connection control block and therefore can not be reported by any of the network management tools. Without this information being reported, administrators have no way of knowing what is going on with the security migration of their Telnet clients.

## Solution - Telnet exploits APPLDATA

- Implement the new APPLDATA Function
  - ▶ Telnet data retrieval common with other applications
  - ▶ Use existing structures and filters for APPLDATA
  - ▶ Telnet-specific and APPLDATA overlap information
- APPLDATA better than adding to existing structure
  - ▶ Common filtering tools provide easy access to data
  - ▶ See "Enable application identifier in NMI, SMF, and Netstat"
- Telnet APPLDATA available



### Offset

1-8	Telnet application identifier ( <b>EZBTNSRV</b> )
10-17	LU name
19-26	Application name
28	Connection mode (TN3270E/TN3270/Linemode/DBCS)
29	Emulator type (Terminal/Printer)
31	Security level ( <b>Basic/Secure/TTLS</b> )
32-33	Protocol (if secure) (TLSv1/SSLv3/SSLv2)
34-35	Cipher (if secure)

- 40 character limit. Not enough room for:
  - ▶ Logmode - Eight characters
    - ✓ Less interesting than LU or application name.
  - ▶ User ID - Eight characters & seldom used
    - ✓ This is the user ID for RestrictAppl, not application user ID.
  - ▶ Definite Response status
  - ▶ Monitoring status

13

TN3270 other enhancements

© 2008 IBM Corporation

The new APPLDATA area provides a strategic location for Telnet application data passed to the TCP connection. There are several advantages to using the APPLDATA area. APPLDATA is a common area for all applications allowing Telnet queries to use standard filters instead of requiring unique tools. An NMI filter already exists for APPLDATA and the SMF record has a new section for APPLDATA. However, this will result in the duplication of some Telnet data that already exists in the TCP connection control block. With this solution the customer will be able to immediately generate summary and detail reports documenting the security level of Telnet connections.

You looked at adding to the existing structure that currently holds the Telnet-specific data described on an earlier slide. The structure in the TCP control block already exists with room to add the security information and the mechanism already exists for Telnet to pass the data to TCPIP. However, if new data is added to a Telnet-specific section of the TCP control block, these fields have to be propagated to the NMI, SMF, and netstat structures with code added to support the new fields. New filters would be required for NMI. Unique tools and methods would be required to access or filter Telnet data.

The advantages of using APPLDATA outweigh using the existing Telnet structure and mechanism. The APPLDATA section was created for general application use and is the strategic location for application data. The existing NMI APPLDATA filter allows the data to be immediately accessible to network managers without waiting for new Telnet-unique filters. For general APPLDATA implementation information, see the "Enable application identifier in NMI, SMF, and Netstat" presentation.

Telnet will use the identifier "EZBTNSRV" and supply the listed information at the specified offsets in the APPLDATA section of the NMI, SMF, and netstat records. The security level, Protocol, and Cipher will satisfy the security information requirement.

The preferred data extraction method for network management users is to use APPLDATA and to phase out the use of the TCP connection Telnet-unique section. However, APPLDATA has a 40 character limit, converts flag bits to readable EBCDIC characters, and new security information is being added. For these reasons, not all data currently saved in the Telnet-unique section can be copied into the new APPLDATA area. As a connection filter, logmode and user ID do not seem to be as interesting as LU name or application name. Also, because the user ID is associated with the RestrictAppl statement, it has limited value.

## Telnet APPLDATA collection points

### ■ Capture data at three major events:

1. Connection negotiation complete
  - Security Level / (Protocol) / (Cipher)
  - Connection mode & Emulator Type
  - (LU name – If TN3270E connection)
2. SNA session established
  - LU name
  - Application name
3. SNA session ends and TCP connection remains  
(If connection drops, data clear is not done)
  - Clear application name
  - (Clear LU name – If not TN3270E connection)

The APPLDATA section is updated at three different key events for a Telnet connection.

