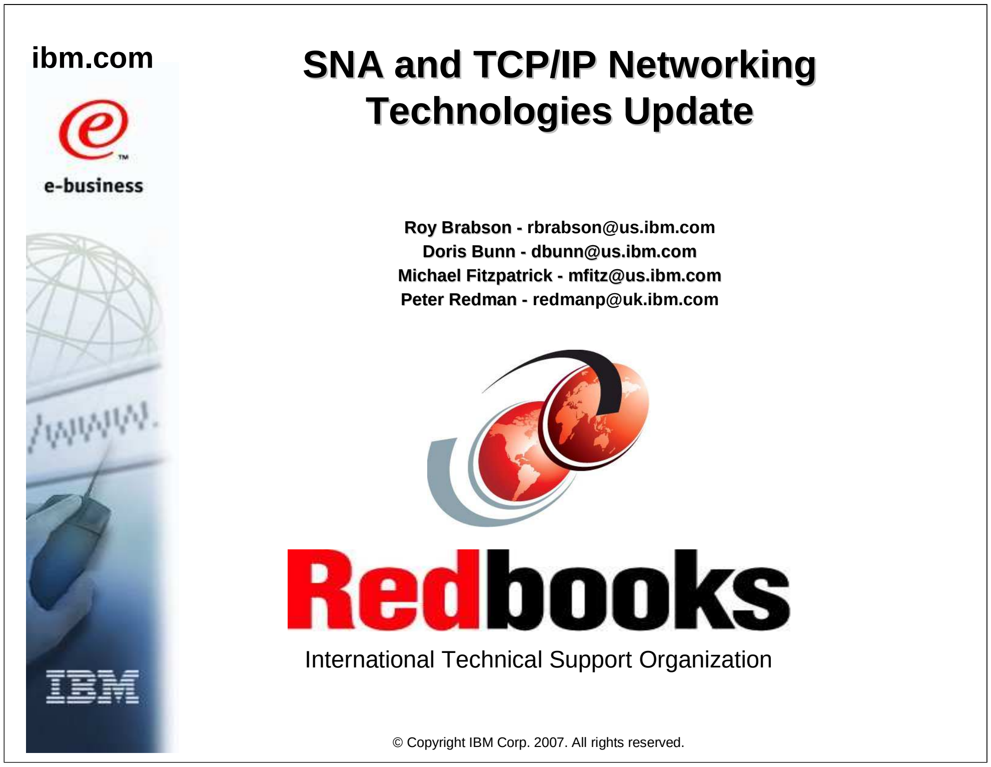
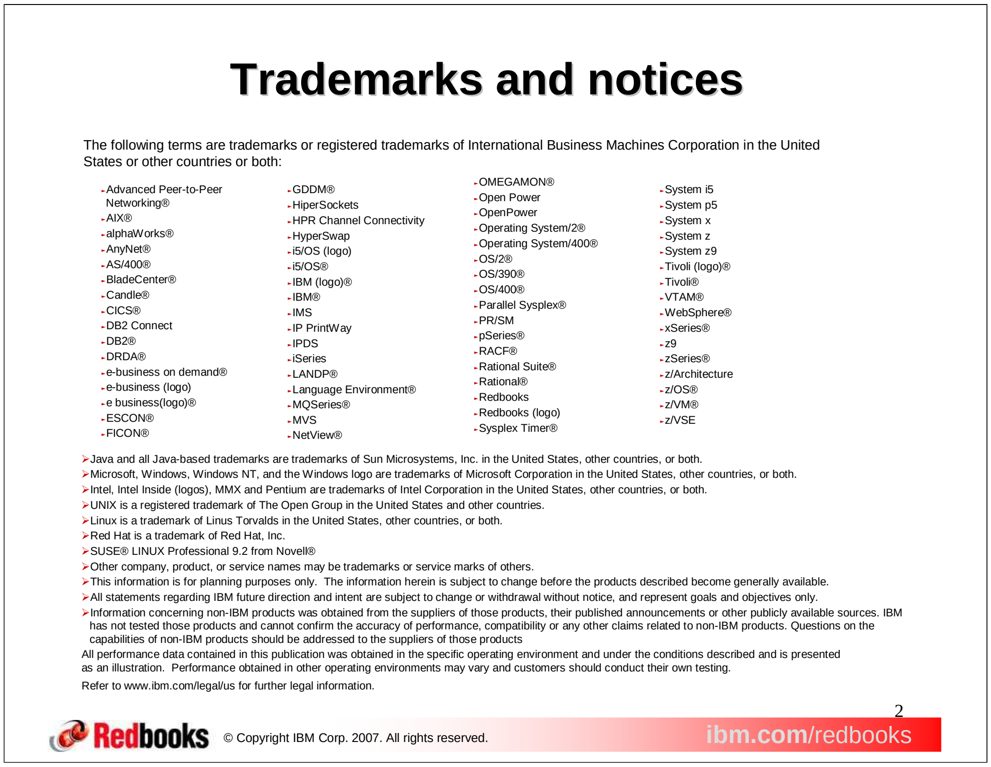


ibm.com

SNA and TCP/IP Networking Technologies Update

Roy Brabson - rbrabson@us.ibm.com
Doris Bunn - dbunn@us.ibm.com
Michael Fitzpatrick - mfitz@us.ibm.com
Peter Redman - redmanp@uk.ibm.com

International Technical Support Organization



Trademarks and notices

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

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- Java and all Java-based trademarks are 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.
- Intel, Intel Inside (logos), MMX and Pentium 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.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- 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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

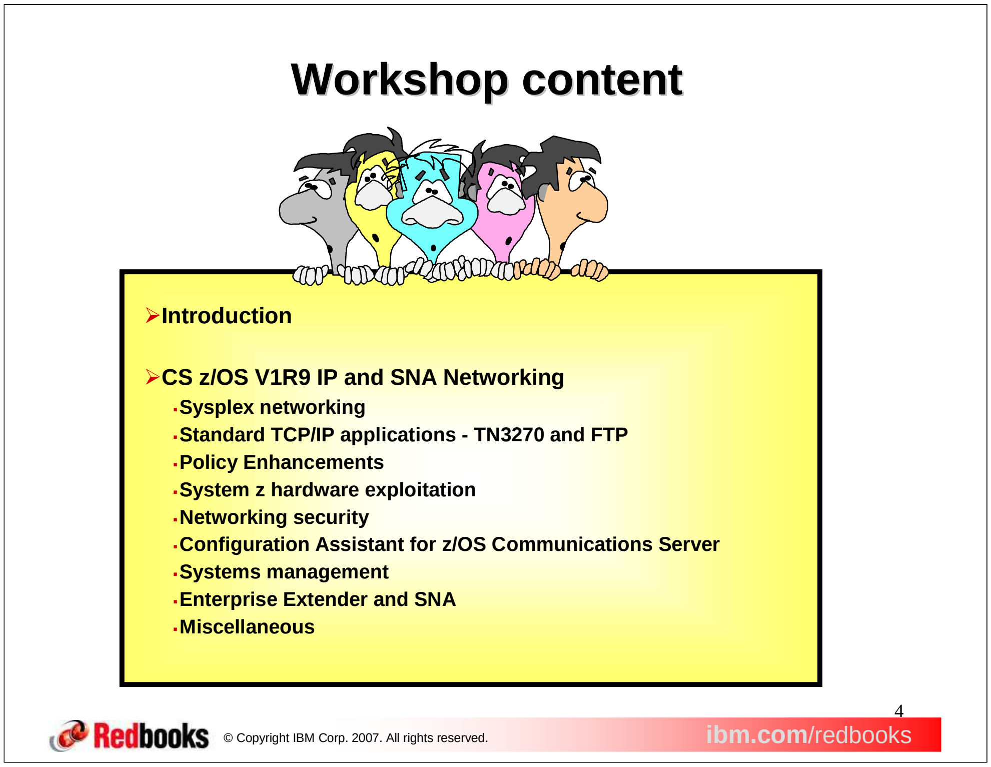
Refer to www.ibm.com/legal/us for further legal information.

Workshop objectives

➢ **The overall objectives of the networking workshop are:**

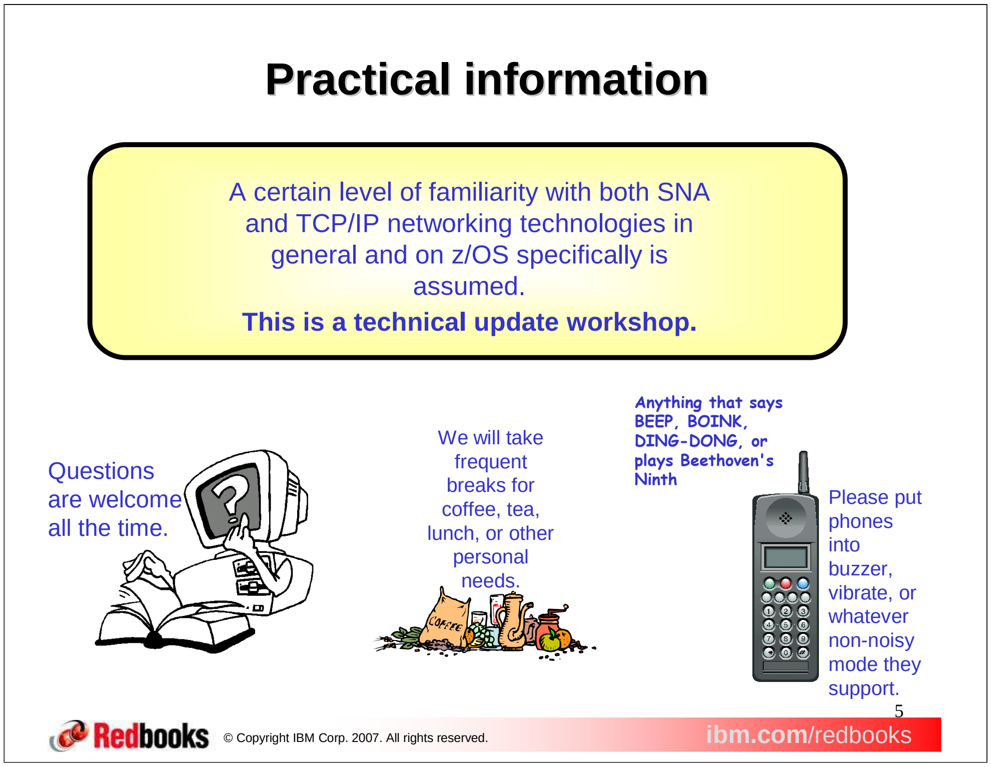
- Make attendees aware, at a conceptual level, of selected new functions and capabilities in the Communications Server for z/OS V1R9.
  - ✓ Focus is on explaining concepts and where the new functions may be useful
  - ✓ Configuration principles will be covered at a conceptual level, but not in detail
  - ✓ For detailed configuration information, the attendees are referred to the product documentation

Redbooks



Workshop content

- **Introduction**
- **CS z/OS V1R9 IP and SNA Networking**
  - **Sysplex networking**
  - **Standard TCP/IP applications - TN3270 and FTP**
  - **Policy Enhancements**
  - **System z hardware exploitation**
  - **Networking security**
  - **Configuration Assistant for z/OS Communications Server**
  - **Systems management**
  - **Enterprise Extender and SNA**
  - **Miscellaneous**



Practical information

A certain level of familiarity with both SNA and TCP/IP networking technologies in general and on z/OS specifically is assumed.

This is a technical update workshop.

Questions are welcome all the time.

We will take frequent breaks for coffee, tea, lunch, or other personal needs.

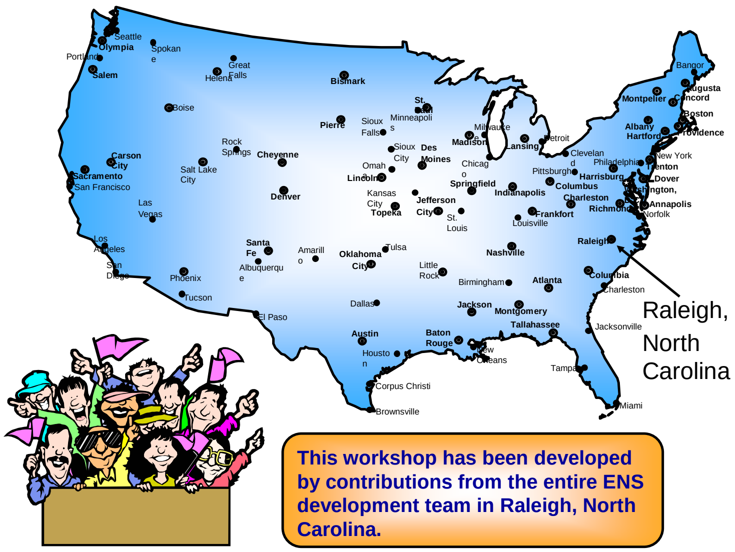
Anything that says BEEP, BOINK, DING-DONG, or plays Beethoven's Ninth

Please put phones into buzzer, vibrate, or whatever non-noisy mode they support.

Redbooks

ibm.com/redbooks

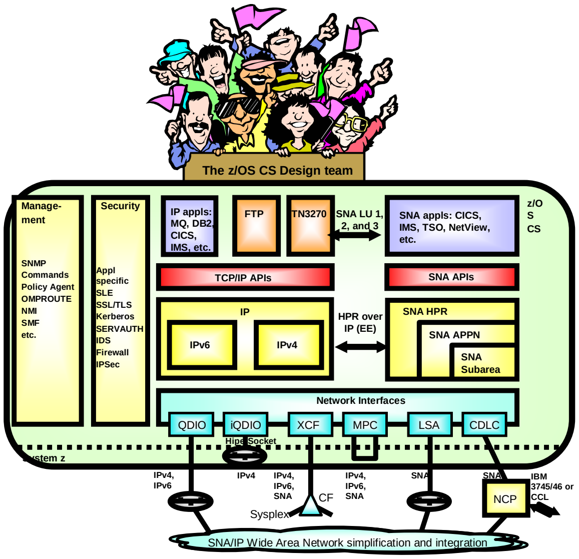
Research Triangle Park, Raleigh, North Carolina - the home of Enterprise Networking Solutions



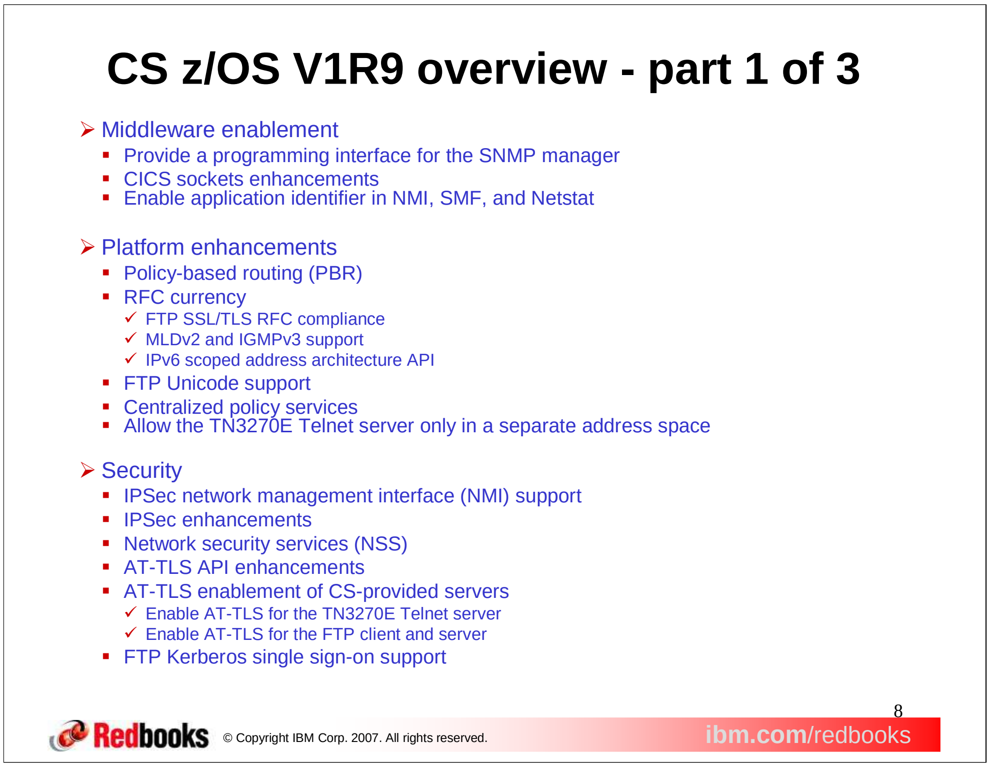
This workshop has been developed by contributions from the entire ENS development team in Raleigh, North Carolina.

6

Communications Server on z/OS - What drives the selection of functions being added to CS z/OS?



- ➢**Satisfy critical customer requirements**
  - ➢Security, availability, reliability, scalability, capability and performance
  - ➢Sysplex, EE, FTP, TN3270E, IDS, Policy, etc.
- ➢**Position z/OS for emerging technology requirements:**
  - ➢Full IPv6 support through staged delivery of the next generation IP network for z/OS
- ➢**Provide networking security infrastructure to meet customers' emerging security compliance requirements**
  - ➢Integrated IP Security
- ➢**Optimize the application and middleware environment on z/OS**
  - ➢Application Transparent TLS, Sysplex Load Balancing Advisor, CICS Socket Optimization, Enterprise Extender, etc.
- ➢**Exploit and add value to System z technology innovations**
  - ➢High speed networking for zSeries
  - ➢Encryption accelarators and technologies
- ➢**Help reducing cost**
  - ➢Network management
  - ➢Autonomics



CS z/OS V1R9 overview - part 1 of 3

- Middleware enablement
  - Provide a programming interface for the SNMP manager
  - CICS sockets enhancements
  - Enable application identifier in NMI, SMF, and Netstat
- Platform enhancements
  - Policy-based routing (PBR)
  - RFC currency
    - ✓ FTP SSL/TLS RFC compliance
    - ✓ MLDv2 and IGMPv3 support
    - ✓ IPv6 scoped address architecture API
  - FTP Unicode support
  - Centralized policy services
  - Allow the TN3270E Telnet server only in a separate address space
- Security
  - IPSec network management interface (NMI) support
  - IPSec enhancements
  - Network security services (NSS)
  - AT-TLS API enhancements
  - AT-TLS enablement of CS-provided servers
    - ✓ Enable AT-TLS for the TN3270E Telnet server
    - ✓ Enable AT-TLS for the FTP client and server
  - FTP Kerberos single sign-on support

8

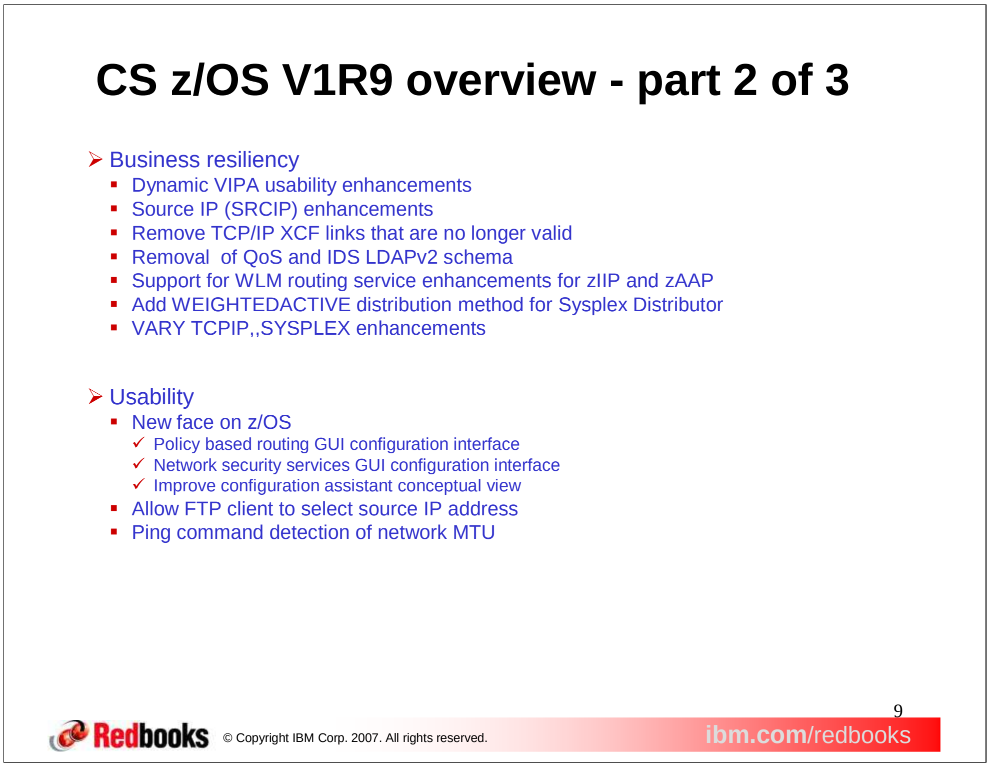
Redbooks

ibm.com/redbooks

Changes were made in the area of Middleware enablement. A new SNMP manager programming interface is now available. Multiple enhancements were made to CICS sockets for improve availability. A new API is available which allows applications to store unique identifying data on TCP sockets. This application data is also provided on the network management interface as well as SMF records.

Changes were made to allow the z/OS platform. Policy based routing allows other criteria, defined via policy, to be used to determine how outbound traffic is routed. Support is included for Multicast protocols MLDv2 and IGMPv3. FTP now fully supports SSL/TLS RFC 4217. Support for scoped addresses is available for link local addresses. Policy for all the disciplines can now be defined and stored in a central location. Finally, the TN3270E telnet server can no longer run in TCP/IP's address space.

Security enhancements were also made in the Communications Server for z/OS V1R9. A new network management interface (NMI) is available to monitor and manage IPSec. Some enhancements were made to the existing IPSec function in the area of Perfect Forward Secrecy and SWSA takeover and giveback. A new function, Network security services, allows certificate services for IPSec to be in a central location. It also allows for remote monitoring and managing of IPSec. The FTP client and server and the TN3270E telnet server have been enabled for AT-TLS. The FTP server now supports Kerberos single sign-on.



CS z/OS V1R9 overview - part 2 of 3

- Business resiliency
  - Dynamic VIPA usability enhancements
  - Source IP (SRCIP) enhancements
  - Remove TCP/IP XCF links that are no longer valid
  - Removal of QoS and IDS LDAPv2 schema
  - Support for WLM routing service enhancements for zIIP and zAAP
  - Add WEIGHTEDACTIVE distribution method for Sysplex Distributor
  - VARY TCPIP,,SYSPLEX enhancements
- Usability
  - New face on z/OS
    - ✓ Policy based routing GUI configuration interface
    - ✓ Network security services GUI configuration interface
    - ✓ Improve configuration assistant conceptual view
  - Allow FTP client to select source IP address
  - Ping command detection of network MTU

9

ibm.com/redbooks

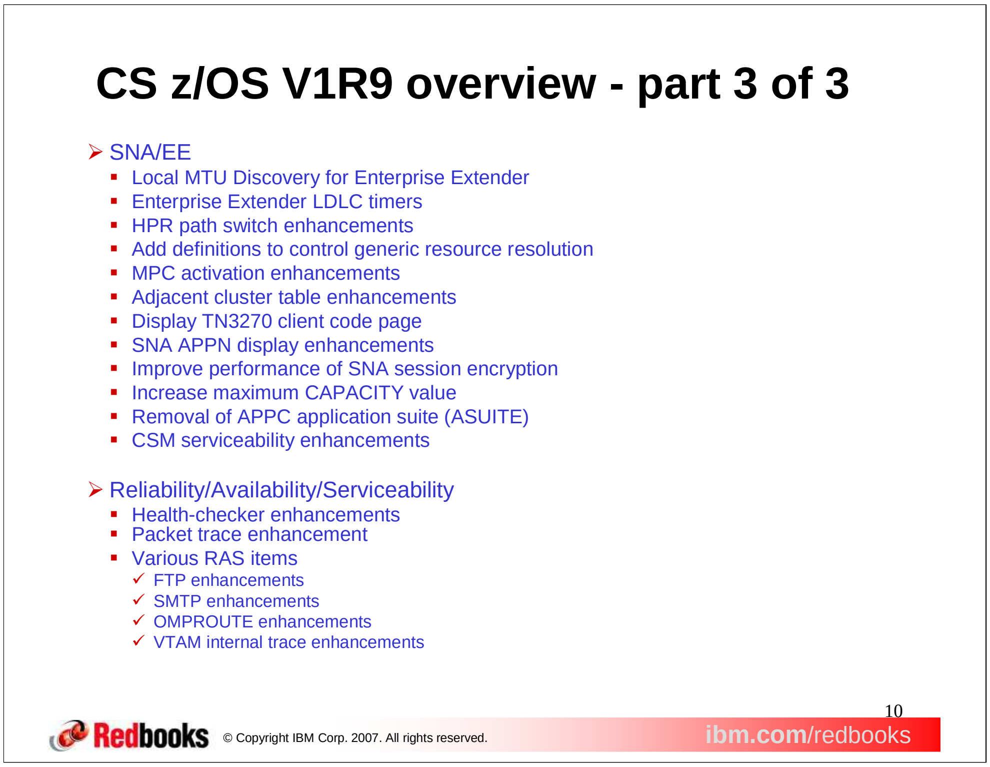
There are several sysplex related enhancements in z/OS V1R9. The starting of AUTOLOG applications, that bind to a dynamic VIPA, can optionally be delayed until the TCP/IP stack has joined the sysplex. A port range can now be specified on the VIPADISTRIBUTE statement. A distributed dynamic VIPA can now be specified as the source IP address on the SRCIP statement. TCP/IP XCF links that are no longer being used are now deleted when the last stack on a LPAR leaves the sysplex. Sysplex Distributor and the Load Balancing Advisor take into account the processing on the specialty processors, zIIP and zAAP, when making load balancing decisions. A new Sysplex Distributor distribution method, WEIGHTEDACTIVE, provides more granular control over workload distribution. A single VARY TCPIP,,SYSPLEX command can now be used to quiesce or resume multiple listeners.

The QoS and IDS LDAPv2 schema is no longer supported.

The IBM Configuration Assistant for z/OS Communications Server supports configuring routing policy for the new Policy based routing function. It also supports configuration for the new function, Network security services. Configuration data is now stack-oriented instead of technology oriented. All the configuration data is kept in the same configuration file.

The FTP client can now optionally provide the source IP address that will be used when connecting to the server.

The Ping command has been enhanced to detect MTU problems in the network.



CS z/OS V1R9 overview - part 3 of 3

- SNA/EE
  - Local MTU Discovery for Enterprise Extender
  - Enterprise Extender LDLC timers
  - HPR path switch enhancements
  - Add definitions to control generic resource resolution
  - MPC activation enhancements
  - Adjacent cluster table enhancements
  - Display TN3270 client code page
  - SNA APPN display enhancements
  - Improve performance of SNA session encryption
  - Increase maximum CAPACITY value
  - Removal of APPC application suite (ASUITE)
  - CSM serviceability enhancements
- Reliability/Availability/Serviceability
  - Health-checker enhancements
  - Packet trace enhancement
  - Various RAS items
    - ✓ FTP enhancements
    - ✓ SMTP enhancements
    - ✓ OMPROUTE enhancements
    - ✓ VTAM internal trace enhancements

Several SNA/EE enhancements were made in z/OS V1R9. Enterprise Extender now learns of MTU changes dynamically. Furthermore, Enterprise Extender logical data link control timers can now be defined for each local EE VIPA. Enhancements were made to improve the performance when HPR path switch occurs in large networks. Generic resource resolution preferences can now be defined using VTAM definitions. Re-activation of MPC groups occurs automatically when the minimum number of subchannels become available for a FICON connected host. Enhancements were made to the adjacent cluster table to allow for more granular control in subnetwork searching. The character set and code page combination used by a TSO session is now provided on the GTTERM macro and the DISPLAY TSOUSER command. New messages are added to displays to aid in the diagnosis of problems related to RTP physical units (PUs) as well as other types of SNA PUs. The performance of SNA session encryption has been improved. The allowed range of CAPACITY values has been increased with an additional range of 1G to 100G (gigabits per second) for high speed connections on all definition statements where CAPACITY can be specified for high speed connections. The APPC application suite (ASUITE) is no longer supported. Messages are now issued to the console when the Communications Storage Manager (CSM) adjusts the maximum ECSA value configured. Messages are also issued when ECSA and FIXED storage are constrained.

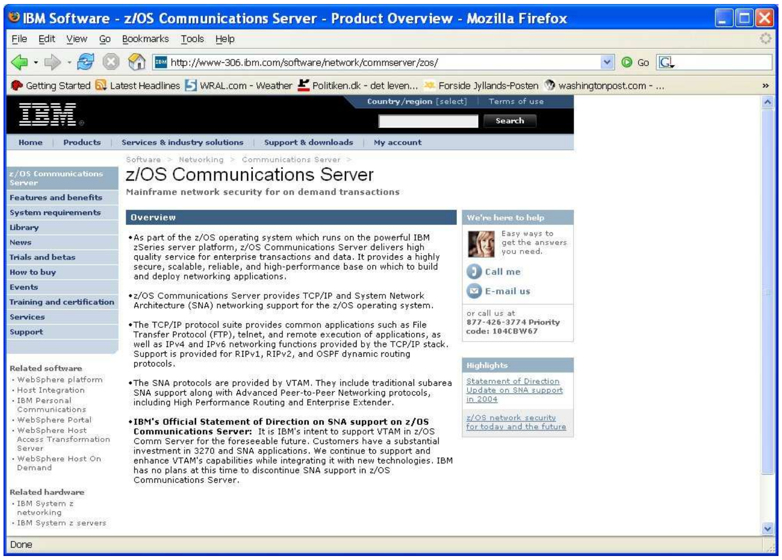
Several new checks were added to the Health-checker in z/OS V1R9 Communications Server. Checks were added for both TCP/IP and VTAM.

In z/OS V1R9 Communications Server, the PKTTRACE command supports the PORTNUM keyword to collect packets with a matching destination and source port number for TCP or UDP packets.

Various RAS items were implement in z/OS Communications Server for V1R9.

z/OS Communications Server home page

URL: http://www.ibm.com/software/network/commserver/zos


IBM Software - z/OS Communications Server - Product Overview - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www-306.ibm.com/software/network/commserver/zos/
Go
Getting Started
Latest Headlines
WRAL.com - Weather
Politiken.dk - det leven...
Forside Jyllands-Posten
washingtonpost.com - ...
IBM
Country/region [select]
Terms of use
Search
Home
Products
Services & industry solutions
Support & downloads
My account
Software > Networking > Communications Server >
z/OS Communications Server
Features and benefits
System requirements
Library
News
Trials and betas
How to buy
Events
Training and certification
Services
Support
Related software
WebSphere platform
Host Integration
IBM Personal Communications
WebSphere Portal
WebSphere Host Access Transformation Server
WebSphere Host On Demand
Related hardware
IBM System z networking
IBM System z servers
We're here to help
Call me
E-mail us
Highlights
Statement of Direction Update on SNA support in 2004
z/OS network security for today and the future
Done

z/OS Communications Server

Mainframe network security for on demand transactions

Overview

- As part of the z/OS operating system which runs on the powerful IBM zSeries server platform, z/OS Communications Server delivers high quality service for enterprise transactions and data. It provides a highly secure, scalable, reliable, and high-performance base on which to build and deploy networking applications.
- z/OS Communications Server provides TCP/IP and System Network Architecture (SNA) networking support for the z/OS operating system.
- The TCP/IP protocol suite provides common applications such as File Transfer Protocol (FTP), telnet, and remote execution of applications, as well as IPv4 and IPv6 networking functions provided by the TCP/IP stack. Support is provided for RIPv1, RIPv2, and OSPF dynamic routing protocols.
- The SNA protocols are provided by VTAM. They include traditional subarea SNA support along with Advanced Peer-to-Peer Networking protocols, including High Performance Routing and Enterprise Extender.
- **IBM's Official Statement of Direction on SNA support on z/OS Communications Server:** It is IBM's intent to support VTAM in z/OS Comm Server for the foreseeable future. Customers have a substantial investment in 3270 and SNA applications. We continue to support and enhance VTAM's capabilities while integrating it with new technologies. IBM has no plans at this time to discontinue SNA support in z/OS Communications Server.

Easy ways to get the answers you need.

or call us at **877-426-3774 Priority code: 104CBW67**

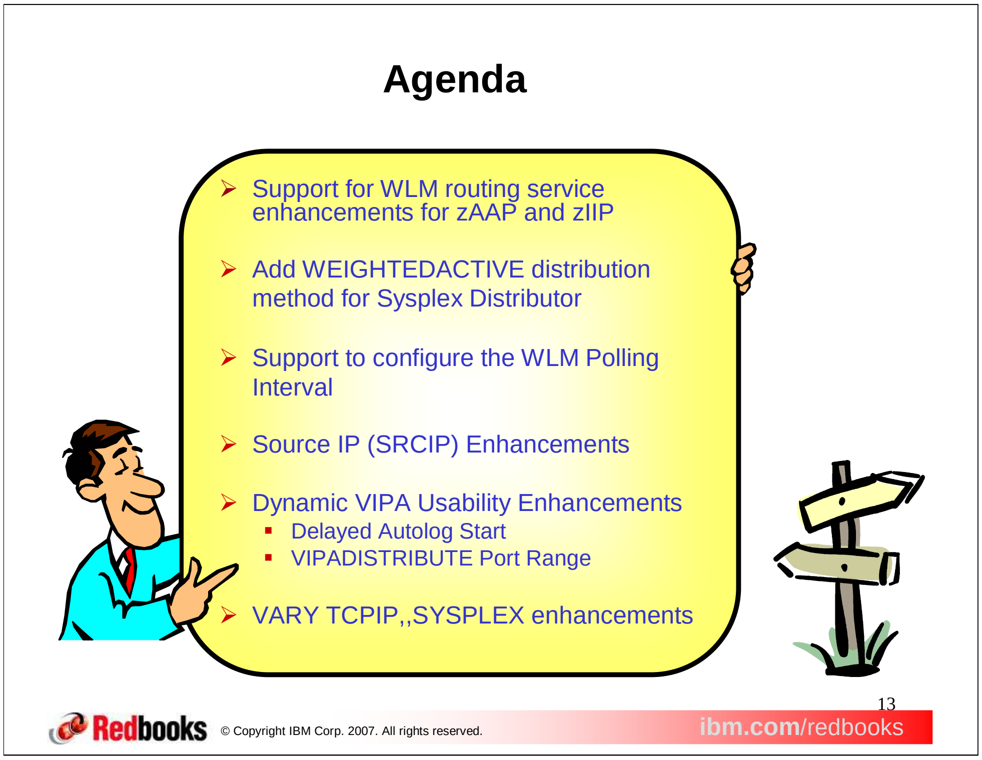
ibm.com

Sysplex Networking

e-business
/www.
IBM
Redbooks

International Technical Support Organization

This presentation describes the sysplex enhancements in Communications Server for z/OS V1R9.