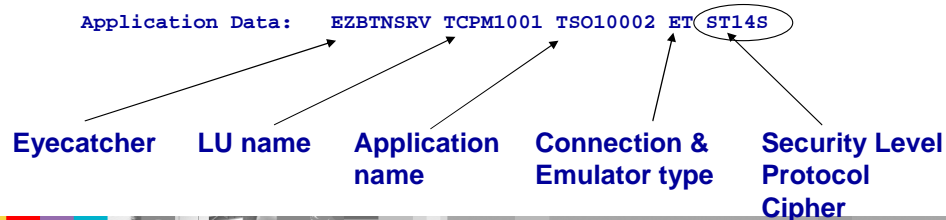
The first update to APPLDATA occurs when Telnet protocol negotiation is complete. By this time the System SSL handshake and Telnet negotiations are complete. The security level, protocol, and cipher are known as are the connection mode and emulator type. If the connection mode is TN3270E, the LU name assigned to the connection is also known. This information will not change over the life of the connection.

The second update to APPLDATA occurs when a SNA session is established. If an LU name was not associated with the connection during Telnet protocol negotiation, an LU name is assigned during SNA session setup. The LU name and application name are added to the APPLDATA section.

The third update to APPLDATA occurs if the LUSESSIONPEND statement is mapped to the Telnet connection which keeps the TCP connection active after logoff from the application. In this case the application name is cleared and the LU name may be cleared depending on connection mode. If the connection is dropped when the session logoff is received, the application name and LU name remain in the APPLDATA section and will be present in the TCP connection termination SMF record.

## Retrieving Telnet APPLDATA

- Telnet APPLDATA - Netstat
  - ▶ Several Netstat commands will show APPLDATA
    - ✓ See "Enable application identifier in NMI, SMF, and Netstat" for examples.
- Telnet APPLDATA – SMF
  - ▶ 3rd section of TCP termination record is the existing Telnet data
  - ▶ 4th section of TCP termination record is APPLDATA
- Telnet APPLDATA – NMI
  - ▶ TCP connection records (GetConnectionDetail)
    - ✓ No filter for Telnet Information
    - ✓ APPLDATA filter is available
  - ▶ New filters not needed to filter on Telnet data!



15

TN3270 other enhancements


© 2008 IBM Corporation

APPLDATA is presented in several netstat commands. See the APPLDATA presentation for details. A previous slide states the possible values for Connection mode, Emulator type, and security values.


The TCP connection termination SMF record reports the Telnet-unique data section and the new APPLDATA section. The 3<sup>rd</sup> section is the Telnet-unique data and the 4<sup>th</sup> section is the APPLDATA section. The 4<sup>th</sup> section includes the same information as the 3<sup>rd</sup> section with the exception of logmode, user ID, Definite Response status and monitoring status. If you do not need the logmode, user ID, Definite Response status, or monitoring status, the 3<sup>rd</sup> section can be ignored. The circled security data is only available in the new 4<sup>th</sup> section.

TCP connection records are retrieved by the Network Management Interface (NMI) using the GetConnectionDetail request. Filters for Telnet-specific information such LU name and application name used by the GetTnConnectionData request are not available for the TCP connection GetConnectionDetail request. However, TCP connection data can be filtered based on APPLDATA which contains the Telnet-specific data typically requested.

The Connection mode and emulator type values ET, in the Application Data, indicate a TN3270E connection mode with a terminal emulator. Security values ST14S indicate a secure connection using TLSv1 and cipher 4S. The connection is represented by LU name TCPM1001 and the end user is in session with application TSO10002.

IBM Software Group 

**Telnet must run in its own address space**

 16  
TN3270 other enhancements © 2008 IBM Corporation

The TN3270E Telnet server is no longer available in the TCPIP address space. Telnet must run in its own address space.



## Background information - TSASO

- The Telnet in a Separate Address Space Option (TSASO) has been available since z/OS V1R6.
- Telnet can run in the TCPIP address space
  - Or
- Telnet can run in its own address space
- Several advantages to TSASO
- Dual support allowed for careful migration

17

TN3270 other enhancements

© 2008 IBM Corporation

Telnet has been able to run in its own address space since z/OS V1R Communications Server which was generally available in September, 2004. Since that time, customers have had the option to continue configuring Telnet and TCPIP to run in a single shared address space or configure Telnet to run in its own address space.