Agenda

- Support for WLM routing service enhancements for zAAP and zIIP
- Add WEIGHTEDACTIVE distribution method for Sysplex Distributor
- Support to configure the WLM Polling Interval
- Source IP (SRCIP) Enhancements
- Dynamic VIPA Usability Enhancements
  - Delayed Autolog Start
  - VIPADISTRIBUTE Port Range
- VARY TCPIP,,SYSPLEX enhancements

13

Redbooks

ibm.com/redbooks

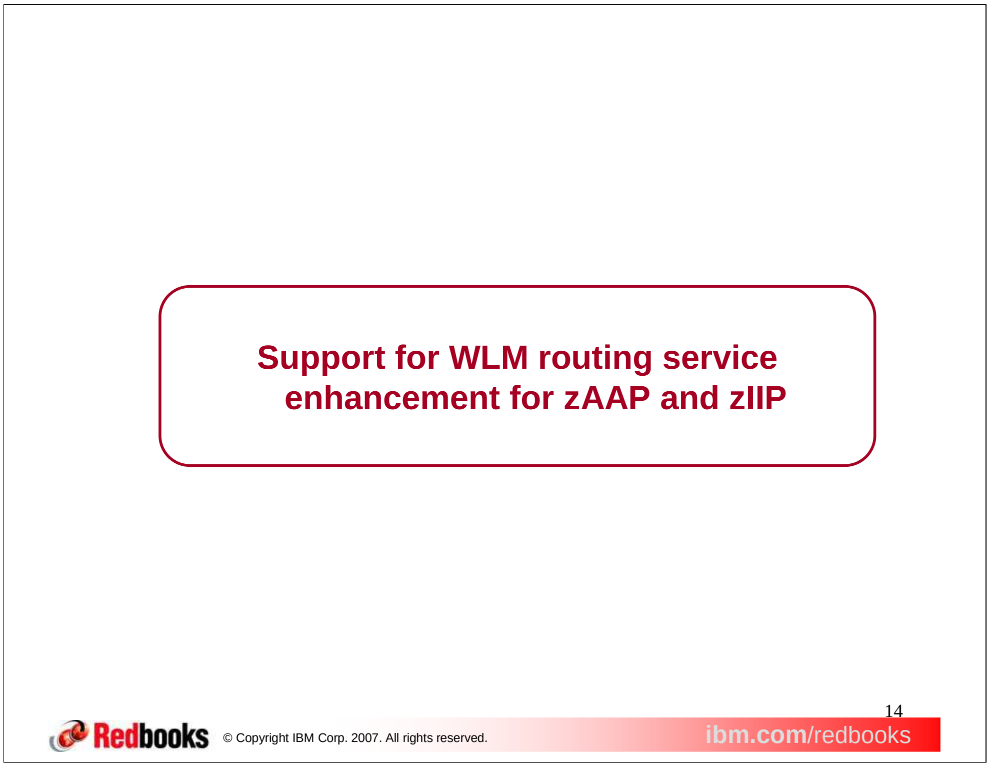
Sysplex Distributor and the Load Balancing Advisor use WLM information about the specialty processors, zAAP and zIIP, for workload balancing.

Support for a new distribution method WEIGHTEDActive is added in this release.

As part of an APAR recently added in V1R6, configuration of a WLM polling interval is also supported.

Enhancements to the SRCIP block has been made to give it more functionality.

We will also be discussing the dynamic VIPA usability and the V TCPIP,,SYSPLEX enhancements that were done in V1R9.

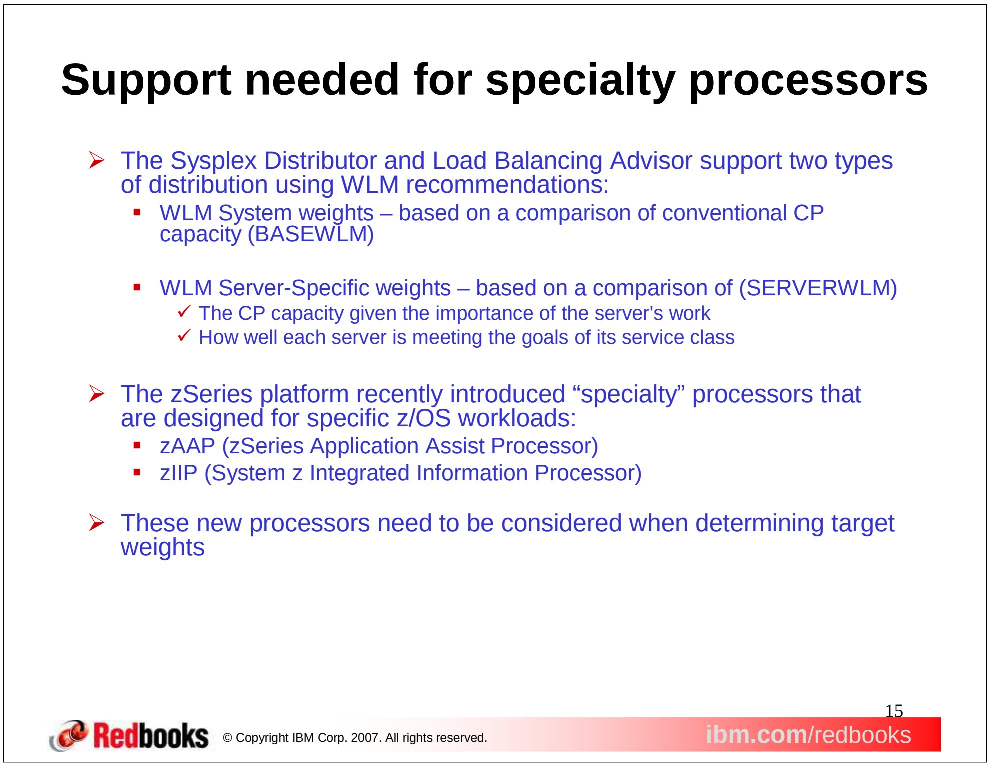


Support for WLM routing service enhancement for zAAP and zIIP

14

ibm.com/redbooks

This section describes the Sysplex Distributor and Load Balancing Advisor enhancements added to support the specialty processors.



Support needed for specialty processors

- The Sysplex Distributor and Load Balancing Advisor support two types of distribution using WLM recommendations:
  - WLM System weights – based on a comparison of conventional CP capacity (BASEWLM)
  - WLM Server-Specific weights – based on a comparison of (SERVERWLM)
    - ✓ The CP capacity given the importance of the server's work
    - ✓ How well each server is meeting the goals of its service class
- The zSeries platform recently introduced “specialty” processors that are designed for specific z/OS workloads:
  - zAAP (zSeries Application Assist Processor)
  - zIIP (System z Integrated Information Processor)
- These new processors need to be considered when determining target weights

15

Redbooks

ibm.com/redbooks

System weights (BaseWLM) and Server-specific weights (ServerWLM) are relative weights that range in value between 0 & 64.

BaseWLM weights are based on a comparison of target Systems in the sysplex

How much CP capacity is available on each system?

When all systems in the sysplex are running at or near 100% utilization, WLM will assign the higher weights to the systems with the largest amounts of lower importance work (systems with the most displaceable capacity).

ServerWLM weights are based on a comparison of target servers within the same service class

How well is a server meeting the goals of its service class?

How much displaceable capacity is available on this system for new work based on the importance of this service class?

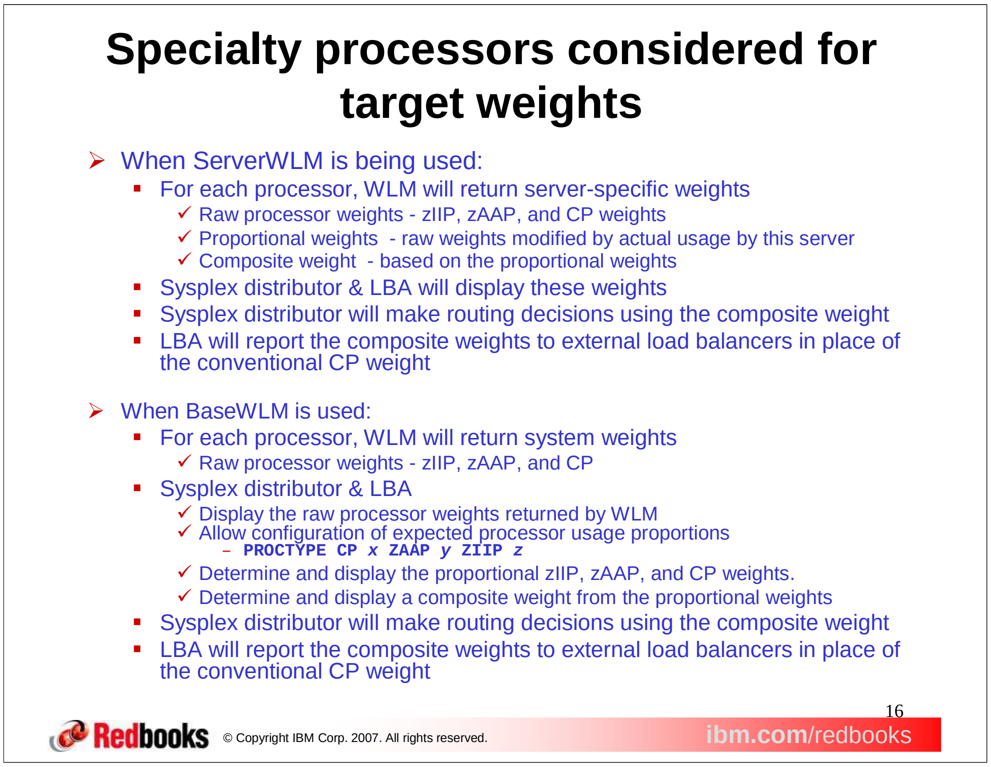
The zSeries platform has recently introduced the notion of specialty processors that can be deployed and exploited by targeted workloads on z/OS. This includes support for:

zAAP (zSeries Application Assist Processor) - these processors can be used for JAVA application workloads on z/OS (including workloads running under WAS).

zIIP (System z Integrated Information Processor) – they can be used for

•z/OS DB2 related workloads, such as z/OS DB2 workload initiated over the network (i.e. using the DRDA protocol) and DB2 BI (Business Intelligence) workloads (i.e. complex queries).

•z/OS IPSEC workloads



Specialty processors considered for target weights

- When ServerWLM is being used:
  - For each processor, WLM will return server-specific weights
    - ✓ Raw processor weights - zIIP, zAAP, and CP weights
    - ✓ Proportional weights - raw weights modified by actual usage by this server
    - ✓ Composite weight - based on the proportional weights
  - Sysplex distributor & LBA will display these weights
  - Sysplex distributor will make routing decisions using the composite weight
  - LBA will report the composite weights to external load balancers in place of the conventional CP weight

- When BaseWLM is used:
  - For each processor, WLM will return system weights
    - ✓ Raw processor weights - zIIP, zAAP, and CP
  - Sysplex distributor & LBA
    - ✓ Display the raw processor weights returned by WLM
    - ✓ Allow configuration of expected processor usage proportions
      - **PROCTYPE CP *x* ZAAP *y* ZIIP *z***
    - ✓ Determine and display the proportional zIIP, zAAP, and CP weights.
    - ✓ Determine and display a composite weight from the proportional weights
  - Sysplex distributor will make routing decisions using the composite weight
  - LBA will report the composite weights to external load balancers in place of the conventional CP weight

16

Redbooks ibm.com/redbooks

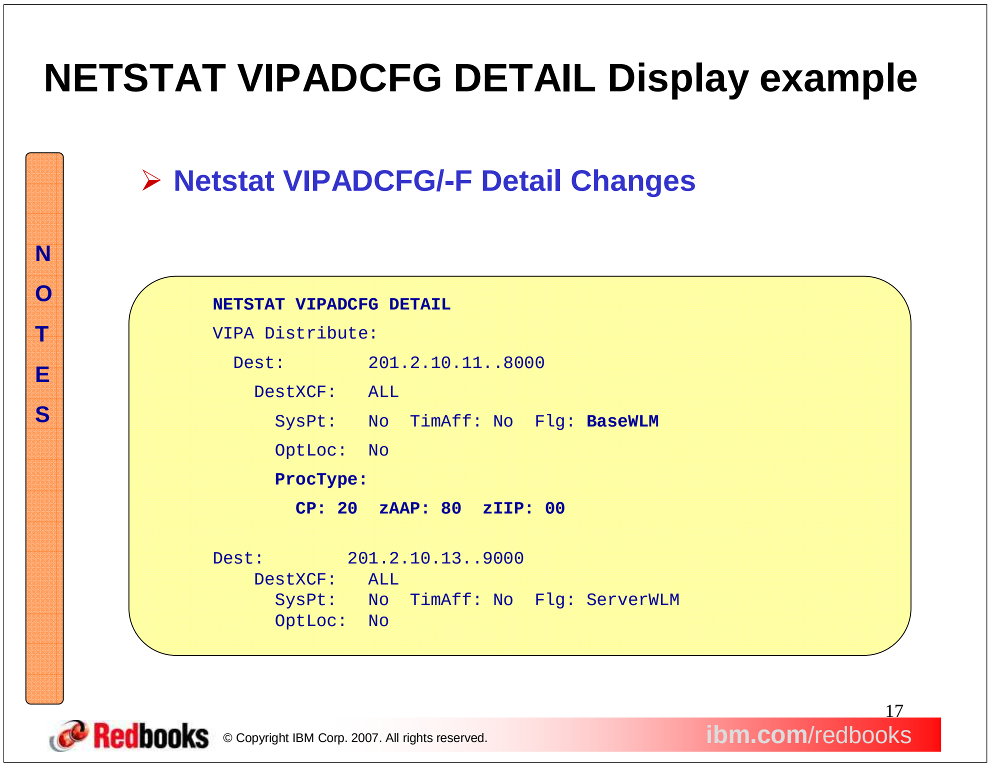
This slide provides an overview of the changes in server-specific weights and system weights support for WLM, Sysplex distributor, and LBA.

For server-specific weights zIIP, zAAP, and CP weights are based on how each server is meeting the goals of its service class and a comparison of that processor’s (available or displaceable) capacity on each target system given the importance of the server’s work. For each processor, WLM will return a composite weight that Sysplex Distributor will use when making routing decisions and the Load Balancing Advisor will report to external load balancers. No additional configuration is required when ServerWLM is being used.

For system weights, zIIP, zAAp, and CP weights are based on the system-level displaceable capacity of each processor type. Because WLM is unaware of how applications are utilizing the various processors, some configuration may be required when BaseWLM is used. In this cases, it will be up to the user to indicate the proportion of each type of processor those workloads will consume. The Communication Server workload distribution technologies retrieve the system WLM raw weights of each type of processor and apply the configured proportions to arrive at the composite weight to be used for workload distribution.

For the Sysplex Distributor, the proctype parameter on the VIPADISTRIBUTE statement can be configured to indicate the expected processor usage for each processor when the distribution method is BaseWLM.   For the LBA, proctype can be defined on the WLM statement or the port_list statement to indicate the expected processor usage when baseWLM is being used. When the wlm parameter is not configured on the port_list, it defaults to the WLM statement configuration.

Values for each processor type, specified on proctype, can range between 0 and 99 so that the proportions can be expressed as percentages if desired. The default for proctype is to only consider convention CP weights and not consider zIIP or zAAP when determining a weight. When proctype is coded, at least one processor type must be specified; any processor types that are not specified will be assigned a value of 0.
Users should evaluate whether SERVERWLM distribution could be used as an alternative to BASEWLM distribution for their application. SERVERWLM has the added advantage that processor proportions will be automatically determined and dynamically updated by WLM based on the actual CPU usage by the application. If BASEWLM is needed, to determine the processor proportions to configure, users need to study their workload usage of processors by analyzing SMF records, and performance monitor reports, such as RMF Workload Activity Reports to determine the expected utilization proportion for each processor type.