There are several advantages to running them separately. Telnet priority can be set to a different priority than that of TCPIP. Telnet can be stopped and restarted without stopping TCPIP. When the TCPIP stack is stopped, Telnet remains active. Separating Telnet and TCPIP makes problem diagnosis easier. You can start as many as eight instances of Telnet. In a common INET environment, Telnet can be associated with multiple stacks, or have affinity to a single stack by using the TCPIPJOBNAME statement in TELNETGLOBALS.

Dual support was implemented to allow careful, deliberate migration of Telnet from the TCPIP address space into its own address space with the strategic direction that all customers will move Telnet to realize the TSASO advantages.

## Problem statement - Dual support

- Dual support creates:
  - ▶ Confusion
    - ✓ Where to run Telnet
    - ✓ Remembering where Telnet is running
    - ✓ Where to add maintenance
    - ✓ What documentation needed for problems
    - ✓ Function added to TSASO only (CheckClientConn)
    - ✓ Display commands
  
  - ▶ Duplicate development, test, and support effort

Over time, the dual support has begun to cause confusion and duplicate effort. In some cases, new functions are implemented in only the TSASO version of Telnet.

## Solution - Telnet not supported in TCPIP

- Stop supporting Telnet in the TCPIP address space
  - ▶ Safeguard the TCPIP stack
  - ▶ Reduce confusion
- Telnet must run in its own address space
  - ▶ RACF® authorization
  - ▶ UNIX® System Services superuser
  - ▶ JCL
  - ▶ Telnet profile
- Remember, before z/OS V1R8 TCPIP configuration accepted profile statements not accepted by TSASO
  - ▶ InternalClientParms/EndInternalClientParms
  - ▶ TelnetDevice in BeginVTAM

The time has come to stop supporting Telnet running in the TCPIP address space. Three releases should give everyone ample opportunity to migrate their Telnet server into its own address space. Most customers who use Telnet heavily have already switched to TSASO to isolate Telnet from TCPIP. Separating Telnet and TCPIP will isolate the TCPIP stack from Telnet problems allowing the stack to continue supporting other applications when Telnet experiences a problem. The confusion created by allowing Telnet to run in either its own or the TCPIP stack address space is eliminated.

If you haven't migrated yet, the major steps include: (1) setting up RACF authorization for the procedure name and associated User ID, (2) giving the user ID superuser status, (3) creating the Job Control Language, JCL, to start Telnet in its own address space, and (4) isolating the Telnet profile statements into their own dataset. For more setup details, see *z/OS Communications Server IP Configuration Guide, Accessing Remote hosts using Telnet - Telnet in its own address space*.

If you are migrating from a release other than z/OS V1R8 be aware that there are some statements and some statement locations that became invalid in z/OS V1R8. For example, before z/OS V1R8 the InternalClientParms block statement was valid and the TelnetDevice statement was accepted in the BeginVTAM block. Telnetdevice must now be in TelnetGlobals, TelnetParms, or ParmsGroup. There are other changes. The best advice is to carefully review your console messages for Telnet profile errors and fix any errors or warnings issued.

## TCP/IP changes

- Telnet profile statements ignored by TCPIP
  - ▶ EZZ0209I TELNET SERVER CONFIGURATION STATEMENTS IGNORED IN TCPIP
- Telnet commands ignored by TCPIP
  - ▶ EZZ0210I TELNET SERVER COMMAND IGNORED BY TCPIP
- TCPIP commands no longer support Telnet
  - ▶ Help commands
  - ▶ Module & storage display
    - ✓ Module display already implemented in Telnet

20

TN3270 other enhancements

© 2008 IBM Corporation

Telnet profile statements and Telnet commands will no longer be recognized by TCPIP. The TCPIP configuration processor will not flag every individual Telnet statement found in a TCPIP profile but will issue a single message indicating Telnet profile statements were ignored. Any Telnet command submitted to the TCPIP address space will be ignored.

TCPIP commands such as Help and Stor no longer provide Telnet information. Telnet has created its own Help commands and enhanced the Stor command to replace the TCPIP versions.