NETSTAT VIPADCFG DETAIL Display example

N
O
T
E
S

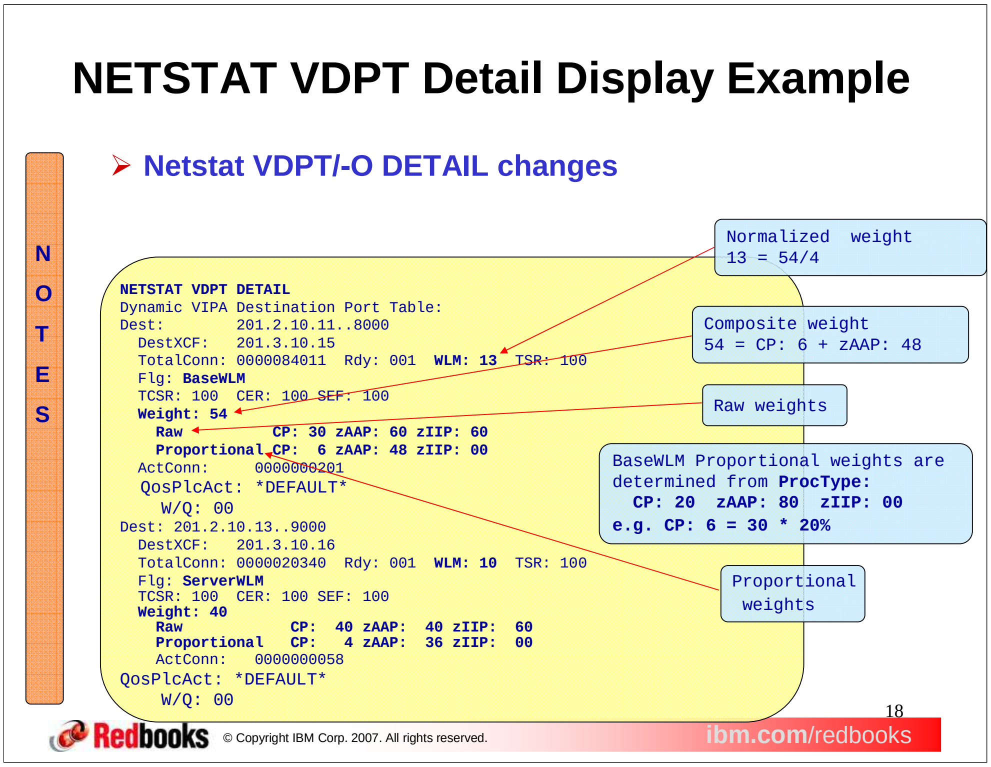
➢ Netstat VIPADCFG/-F Detail Changes

```
NETSTAT VIPADCFG DETAIL
VIPA Distribute:
  Dest:        201.2.10.11..8000
    DestXCF:   ALL
      SysPt:   No  TimAff: No  Flg: BaseWLM
      OptLoc:  No
      ProcType:
        CP: 20  zAAP: 80  zIIP: 00

Dest:        201.2.10.13..9000
    DestXCF:   ALL
      SysPt:   No  TimAff: No  Flg: ServerWLM
      OptLoc:  No
```

Redbooks

ibm.com/redbooks

The VIPADCFG Detail display is modified to display the configured processor proportions for each target when the distribution method is BaseWLM. The proportions will be used to modify the raw weights received from WLM. There are no changes to the display when ServerWLM is being used.



NETSTAT VDPT Detail Display Example

- Netstat VDPT/-O DETAIL changes

```
NETSTAT VDPT DETAIL
Dynamic VIPA Destination Port Table:
Dest:         201.2.10.11..8000
  DestXCF:    201.3.10.15
  TotalConn: 0000084011  Rdy: 001  WLM: 13  TSR: 100
  Flg: BaseWLM
  TCSR: 100  CER: 100 SEF: 100
  Weight: 54
    Raw          CP: 30 zAAP: 60 zIIP: 60
    Proportional CP:  6 zAAP: 48 zIIP: 00
  ActConn:      0000000201
  QosPlcAct: *DEFAULT*
    W/Q: 00
Dest: 201.2.10.13..9000
  DestXCF:    201.3.10.16
  TotalConn: 0000020340  Rdy: 001  WLM: 10  TSR: 100
  Flg: ServerWLM
  TCSR: 100  CER: 100 SEF: 100
  Weight: 40
    Raw            CP:  40 zAAP:  40 zIIP:  60
    Proportional   CP:   4 zAAP:  36 zIIP:  00
    ActConn:    0000000058
QosPlcAct: *DEFAULT*
    W/Q: 00
```

Normalized weight
13 = 54/4

Composite weight
54 = CP: 6 + zAAP: 48

Raw weights

BaseWLM Proportional weights are determined from **ProcType:**
CP: 20 zAAP: 80 zIIP: 00
e.g. CP: 6 = 30 * 20%

Proportional weights

18

ibm.com/redbooks

Use the Netstat VDPT/-O DETAIL report to display the raw weights, proportionally modified weights, raw composite weight and composite weight after normalization when BaseWLM or ServerWLM is being used.

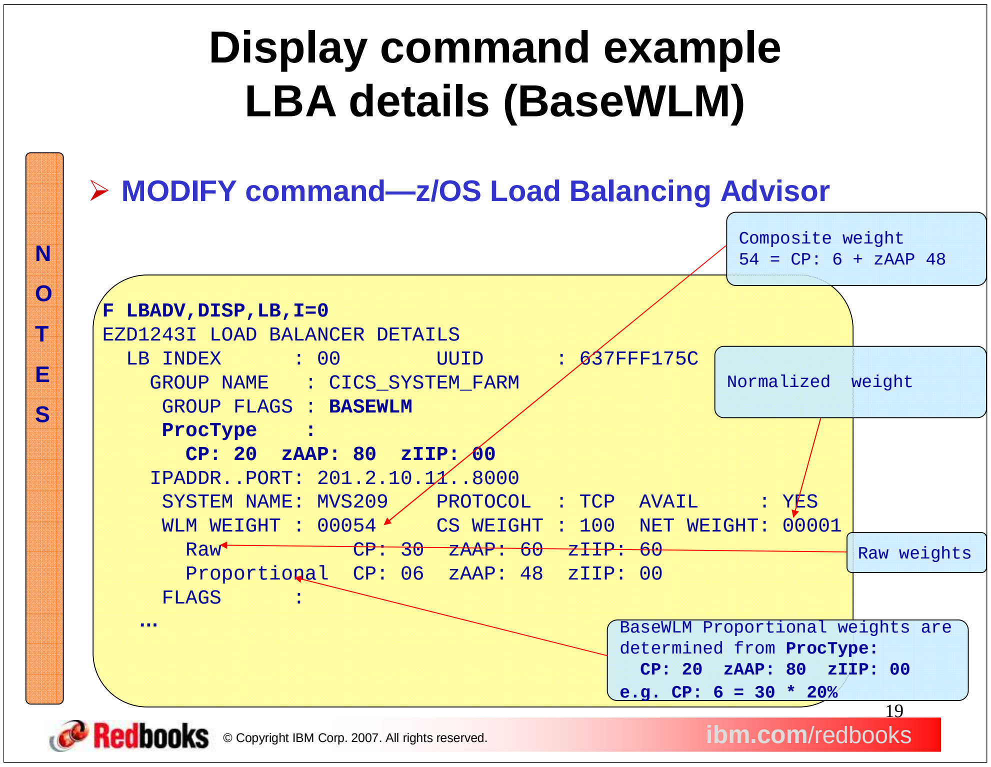
With z/OS V1R9 some of the detailed displays are simplified to only show values that pertain to a distribution method.

This example shows how the weights are determined for BaseWLM given the Proctype configuration on the previous slide of CP 20 zAAP 80.

Looking at the port 8000 BaseWLM target, the WLM weight of 13 is determined as follows. The raw weights, (CP 30, zAAP 60, and zIIP 60), were received from WLM. Each raw weight ranges from 0 through 64. The configured proportions are CP 20 zAAP 80. A processor's proportional fraction is determine by dividing the configured proportion by the sum of all configured processor proportions: CP proportional fraction 20% = CP 20/(CP 20 + zAAP 80 + zIIP 0) and zAAP proportional fraction 80% = zAAP 80/(CP 20 + zAAP 80 + zIIP 0). Each proportional weight is determined using the proportional fraction against the raw weight received from WLM: CP 6 = (Raw CP 30) * (CP Proportional fraction 20%) and zAAP 48 = (Raw zAAP 60) * (zAAP Proportional fraction 80%). The composite raw weight is the sum of the proportional weights (Weight 54 = CP 6 + zAAP 48). The TSR fraction is applied against the composite weight (no change since TSR fraction is 100%). The Normalized weight is determined by dividing the TSR modified weight by 4 (WLM 13 = 54/4).

Looking at the port 9000 ServerWLM target, the weight of 10 is determined as follows. The following raw weights, (CP 40, zAAP 40, and zIIP 60), were received from WLM. The proportional weights, (CP 4, zAAP 36, and zIIP 0), received from WLM are based on current processor usage by the application. The composite weight is the sum of the proportional weights (Weight 40 = CP 4 + zAAP 36).

The TSR fraction is applied against the composite weight (no change since TSR fraction is 100%). The Normalized weight is determined by dividing the TSR modified weight by 4 (WLM 10 = 40/4). The processor usage proportions can be determined from the raw and proportional weights: CP proportion 10% = (WLM CP Proportion: 4)/(WLM Raw CP weight: 40) and zAAP proportion 90% = (zAAP Proportion: 36)/(Raw ZAAP weight: 40).



Display command example LBA details (BaseWLM)

NOTES

➢ MODIFY command—z/OS Load Balancing Advisor

```
F LBADV,DISP,LB,I=0
EZD1243I LOAD BALANCER DETAILS
  LB INDEX      : 00         UUID       : 637FFF175C
    GROUP NAME    : CICS_SYSTEM_FARM
     GROUP FLAGS : BASEWLM
     ProcType     :
       CP: 20  zAAP: 80  zIIP: 00
    IPADDR..PORT: 201.2.10.11..8000
     SYSTEM NAME: MVS209     PROTOCOL  : TCP  AVAIL     : YES
     WLM WEIGHT : 00054      CS WEIGHT : 100  NET WEIGHT: 00001
       Raw            CP: 30  zAAP: 60  zIIP: 60
       Proportional  CP: 06  zAAP: 48  zIIP: 00
     FLAGS       :
   …
```

Composite weight
54 = CP: 6 + zAAP 48

Normalized weight

Raw weights

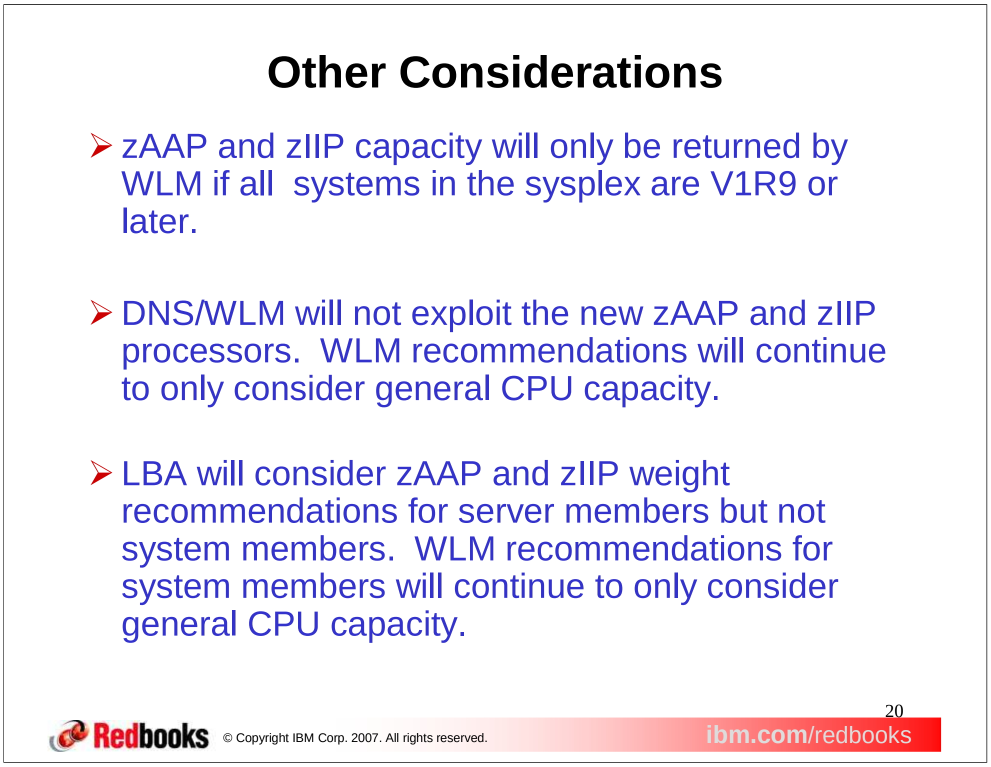
BaseWLM Proportional weights are determined from **ProcType:**
CP: 20 zAAP: 80 zIIP: 00
e.g. CP: 6 = 30 * 20%

19

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

The load balancing advisor detail report shows the proctype proportions configured when BaseWLM is being used (CP: 20 zAAP: 80 zIIP:00), the Raw weights received from WLM (CP: 30 zAAP: 60 zIIP:60), and the proportional weights (CP: 06 zAAP: 48 zIIP: 00). When BaseWLM is being used, the proportional weights are modified by the advisor based on the configured proctype proportions.

LBA determines a normalized weight by dividing by the highest common denominator of all WLM weights received for a Port and Protocol. In this case with only one WLM weight (54), the highest common denominator is 54. 54/54 = 1.



Other Considerations

- zAAP and zIIP capacity will only be returned by WLM if all systems in the sysplex are V1R9 or later.
- DNS/WLM will not exploit the new zAAP and zIIP processors. WLM recommendations will continue to only consider general CPU capacity.
- LBA will consider zAAP and zIIP weight recommendations for server members but not system members. WLM recommendations for system members will continue to only consider general CPU capacity.

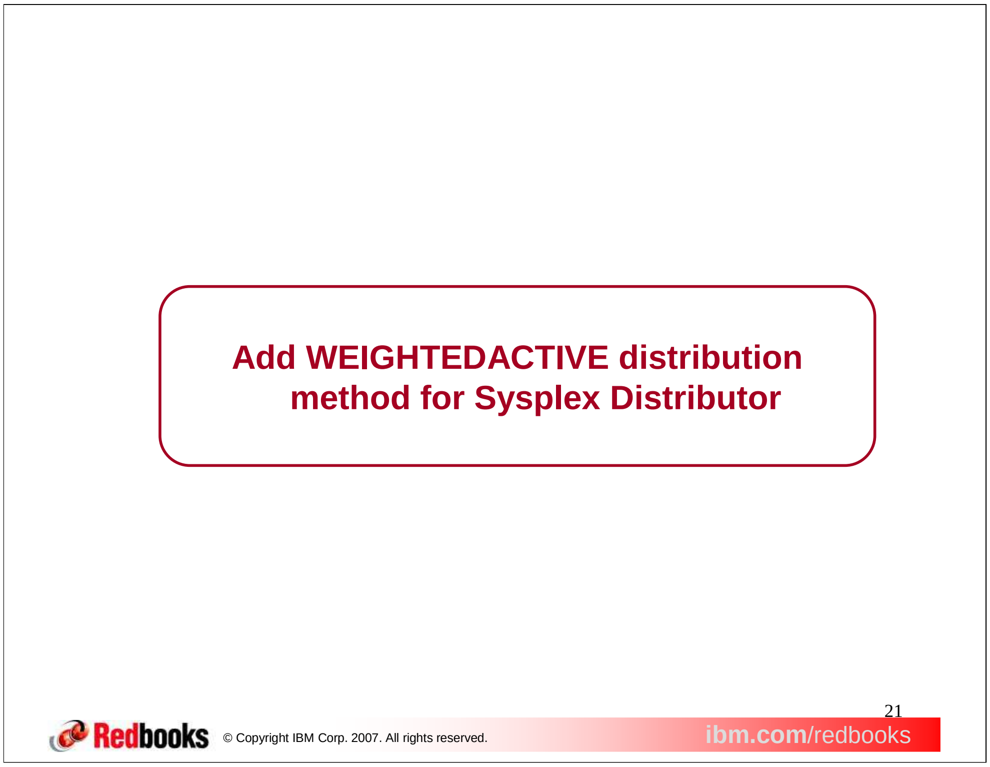
Redbooks ibm.com/redbooks

This first two bullets describe concerns when using zAAP and zIIP capacity in a mixed release environment.

In the 4th bullet,

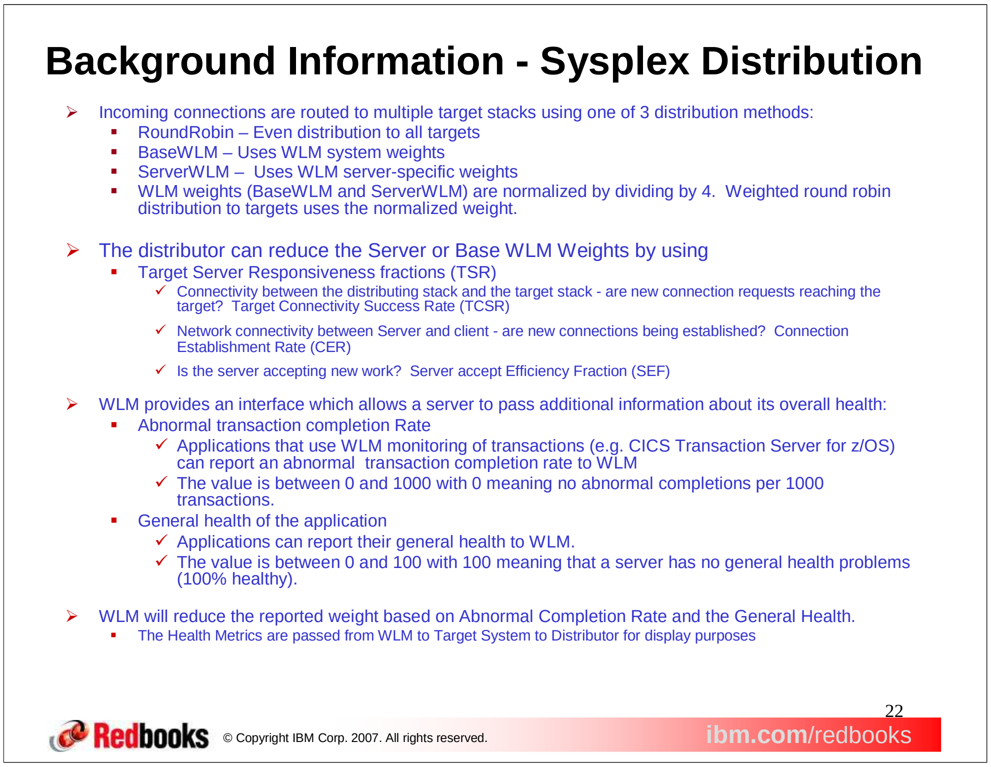
•server members are those that are identified by IP address, port, and protocol - a server member is considered available when there is a protocol “listener” for that IP address, port, and protocol

•system members are only identified by IP address – a system member is considered available when its address is active (in the Home list)



Add WEIGHTEDACTIVE distribution method for Sysplex Distributor

This section describe why support for this function was added and how it works.



Background Information - Sysplex Distribution

- Incoming connections are routed to multiple target stacks using one of 3 distribution methods:
  - RoundRobin – Even distribution to all targets
  - BaseWLM – Uses WLM system weights
  - ServerWLM – Uses WLM server-specific weights
  - WLM weights (BaseWLM and ServerWLM) are normalized by dividing by 4. Weighted round robin distribution to targets uses the normalized weight.
- The distributor can reduce the Server or Base WLM Weights by using
  - Target Server Responsiveness fractions (TSR)
    - ✓ Connectivity between the distributing stack and the target stack - are new connection requests reaching the target? Target Connectivity Success Rate (TCSR)
    - ✓ Network connectivity between Server and client - are new connections being established? Connection Establishment Rate (CER)
    - ✓ Is the server accepting new work? Server accept Efficiency Fraction (SEF)
- WLM provides an interface which allows a server to pass additional information about its overall health:
  - Abnormal transaction completion Rate
    - ✓ Applications that use WLM monitoring of transactions (e.g. CICS Transaction Server for z/OS) can report an abnormal transaction completion rate to WLM
    - ✓ The value is between 0 and 1000 with 0 meaning no abnormal completions per 1000 transactions.
  - General health of the application
    - ✓ Applications can report their general health to WLM.
    - ✓ The value is between 0 and 100 with 100 meaning that a server has no general health problems (100% healthy).
- WLM will reduce the reported weight based on Abnormal Completion Rate and the General Health.
  - The Health Metrics are passed from WLM to Target System to Distributor for display purposes

Redbooks ibm.com/redbooks

Currently, the Sysplex Distributor supports three distribution methods RoundRobin which evenly distributes incoming connections to all targets, BASEWLM which uses WLM system weights, and ServerWLM which uses WLM server-specific weights. When WLM weights are being used, they are normalized or reduced by dividing by 4. Incoming connection requests are distributed based on the normalized weights.

System weights (BaseWLM) and Server-specific weights (ServerWLM) are relative weights that range in value between 0 & 64. BaseWLM weights are based on a comparison of target Systems in the sysplex. How much CPU capacity is available on each system? When all systems in the sysplex are running at or near 100% utilization, WLM will assign the higher weights to the systems with the largest amounts of lower importance work (systems with the most displaceable capacity). The distributor polls WLM for system weights each minute. ServerWLM weights are based on a comparison of target servers within the same service class. How well is a server meeting the goals of its service class? How much displaceable capacity is available on this system for new work based on the importance of this service class? The target systems poll WLM for their server weights each minute and forward the weights to the distributor.

The received weights can optionally modified by a QoS Service level fraction. A Service Level fraction measures the performance of the established connections that map to a DVIPA/Port on a target server. This includes the target to client performance, the ratio of retransmits and timeouts to number of packets sent, overall throughput and throughput/connection against desired values, and the ratio of current connections against maximum connection limits. After the fraction is applied the weights are normalized (reduced) by dividing by 4. If all of the received WLM weights for a DVIPA/Port are less than or equal to 16, normalization is not done. After the fraction is applied against the raw weight, the weights are left unchanged.

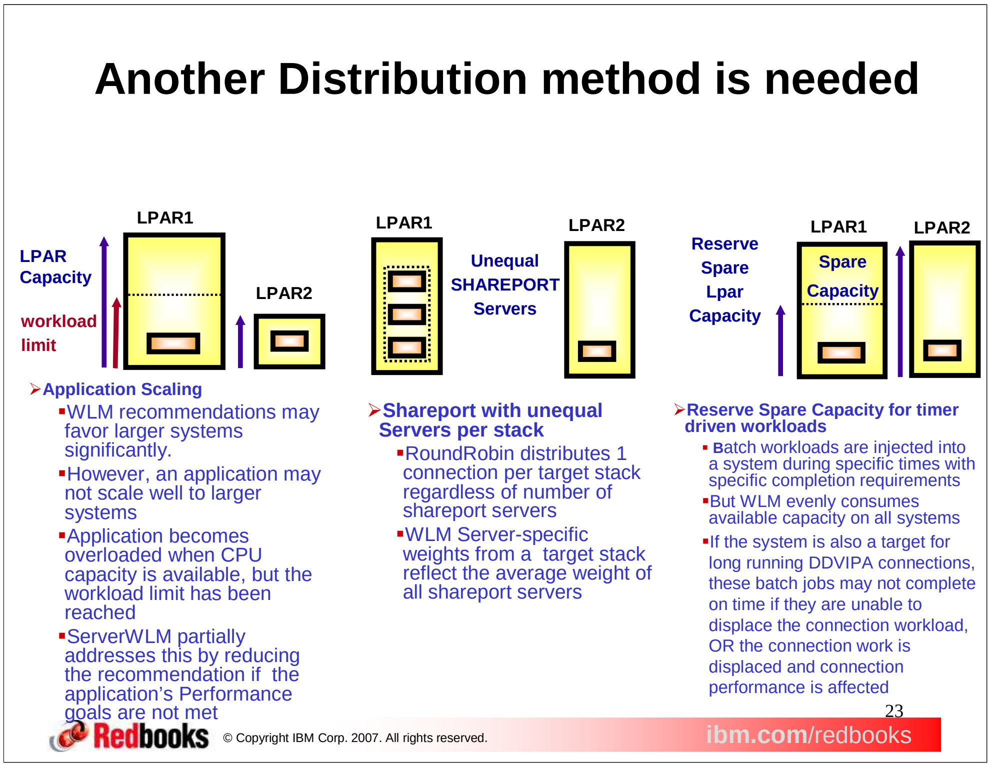
The Target Server Responsiveness (TSR) fraction consists of 3 components, Target Connectivity Success Rate (TCSR) which is a measure of connectivity between the distributing stack and the target stack, Connection Establishment Rate (CER) which is a measure of network connectivity between Server and client (is the 3-way connection set up exchange completing?), and Server accept Efficiency Fraction (SEF) which is a measure of the Target Server’s health. The weights are modified by the TSR fraction, and optionally the QoS fraction, before normalizing.

WLM provides an interface which allows a server to pass additional information about its overall health. The following information may be used to reduce the weight passed to the stack.

•Abnormal transaction completion Rate - Applications such as the CICS Transaction Server for z/OS act as Subsystem Work Managers. They establish WLM Service Class goals, using WLM to monitor transactions against these goals; as part of this monitoring process, they can report an abnormal transaction completion rate to WLM (abnormal completions per 1000 transactions). The value is between 0 and 1000 with 0 meaning no abnormal completions.

•General health of the application - Applications can report their general health to WLM. The value is between 0 and 100 with 100 meaning that a server has no general health problems (100% healthy).

WLM will reduce the reported weight based on Abnormal Completion Rate and the General Health. The Health Metrics are passed from WLM to Target System to Distributor for display purposes.



Another Distribution method is needed

LPAR1
LPAR Capacity
workload limit
LPAR2
LPAR1
Unequal SHAREPORT Servers
LPAR2
Reserve Spare Lpar Capacity
LPAR1
Spare Capacity
LPAR2

- Application Scaling
  - WLM recommendations may favor larger systems significantly.
  - However, an application may not scale well to larger systems
  - Application becomes overloaded when CPU capacity is available, but the workload limit has been reached
  - ServerWLM partially addresses this by reducing the recommendation if the application's Performance goals are not met
- Shareport with unequal Servers per stack
  - RoundRobin distributes 1 connection per target stack regardless of number of shareport servers
  - WLM Server-specific weights from a target stack reflect the average weight of all shareport servers
- Reserve Spare Capacity for timer driven workloads
  - Batch workloads are injected into a system during specific times with specific completion requirements
  - But WLM evenly consumes available capacity on all systems
  - If the system is also a target for long running DDVIPA connections, these batch jobs may not complete on time if they are unable to displace the connection workload, OR the connection work is displaced and connection performance is affected

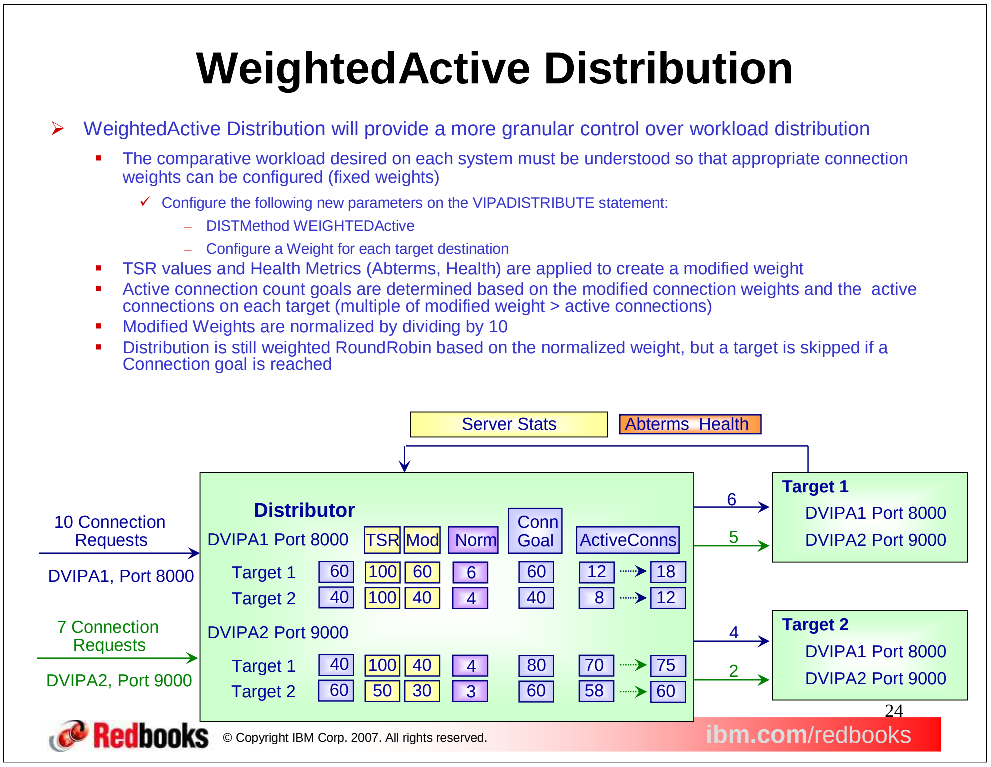
23

ibm.com/redbooks

Application Scaling - Target systems can vary significantly in terms of capacity (small systems along with larger systems). WLM recommendations may favor the larger systems significantly. However, a target application may not scale well to larger systems; due to its design it may not be able to take full advantage of the additional CPU capacity on the larger systems. As a result this type of server can get inflated WLM recommendations when running on larger systems causing it to be overloaded with work.

Unequal numbers of SHAREPORT Servers – If SHAREPORT is used, but not all systems have the same number of SHAREPORT server instances (one has 2 the other has 3). The current RR or WLM recommendations do not change distribution based on the number of server instances on each target. RR distributes 1 connection per target stack regardless of the number of shareport server instances on that stack. WLM Server-specific weights from a target stack with multiple server instances reflect the average weight.

No reservation ability for time driven workloads - Userss would like to reserve some capacity on certain systems for types of batch workloads that run during specific time periods with specific completion requirements. If that system is also a target for long running DDVIPA connections, WLM recommendations will allow that available capacity to be consumed. This could potentially impact the completion times of the batch jobs if they are not able to displace the existing non-batch workloads or vice versa (the connections on that system may suffer from a performance perspective if the batch jobs displace those workloads).



WeightedActive Distribution

- WeightedActive Distribution will provide a more granular control over workload distribution
  - The comparative workload desired on each system must be understood so that appropriate connection weights can be configured (fixed weights)
    - ✓ Configure the following new parameters on the VIPADISTRIBUTE statement:
      - DISTMethod WEIGHTEDActive
      - Configure a Weight for each target destination
  - TSR values and Health Metrics (Abterms, Health) are applied to create a modified weight
  - Active connection count goals are determined based on the modified connection weights and the active connections on each target (multiple of modified weight > active connections)
  - Modified Weights are normalized by dividing by 10
  - Distribution is still weighted RoundRobin based on the normalized weight, but a target is skipped if a Connection goal is reached

DVIPA1 Port 8000		TSR	Mod	Norm	Conn Goal	ActiveConns	
Target 1	60	100	60	6	60	12	18
Target 2	40	100	40	4	40	8	12
DVIPA2 Port 9000							
Target 1	40	100	40	4	80	70	75
Target 2	60	50	30	3	60	58	60

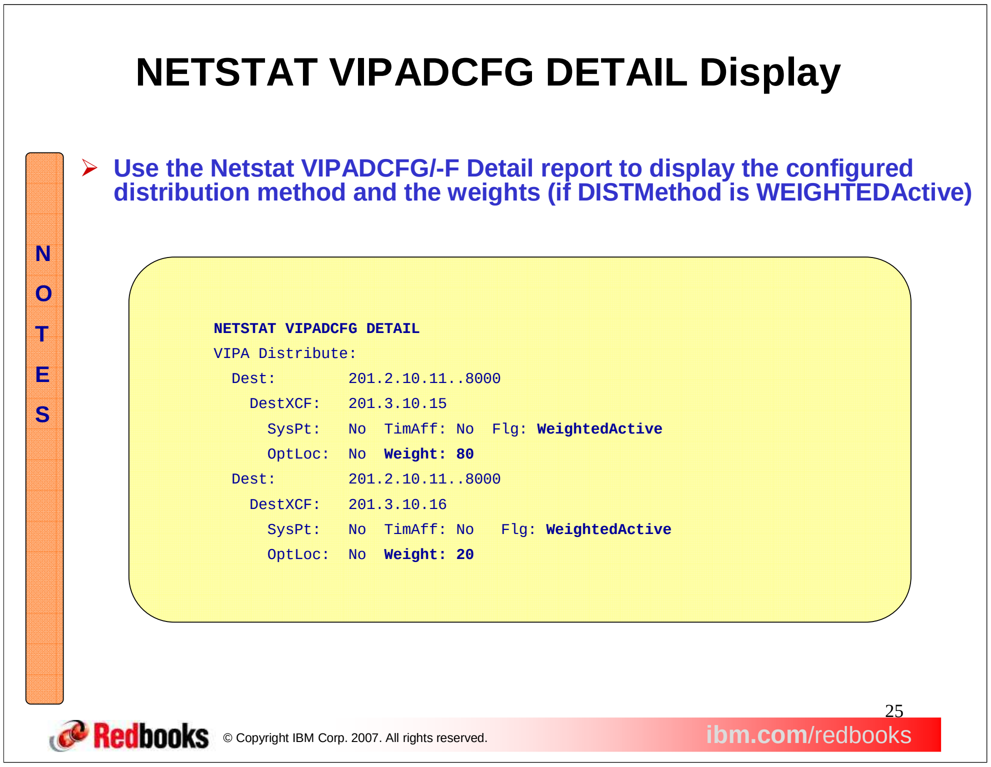
24

Weightedactive Distribution provides more granular control over workload distribution. The comparative workload desired on each system must be understood so that appropriate connection weights can be configured (fixed weights). A new distribution method value of WEIGHTEDActive is added to the DISTMethod parameter. A weight can be configured for each DESTIP destination. Each weight can range in value from 1 to 99 so that the weights can be expressed as percentages if desired. This example was configured using this method; the configured weights added up to 100, so that each weight could be shown as a percentage. Ideally each weight should be greater than 10 so that granularity is preserved when Autonomic fractions need to be applied to determine a modified weight. It defaults to 10, so if DESTIP ALL is configured, then the default weight of 10 is assumed which results in a connection distribution goal to have an equal number of active connections on each target.