## TSASO commands

- New Telnet help commands
  - ▶ D TCPIP, *tnproc*, Telnet, HEIp
    - ✓ D TCPIP, *tnproc*, Telnet, HEIp, STOR
    - ✓ D TCPIP, *tnproc*, Telnet, HEIp, Telnet
      - <CLientID | CONNection | INACTLUS | OBJect | PROFile | WLM>
  - ▶ V TCPIP, *tnproc*, Telnet, HEIp
    - ✓ V TCPIP, *tnproc*, Telnet, HEIp, Obeyfile
    - ✓ V TCPIP, *tnproc*, Telnet, HEIp, Telnet
      - <ABENDTRAP | ACT | DEBug | INACT | QUIesce | RESUME | STOP>
- Telnet STOR command is enhanced
  - ▶ Existing display
    - ✓ D TCPIP, *tnproc*, Telnet, Stor, module=*modname*
  - ▶ Enhanced display to match the TCPIP display
    - ✓ D TCPIP, *tnproc*, Telnet, STOR
      - The display includes the current and maximum storage

21

TN3270 other enhancements

© 2008 IBM Corporation

Telnet help commands are used to see the format of any Telnet Display or Vary command. Each help command will show the options available at the next level of detail. For example, if you issue `d tcpip, tnproc, t, help` you will see that you can issue help for either STOR or TELNET. If you then issue `d tcpip, tnproc, t, help, telnet` you will see all the display Telnet options available. The next slide shows an example of the connection help display.

The TCPIP STOR command has two functions. It can display the load module name that contains the specified module name along with the module address and the first 48 bytes of storage. If no module name is specified, the command displays the current and maximum storage usage for the TCPIP address space and any storage limits. When Telnet ran in the TCPIP address space, both of these functions included Telnet.

When TSASO was implemented the module display function of the STOR command was added to help identify the service level of Telnet modules. The current and maximum storage display was not added to Telnet. It has been added in the z/OS V1R9 Communications Server.

## Display command example - TSASO help conn

- **Display**  
**TCPIP,tnproc,<Telnet>,HElp,Telnet,CONNection**
  - ▶ ( Choose no more than one within the parentheses )
  - ▶ < Optional keyword >

```

d tcpip,tnproc,help,conn

EZZ6107I D TCPIP,TNPROC,TELNET,CONNECTION
1  (9), (CONN=XCONNID | IPPORT=XIPADDR..XPORT | LUNAME=XLUNM)
   <, (DETAIL | SUMMARY) >>
   |
   <, (LUNAME=XLUNM* | APPL=(XAPPLNM | XAPPLNM*) |
2   TCPIPJOBNAME=XTCPINM | PROTOCOL=XPROTMODE |
   LUGROUP=XLUGRPNM | IPGROUP=XIPGRPNM |
   IPADDR=(XIPADDR|XV4MASK:XV4SUBNET|XIPADDR/XPREFIXLEN))
   <, (NOHNAME | HNAME) >>
   |
3  <, (HNAME=X*HOSTNAME | HNGROUP=XHNGROUPNM)
   <, (NOHNAME | HNAME) > (1)
   <, PORT=( ALL | XNUM | XNUM1..XNUM2 | XNUM, XQUAL) >
   <, PROF=( CURRENT | XPROFID | ACTIVE | ALL | BASIC | SECURE) >
   <, MAX=( XNN | * ) >

```

This slide shows the result of issuing the Telnet help command for the connection display. This command has more options than any other Telnet command. Note that parentheses indicate no more than one choice can be made and the less than/greater fences indicate optional input. For example, the display allows either section 1,2, or 3 but not a combination. Within section 1 you can specify either Conn= or IPPORT= or LUNAME= and optionally specify detail or summary. The circled left parenthesis indicates the end of the “choose one” section. PORT=, PROF=, and MAX= are all available as optional keywords.