The Target Server Responsiveness (TSR) fraction, abnormal completion rate fraction, and General Health fraction are applied against the configured weight to determine a modified weight. Connection goals are established based on the modified weight and the active connection count. Normalized weights are established by dividing the modified weight by 10.

In the example, the Port 9000 Server distribution is determined as follows. Based on configuration, it is desired that Target 1 will have 40% of the connection load and Target 2 will have 60% of the connection load. Since the TSR, abnormal terminations, and health are normal for Target 1, but 50 % for Target 2, the modified weight for Target 1 is 40 and the modified weight for Target 2 is 30 (60 * 50%). The Active Connection Goal is a value for each target such that if achieved would exactly match the desired distribution proportions (it is always a multiple of the modified weight). The total number of active connections for both targets is 128. The total modified weight is 70. The multiplier used determine the connection weight goal is 2 (128/70 + 1). Active connection goals are determined using the modified weight of each target and the multiplier. Target 1's goal is 80 (40 * 2) and Target 2's goal is 60 (30 * 2). The Normalized weight is the modified weight divided by 10. Therefore Target 1's normalized weight is 4 (40/10) and Target 2's normalized weight is 3 (30/10). As 7 connection requests are received: After the first 4 requests are evenly distributed between Target 1 & Target 2, Target 1 will have 72 active connections (Unused Normalized weight is 2) and Target 2 will have 60 active connections (Unused Normalized weight is 1). The next 3 requests will go to Target 1; although the normalized weight for Target 2 is not used up, the connection goal of 60 has been reached while Target 1's connection goal of 80 has not been reached. Assuming that the active connection counts do not change, the next 5 connection requests will go to Target 1. At this point both Target 1 and 2 will have reached their connection goals so the next connection request will cause a calculation of new target goals.

The existing MIB object, ibmMvsDVIPADistConfDistMethod, will indicate if WeightedActive Distribution is configured. A new MIB object, ibmMvsDVIPADistConfTargetWeight, will display the configured weight for each target.

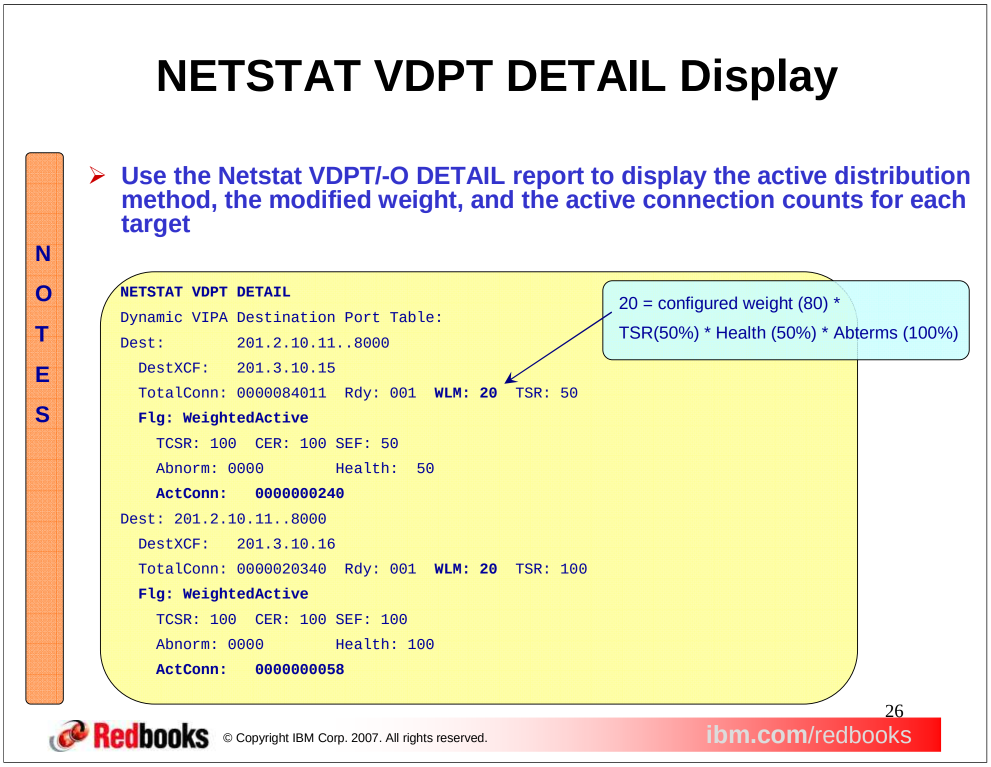


NETSTAT VIPADCFG DETAIL Display

- Use the Netstat VIPADCFG/-F Detail report to display the configured distribution method and the weights (if DISTMethod is WEIGHTEDActive)

```
NETSTAT VIPADCFG DETAIL
VIPA Distribute:
  Dest:         201.2.10.11..8000
    DestXCF:    201.3.10.15
      SysPt:    No  TimAff: No  Flg: WeightedActive
      OptLoc:   No  Weight: 80
  Dest:         201.2.10.11..8000
    DestXCF:    201.3.10.16
      SysPt:    No  TimAff: No   Flg: WeightedActive
      OptLoc:   No  Weight: 20
```

VIPADCFG is modified to display the new configured distribution method of WEIGHTEDActive along with the configured weights for each target.



N
O
T
E
S

NETSTAT VDPT DETAIL Display

- Use the Netstat VDPT/-O DETAIL report to display the active distribution method, the modified weight, and the active connection counts for each target

```
NETSTAT VDPT DETAIL
Dynamic VIPA Destination Port Table:
Dest:        201.2.10.11..8000
  DestXCF:    201.3.10.15
  TotalConn: 0000084011  Rdy: 001  WLM: 20  TSR: 50
  Flg: WeightedActive
    TCSR: 100  CER: 100 SEF: 50
    Abnorm: 0000        Health:  50
    ActConn:   0000000240
Dest: 201.2.10.11..8000
  DestXCF:    201.3.10.16
  TotalConn: 0000020340  Rdy: 001  WLM: 20  TSR: 100
  Flg: WeightedActive
    TCSR: 100  CER: 100 SEF: 100
    Abnorm: 0000        Health: 100
    ActConn:   0000000058
```

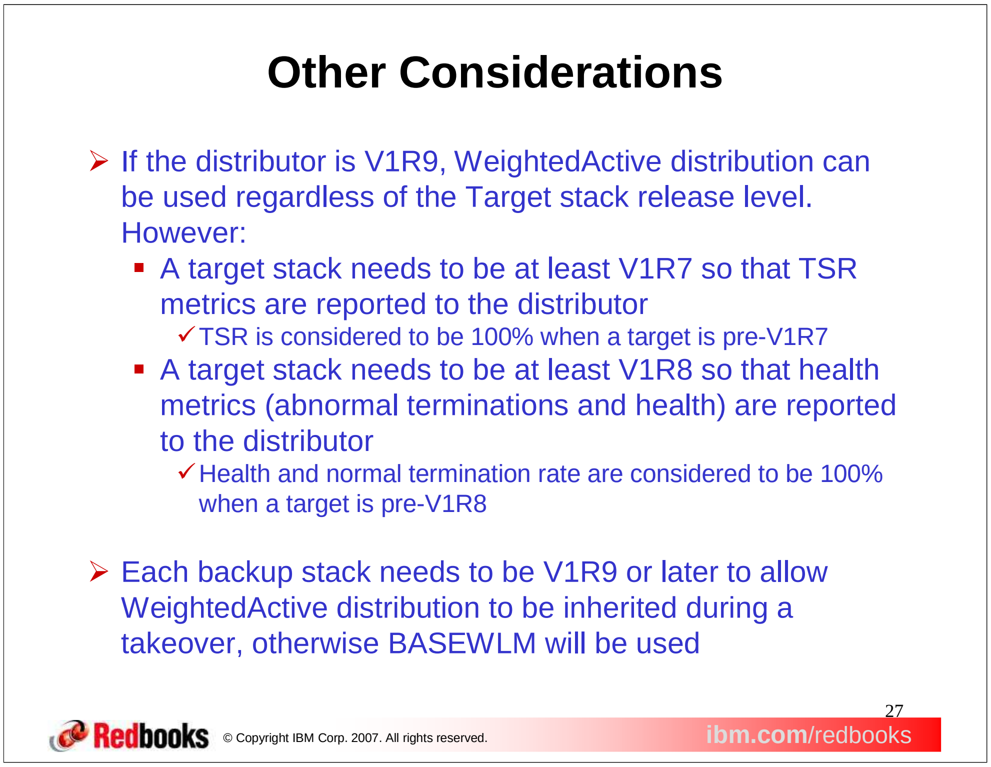
20 = configured weight (80) * TSR(50%) * Health (50%) * Abterms (100%)

26

ibm.com/redbooks

With this release some of the detailed displays are simplified to only show values that pertain to a distribution method.

The detailed version of the report when the distribution method is WEIGHTEDActive shows the TSR metric components (TCSR, CER, and SEF) and the Health metrics (abnormal terminations and health) that are used to determine the modified weight along with the active connection counts.



Other Considerations

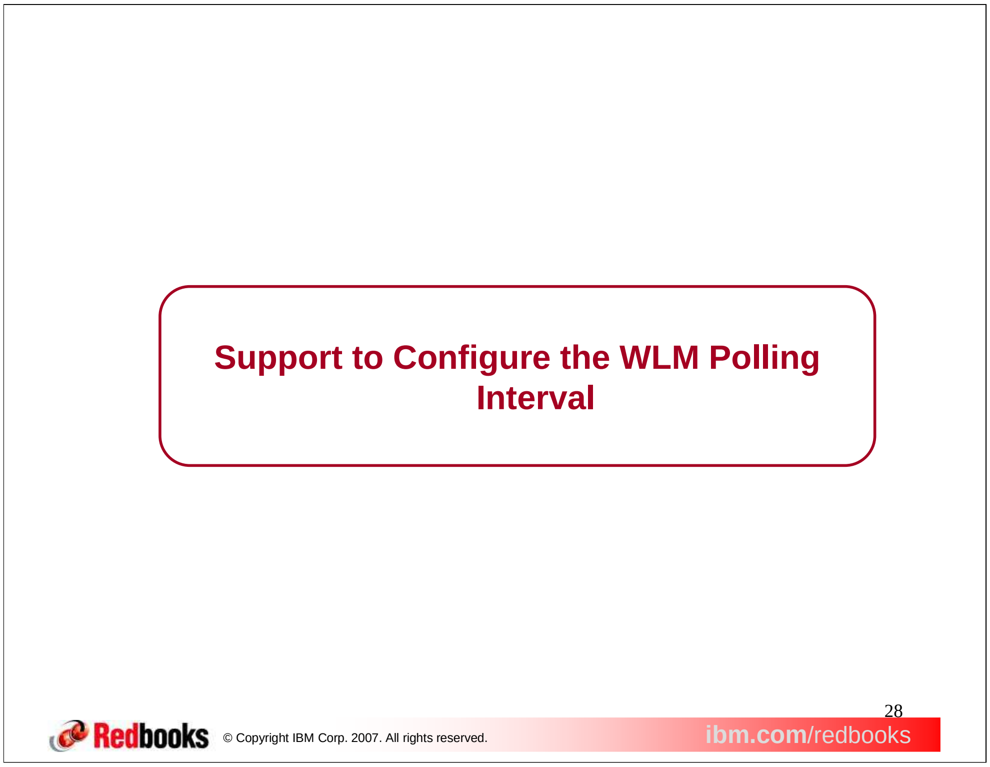
- If the distributor is V1R9, WeightedActive distribution can be used regardless of the Target stack release level. However:
  - A target stack needs to be at least V1R7 so that TSR metrics are reported to the distributor
    - ✓TSR is considered to be 100% when a target is pre-V1R7
  - A target stack needs to be at least V1R8 so that health metrics (abnormal terminations and health) are reported to the distributor
    - ✓Health and normal termination rate are considered to be 100% when a target is pre-V1R8

- Each backup stack needs to be V1R9 or later to allow WeightedActive distribution to be inherited during a takeover, otherwise BASEWLM will be used

Redbooks

ibm.com/redbooks

This slide describes concerns when systems are in a mixed release environment.

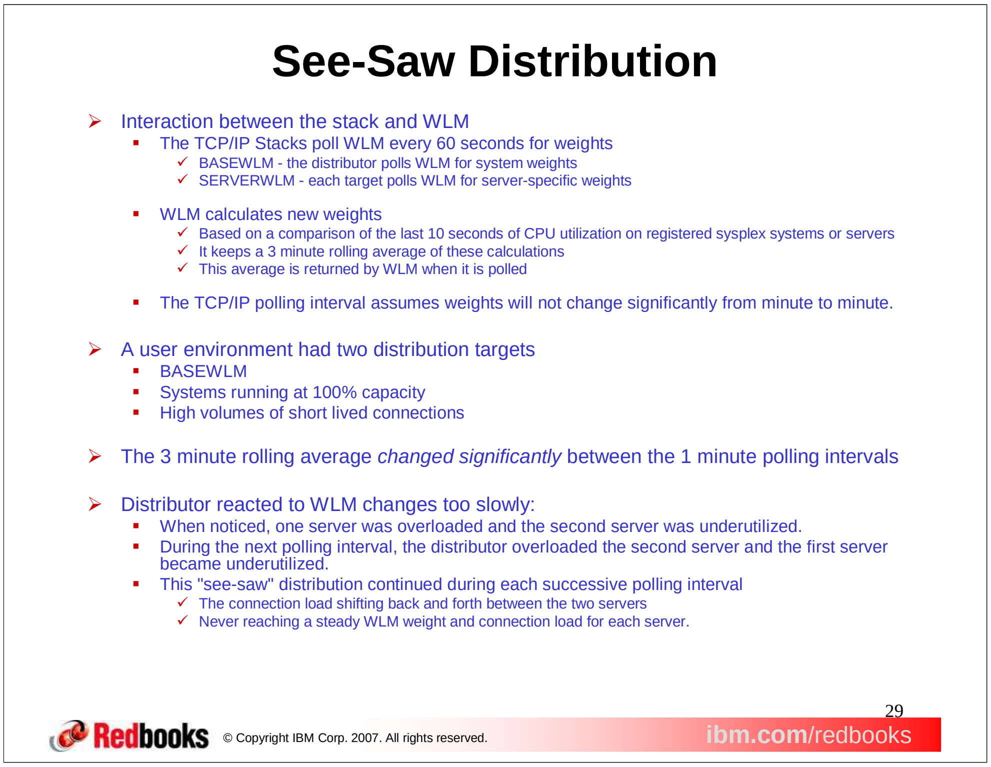


Support to Configure the WLM Polling Interval

28

Redbooks
ibm.com/redbooks

The next group of slides describe a new function APAR which allows the sysplex WLM polling interval to be configured.



See-Saw Distribution

- Interaction between the stack and WLM
  - The TCP/IP Stacks poll WLM every 60 seconds for weights
    - ✓ BASEWLM - the distributor polls WLM for system weights
    - ✓ SERVERWLM - each target polls WLM for server-specific weights
  - WLM calculates new weights
    - ✓ Based on a comparison of the last 10 seconds of CPU utilization on registered sysplex systems or servers
    - ✓ It keeps a 3 minute rolling average of these calculations
    - ✓ This average is returned by WLM when it is polled
  - The TCP/IP polling interval assumes weights will not change significantly from minute to minute.
- A user environment had two distribution targets
  - BASEWLM
  - Systems running at 100% capacity
  - High volumes of short lived connections
- The 3 minute rolling average *changed significantly* between the 1 minute polling intervals
- Distributor reacted to WLM changes too slowly:
  - When noticed, one server was overloaded and the second server was underutilized.
  - During the next polling interval, the distributor overloaded the second server and the first server became underutilized.
  - This "see-saw" distribution continued during each successive polling interval
    - ✓ The connection load shifting back and forth between the two servers
    - ✓ Never reaching a steady WLM weight and connection load for each server.

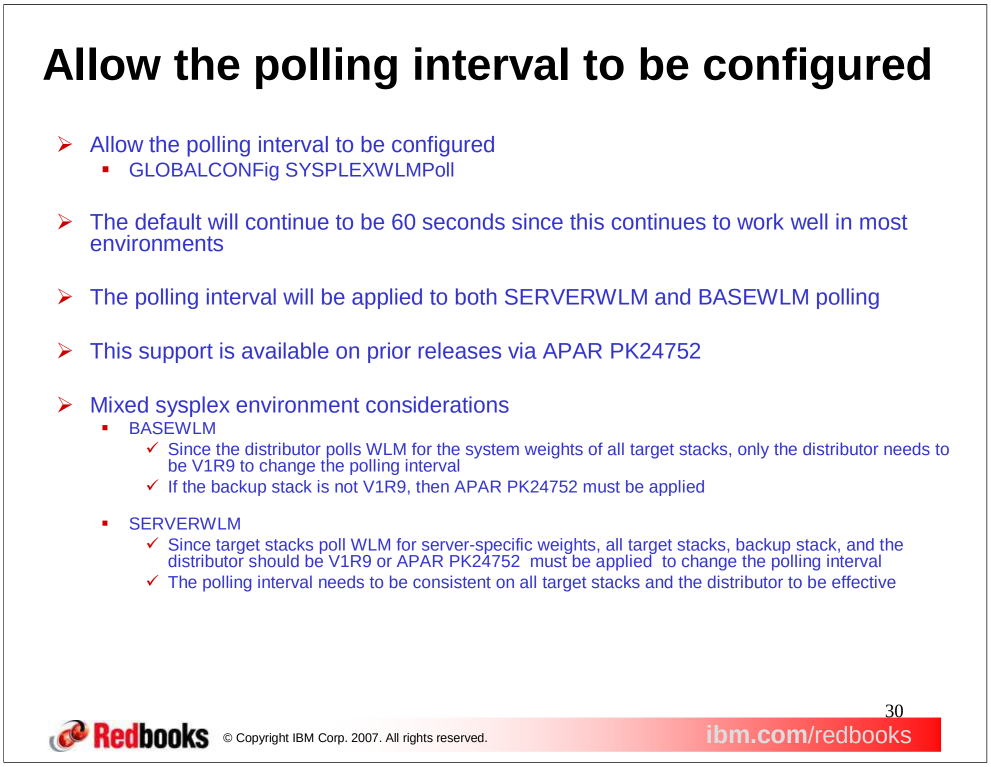
Interaction between the Sysplex stacks and WLM depends on the distribution method. When BASEWLM is being used, the Sysplex Distributor polls WLM every 60 seconds for weights from all systems. When SERVERWLM is being used, each target polls WLM for server-specific weights which are then sent to the distributor. The distributor uses the received WLM weights to determine how to distribute connections to the target systems; a weighted round-robin distribution is used based on the WLM weights.

WLM calculates new weights based on a comparison of the last 10 seconds of CPU utilization on registered sysplex systems. It keeps a three minute rolling average of these calculations. When WLM receives a poll request, the three minute rolling average is returned for each system in the sysplex. The 3 minute average is used to smooth the weight changes.

The distributor's 1 minute polling interval was determined based on the assumption that WLM weights would not change significantly from minute to minute.

In the user environment, the load on target systems was close to 100% capacity with a workload consisting of high volumes of short lived connections. In this type of environment the 3 rolling minute average *changed significantly* between the 1 minute polling intervals. So the original design point was no longer valid for this environment.

The distributor was reacting too slowly to changes in the WLM recommendations between the target servers. At the time the problem was noticed by the user, the first target server was overloaded with connections and the second server was underutilized. From this point on, there was a "see-saw" distribution. During the next minute interval, the new WLM weights caused the distributor to direct most of the connections to the second server causing it to be overloaded and the first server to be underutilized. This continued from interval to interval with the distributor continually shifting most of the connection load back and forth between the two servers, never reaching a steady WLM weight and connection load for each server.



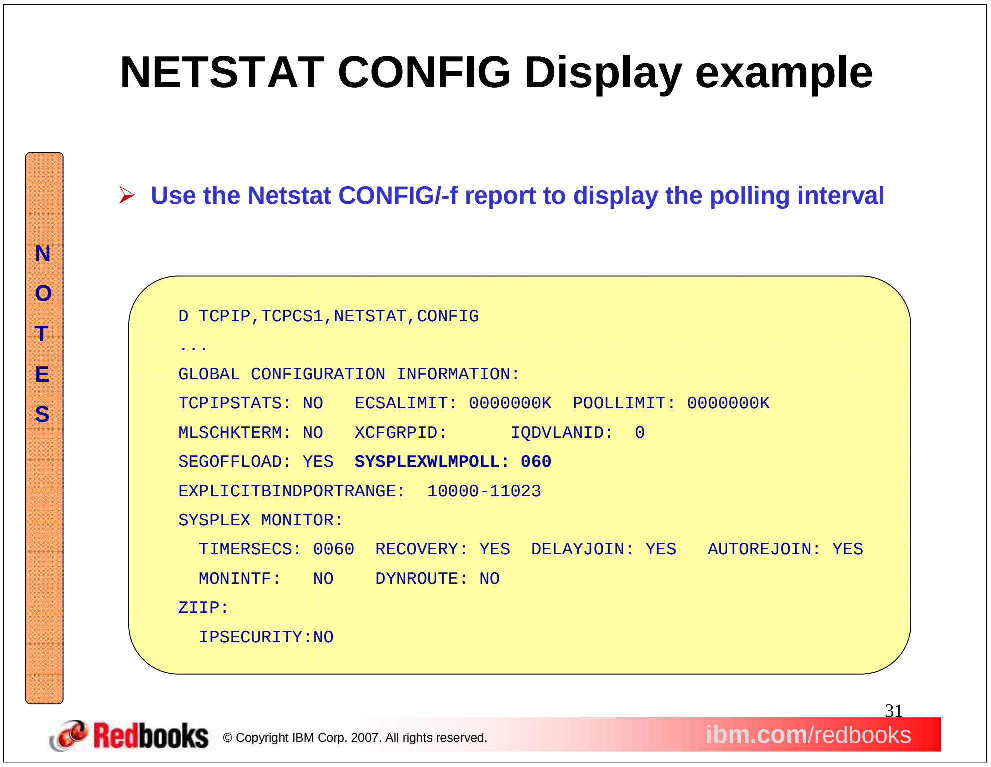
Allow the polling interval to be configured

- Allow the polling interval to be configured
  - GLOBALCONFig SYSPLEXWLMPoll
- The default will continue to be 60 seconds since this continues to work well in most environments
- The polling interval will be applied to both SERVERWLM and BASEWLM polling
- This support is available on prior releases via APAR PK24752
- Mixed sysplex environment considerations
  - BASEWLM
    - ✓ Since the distributor polls WLM for the system weights of all target stacks, only the distributor needs to be V1R9 to change the polling interval
    - ✓ If the backup stack is not V1R9, then APAR PK24752 must be applied
  - SERVERWLM
    - ✓ Since target stacks poll WLM for server-specific weights, all target stacks, backup stack, and the distributor should be V1R9 or APAR PK24752 must be applied to change the polling interval
    - ✓ The polling interval needs to be consistent on all target stacks and the distributor to be effective

30

Redbooks ibm.com/redbooks

A new parameter, SYSPLEXWLMPoll, is added to the GLOBALCONFig statement to allow a user to control the polling interval. It can range between 1 and 60 seconds. The default polling interval will remain 60 seconds. As a guideline the polling interval should not be lower than the WLM weight calculation interval (currently 10 seconds). The polling interval will apply to both SERVERWLM and BASEWLM polling.
In a mixed sysplex environment where some of the systems are not at the V1R9 level, then APAR PK24752 may need to be applied to the pre-V1R9 system to activate this support.



NETSTAT CONFIG Display example

N O T E S

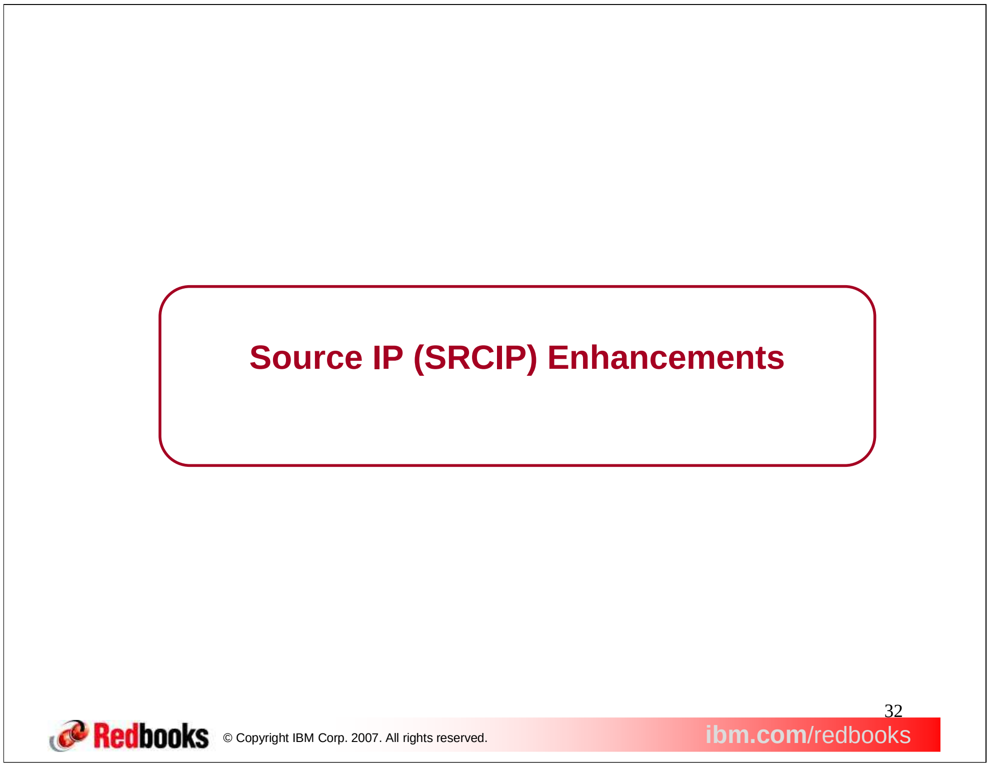
- Use the Netstat CONFIG/-f report to display the polling interval

```
D TCPIP,TCPCS1,NETSTAT,CONFIG
...
GLOBAL CONFIGURATION INFORMATION:
TCPIPSTATS: NO   ECSALIMIT: 0000000K  POOLLIMIT: 0000000K
MLSCHKTERM: NO   XCFGRPID:      IQDVLANID:  0
SEGOFFLOAD: YES  SYSPLEXWLMPOLL: 060
EXPLICITBINDPORTRANGE:  10000-11023
SYSPLEX MONITOR:
  TIMERSECS: 0060  RECOVERY: YES  DELAYJOIN: YES   AUTOREJOIN: YES
  MONINTF:   NO    DYNROUTE: NO
ZIIP:
  IPSECURITY:NO
```

31

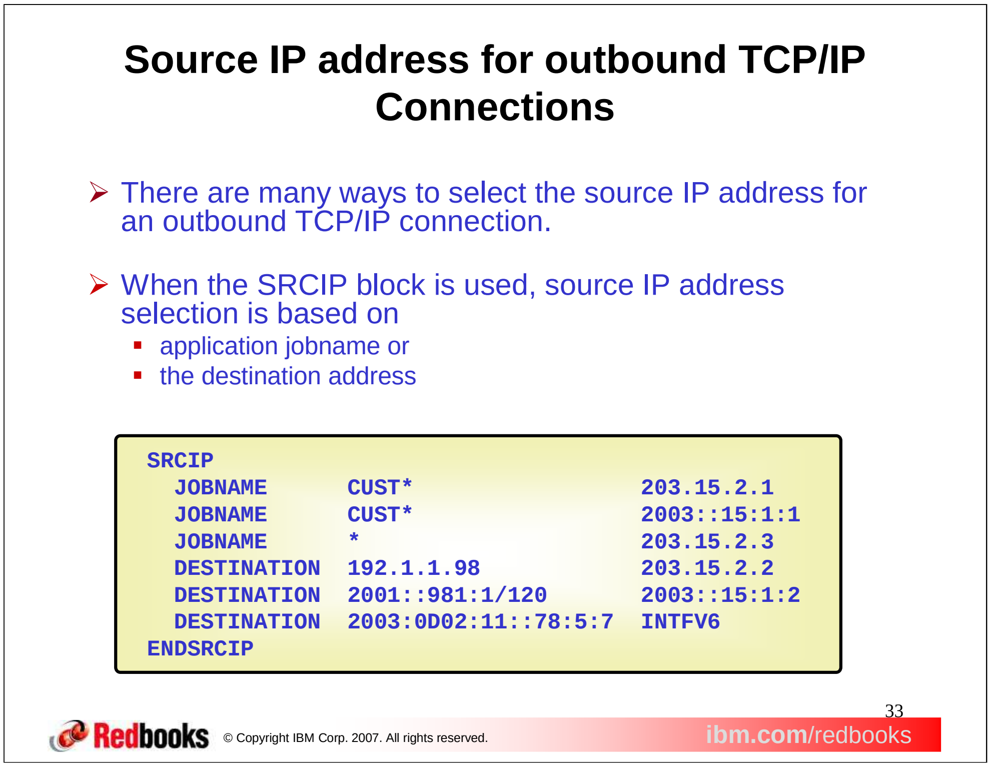
Redbooks ibm.com/redbooks

The sysplex WLM polling interval will be displayed as part of the Global Configuration Information.



Source IP (SRCIP) Enhancements

This section describes the z/OS Communications Server Source IP Enhancements for V1R9.



Source IP address for outbound TCP/IP Connections

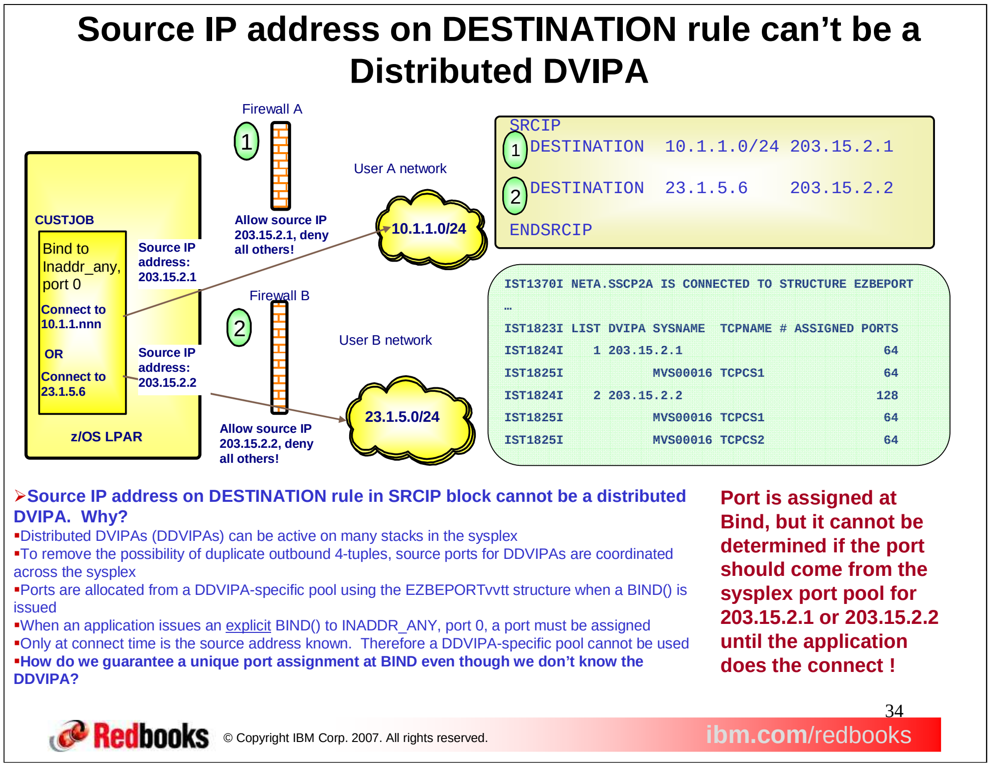
- There are many ways to select the source IP address for an outbound TCP/IP connection.
- When the SRCIP block is used, source IP address selection is based on
  - application jobname or
  - the destination address

```
SRCIP
  JOBNAME      CUST*                    203.15.2.1
  JOBNAME      CUST*                    2003::15:1:1
  JOBNAME      *                        203.15.2.3
  DESTINATION  192.1.1.98               203.15.2.2
  DESTINATION  2001::981:1/120          2003::15:1:2
  DESTINATION  2003:0D02:11::78:5:7     INTFV6
ENDSRCIP
```

33

Redbooks ibm.com/redbooks

TCP/IP provides a number of different mechanisms for selecting the source IP address of an outbound connection when an application has not explicitly specified one. This presentation will focus on one of the mechanisms, the SRCIP block. The SRCIP block allows selection of a source IP address based on the applications jobname or based on the destination address. In the example shown, you can use the keyword JOBNAME to specify an application name (or part of a name, using the asterisk as a wildcard), and specify the source IP address to be used if there is a match on that name. Or you can use the DESTINATION keyword to supply a destination IP address. When a connection is made to that destination address, the associated source IP address will be used.