## Display command example - TSASO STOR command

- Display TCPIP,*proc*,<Telnet,>STOR to see storage usage

TCPIP	<pre> EZZ8453I TCPIP STORAGE EZZ8454I TCPCS   STORAGE   CURRENT   MAXIMUM   LIMIT EZZ8455I TCPCS   ECSA      8722K    9419K    NOLIMIT EZZ8455I TCPCS   POOL      5952K    5967K    NOLIMIT EZZ8459I DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY </pre>
Telnet	<pre> EZZ8453I TELNET STORAGE EZZ8454I TELNET6 STORAGE   CURRENT   MAXIMUM   LIMIT EZZ8455I TELNET6 ECSA      85K      137K     NOLIMIT EZZ8455I TELNET6 POOL      6810K    7241K    NOLIMIT EZZ8455I TELNET6 CTRACE    262372K  262372K  262372K EZZ8459I DISPLAY TELNET STOR COMPLETED SUCCESSFULLY </pre>

- CTrace is part of Telnet private pool storage

This slide shows the TCPIP and Telnet display storage output. They are very similar except for an additional line in the Telnet display for CTRACE. TCPIP CTRACE storage resides in a separate dataspace and is not part of the storage display. Telnet CTRACE resides in Telnet's private address space. If only the total POOL value were shown, the CTRACE amount would obscure the amount of storage used by Telnet processes. Therefore, before the data is presented, the CTRACE amount is subtracted from the total POOL amount and presented on its own line.

The CTRACE amount appears large because Telnet always allocates a 256M block of storage to support the largest CTRACE BUFSIZE amount. This storage is not backed until it is filled in with data. You will never use real storage resources for more than the amount you define on the CTRACE BUFSIZE parameter.

## TSASO functions

- Multilevel security now supported by TSASO
  - ▶ Previously, Multilevel security for Telnet required Telnet to run in the TCPIP address space
- INTCLIEN no longer supported
  - ▶ Use the Telnet job name on the TCPIP PORT statement
- NACUSERID not required for Telnet
  - ▶ The Telnet procedure can be associated with its own user ID

Multilevel security for Telnet required that Telnet run in the TCPIP address space because Telnet did not incorporate a needed interface to the security server. That interface has been added in this release.

When Telnet ran in the TCPIP address space, INTCLIEN was used instead of a jobname to reserve a port for Telnet. NACUSERID was required for Network Access Control of Telnet connections. With TSASO as the only option now, INTCLIEN should not be used and NACUSERID is optional, not required. These changes are described in the z/OS V1R9 Communications Server: Configuration Guide under "Accessing Remote Hosts using Telnet".



## Things to think about TSASO

- TCP/IP affinity is required to obtain stack job name and host name for these functions:
  - ▶ Telnet SNMP Subagent activation - Must direct registration by stack jobname.
  - ▶ WLM Registration - Must specify stack host name during registration.
  - ▶ SMF Hostname - Stack host name used.
- Command processing - The procedure name must be specified to route commands to Telnet. Otherwise the command is routed to the default TCPIP stack.
  - ▶ D TCPIP,TELNET1,T,PROFILE
- In a CINET environment, Telnet connections can be supported by different stacks.
  - ▶ Netstat Telnet displays show only connections on the stack where the command was issued.
- SNMP Subagent Limitation - Agent/subagent connection requires a one-to-one match up.
  - ▶ Any particular agent can support only one Telnet subagent
- SMF address space name - Will be the name of the Telnet procedure, not the stack.
  - ▶ SMF 118 Started Task name (SMFTNTST)
  - ▶ SMF 119 Address Space Name of the Writer (SMF119TI\_ASName)
- Can not change IPv4/IPv6 or INET/CINET Environments while running - Unpredictable results.
  - ▶ New port activations will fail if environment change is detected.
  - ▶ Recommend stop Telnet, change environment, restart Telnet.
- INTCLIEN Port Reservation - Valid only for Telnet as part of the stack.
  - ▶ If Telnet is running as its own procedure and tries to listen on a reserved port, the BIND will fail.
  - ▶ Specify the Telnet jobname instead.

If you are already running Telnet in its own address space, you do not need to read this list. However, if you are still running Telnet in the TCPIP address space, there may be some useful tips here for you. Also, be sure to review the z/OS V1R9 Communications Server: Configuration Guide, “Accessing Remote Hosts using Telnet”.

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_TN3270\\_Other.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_TN3270_Other.ppt)

This module is also available in PDF format at: [../TN3270\\_Other.pdf](http://../TN3270_Other.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

## Trademarks, copyrights, and disclaimers

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

IBM            RACF            z/OS

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. 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.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.