Source IP address on DESTINATION rule can't be a Distributed DVIPA

CUSTJOB
Bind to Inaddr_any, port 0
Connect to 10.1.1.nnn
OR
Connect to 23.1.5.6
Source IP address: 203.15.2.1
Source IP address: 203.15.2.2
z/OS LPAR
Firewall A
1
Allow source IP 203.15.2.1, deny all others!
Firewall B
2
Allow source IP 203.15.2.2, deny all others!
User A network
10.1.1.0/24
User B network
23.1.5.0/24

```
SRCIP
1 DESTINATION  10.1.1.0/24 203.15.2.1

2 DESTINATION  23.1.5.6    203.15.2.2

ENDSRCIP
```

```
IST1370I NETA.SSCP2A IS CONNECTED TO STRUCTURE EZBEPORT
…
IST1823I LIST DVIPA SYSNAME  TCPNAME # ASSIGNED PORTS
IST1824I    1 203.15.2.1                          64
IST1825I             MVS00016 TCPCS1              64
IST1824I    2 203.15.2.2                         128
IST1825I             MVS00016 TCPCS1              64
IST1825I             MVS00016 TCPCS2              64
```

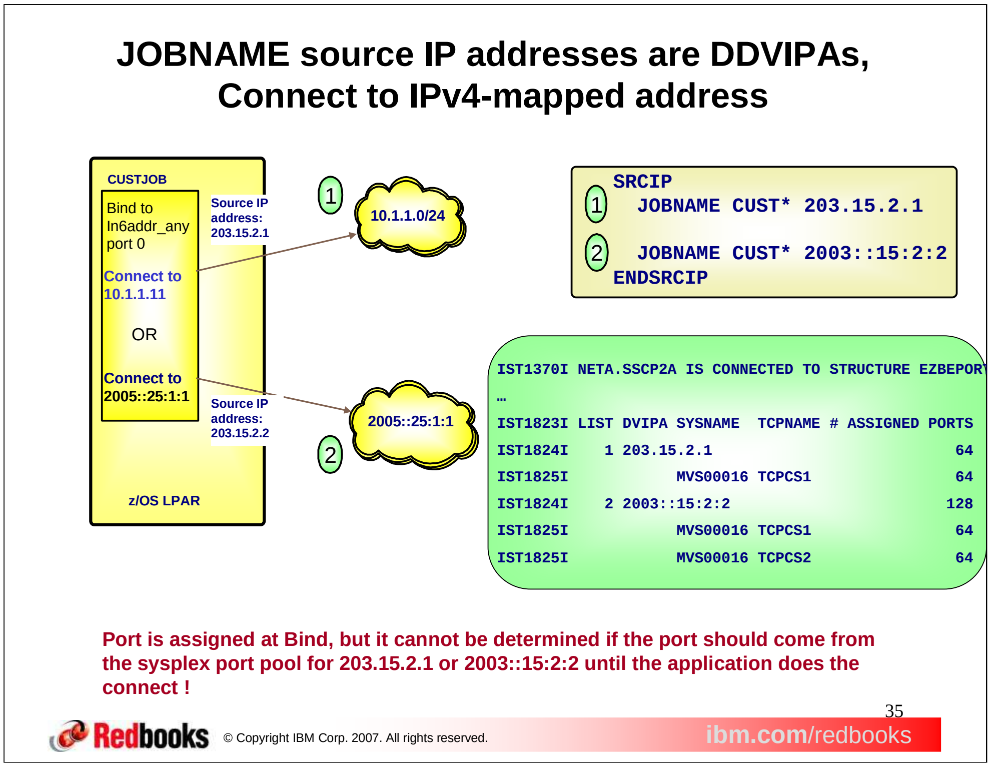
➢**Source IP address on DESTINATION rule in SRCIP block cannot be a distributed DVIPA. Why?**

- Distributed DVIPAs (DDVIPAs) can be active on many stacks in the sysplex
- To remove the possibility of duplicate outbound 4-tuples, source ports for DDVIPAs are coordinated across the sysplex
- Ports are allocated from a DDVIPA-specific pool using the EZBEPORTvvtt structure when a BIND() is issued
- When an application issues an explicit BIND() to INADDR_ANY, port 0, a port must be assigned
- Only at connect time is the source address known. Therefore a DDVIPA-specific pool cannot be used
- **How do we guarantee a unique port assignment at BIND even though we don't know the DDVIPA?**

Port is assigned at Bind, but it cannot be determined if the port should come from the sysplex port pool for 203.15.2.1 or 203.15.2.2 until the application does the connect !

In previous TCP/IP releases, there is a restriction on the type of IP address that may be specified on a SRCIP block DESTINATION rule. Specifically, it could not be a distributed DVIPA. Because distributed DVIPAs may be active on many stacks in the sysplex, source ports that are allocated for connections from these DVIPAs must be coordinated across the sysplex. If they were not, an application on node A in the sysplex, connecting to a destination IP address and port using a specific distributed DVIPA might choose the same port as an application on node B, using the same distributed DVIPA to connect to the same destination IP address and port. This would result in two connection requests with the same 4-tuple (of source IP address, source port, destination IP address, destination port) being sent to the same destination. To prevent this, allocation of source ports (known as sysplexports) for distributed DVIPAs is coordinated using the EZBEPORTvvtt structure, which establishes an allocated source port pool for each specific distributed DVIPA. A problem occurs when an application uses an explicit BIND to INADDR_ANY and port 0 prior to issuing a CONNECT. The BIND protocols require that a port be assigned at this time. However, since the CONNECT has not yet been issued, TCP/IP does not know the destination address, so it would not know to allocate the port from the sysplex port pool associated with the matching DESTINATION rule's source IP address, if that source IP address were a distributed DVIPA.

In this example, you can see that the application CUSTJOB can connect to either an address in the 10.1.1.0 network, or the address 23.1.5.6 in another network. If distributed DVIPAs are specified as the source IP address on the corresponding DESTINATION rules, the source ports should be allocated from the sysplexports pool associated with the specific DVIPA, as depicted in the green box. However, if the CUSTJOB application issues an explicit BIND to inaddr_any and port 0 before doing the CONNECT, TCP/IP cannot tell which specific distributed DVIPA pool to allocate the source port from.



JOBNAME source IP addresses are DDVIPAs, Connect to IPv4-mapped address

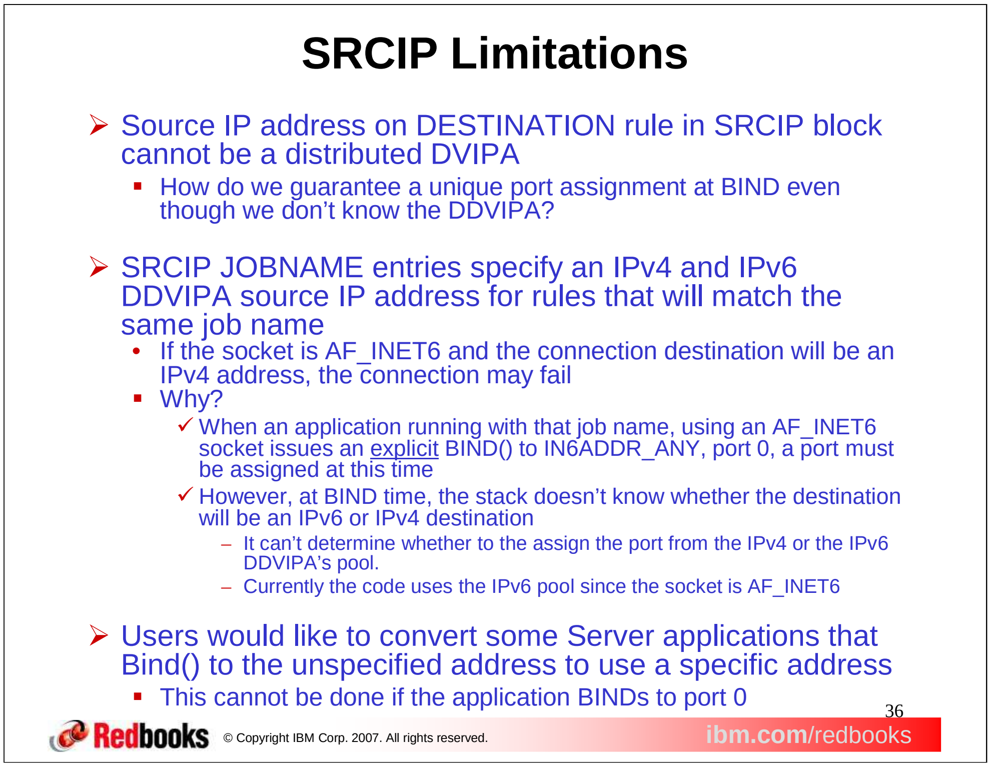
```
SRCIP
(1) JOBNAME CUST* 203.15.2.1
(2) JOBNAME CUST* 2003::15:2:2
ENDSRCIP
```

```
IST1370I NETA.SSCP2A IS CONNECTED TO STRUCTURE EZBEPORT
…
IST1823I LIST DVIPA SYSNAME  TCPNAME # ASSIGNED PORTS
IST1824I    1 203.15.2.1                          64
IST1825I             MVS00016 TCPCS1              64
IST1824I    2 2003::15:2:2                       128
IST1825I             MVS00016 TCPCS1              64
IST1825I             MVS00016 TCPCS2              64
```

Port is assigned at Bind, but it cannot be determined if the port should come from the sysplex port pool for 203.15.2.1 or 2003::15:2:2 until the application does the connect !

35

This diagram illustrates the problem with mapped IPv4 addresses and JOBNAME rules that specify distributed DVIPAs. Here, application CUSTJOB can match either of the two SRCIP JOBNAME rules, one of which specifies an IPv4 distributed DVIPA address, and the other an IPv6 distributed DVIPA address. If CUSTJOB issues a CONNECT to a destination IP address in the IPv4 network 10.1.1.0, TCP/IP can allocate the source IP port from the sysplexport pool associated with the IPv4 source IP address. If CUSTJOB connects to a destination address in the IPv6 network, TCP/IP can allocate the source IP port from the sysplexport pool associated with the IPv6 source IP address. However, if CUSTJOB issues a BIND to in6addr_any and port 0 on an AF_INET6 socket before the CONNECT, TCP/IP cannot know which sysplexport pool to allocate the source port from.



SRCIP Limitations

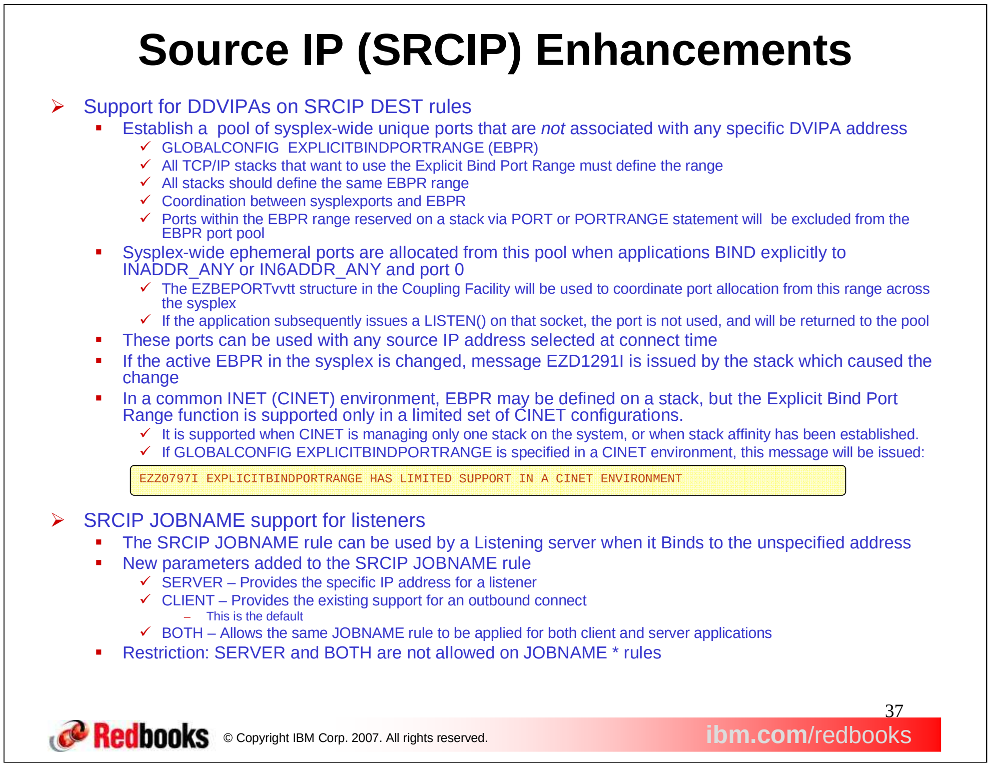
- Source IP address on DESTINATION rule in SRCIP block cannot be a distributed DVIPA
  - How do we guarantee a unique port assignment at BIND even though we don’t know the DDVIPA?
- SRCIP JOBNAME entries specify an IPv4 and IPv6 DDVIPA source IP address for rules that will match the same job name
  - If the socket is AF_INET6 and the connection destination will be an IPv4 address, the connection may fail
  - Why?
    - ✓ When an application running with that job name, using an AF_INET6 socket issues an explicit BIND() to IN6ADDR_ANY, port 0, a port must be assigned at this time
    - ✓ However, at BIND time, the stack doesn’t know whether the destination will be an IPv6 or IPv4 destination
      - It can’t determine whether to the assign the port from the IPv4 or the IPv6 DDVIPA’s pool.
      - Currently the code uses the IPv6 pool since the socket is AF_INET6
- Users would like to convert some Server applications that Bind() to the unspecified address to use a specific address
  - This cannot be done if the application BINDs to port 0

Redbooks ibm.com/redbooks

In previous TCP/IP releases, there is a restriction on the type of IP address that may be specified on a SRCIP block DESTINATION rule. Specifically, it could not be a distributed DVIPA. Because distributed DVIPAs may be active on many stacks in the sysplex, source ports that are allocated for connections from these DVIPAs must be coordinated across the sysplex. If they were not, an application on node A in the sysplex, connecting to a destination IP address and port using a specific distributed DVIPA might choose the same port as an application on node B, using the same distributed DVIPA to connect to the same destination IP address and port. This would result in two connection requests with the same 4-tuple (of source IP address, source port, destination IP address, destination port) being sent to the same destination. To prevent this, allocation of source ports (known as sysplexports) for distributed DVIPAs is coordinated using the EZBEPORTvvtt structure, which establishes an allocated source port pool for each specific distributed DVIPA. A problem occurs when an application uses an explicit BIND to INADDR_ANY and port 0 prior to issuing a CONNECT. The BIND protocols require that a port be assigned at this time. However, since the CONNECT has not yet been issued, TCP/IP does not know the destination address, so it would not know to allocate the port from the sysplex port pool associated with the matching DESTINATION rule’s source IP address, if that source IP address were a distributed DVIPA.

Another problem that can occur is if there are JOBNAME rules in a SRCIP block that can match the same jobname and have an IPv4 and an IPv6 distributed DVIPA source address. Here, if a CONNECT is issued on an AF_INET6 socket and the destination address is an IPv4 address, then the JOBNAME rule with the IPv4 source IP address will be selected, and the source IP port will be allocated from the IPv4 address’s sysplexport pool. If the destination is an IPv6 address, then the JOBNAME rule specifying the IPv6 source IP address will be selected, and the source IP port will be allocated from the IPv6 address’s sysplexport pool. However, if an explicit BIND for IN6ADDR_ANY and port 0 is done before the connect, TCP/IP does not know which sysplexport pool to allocate the port from. TCP/IP chooses the IPv6 pool, since the socket is AF_INET6, which may cause the connection to fail, if the connection is to the IPv4 destination address.

The BIND parameter on the port statement can be used to specify the source IP address when the application binds to the unspecified address and a port in the range of 1 – 65535. However this function can not be used when the application binds to port 0. Users would like the ability to convert the unspecified address to a specific IP address for this case. Here is an example of when this may be desired. A user has multiple z/OS WebSphere Application Server (WAS) instances running in the same z/OS image. One way to provide isolation of these instances is to have each instance associated with a unique VIPA for all of its listening sockets. A single WAS instance may have multiple address spaces acting as TCP listeners and some of these listeners may use ephemeral ports. Existing WAS configuration options and scripts could be used to force each



Source IP (SRCIP) Enhancements

- ➢ Support for DDVIPAs on SRCIP DEST rules
  - Establish a pool of sysplex-wide unique ports that are *not* associated with any specific DVIPA address
    - ✓ GLOBALCONFIG EXPLICITBINDPORTRANGE (EBPR)
    - ✓ All TCP/IP stacks that want to use the Explicit Bind Port Range must define the range
    - ✓ All stacks should define the same EBPR range
    - ✓ Coordination between sysplexports and EBPR
    - ✓ Ports within the EBPR range reserved on a stack via PORT or PORTRANGE statement will be excluded from the EBPR port pool
  - Sysplex-wide ephemeral ports are allocated from this pool when applications BIND explicitly to INADDR_ANY or IN6ADDR_ANY and port 0
    - ✓ The EZBEPORTvvtt structure in the Coupling Facility will be used to coordinate port allocation from this range across the sysplex
    - ✓ If the application subsequently issues a LISTEN() on that socket, the port is not used, and will be returned to the pool
  - These ports can be used with any source IP address selected at connect time
  - If the active EBPR in the sysplex is changed, message EZD1291I is issued by the stack which caused the change
  - In a common INET (CINET) environment, EBPR may be defined on a stack, but the Explicit Bind Port Range function is supported only in a limited set of CINET configurations.
    - ✓ It is supported when CINET is managing only one stack on the system, or when stack affinity has been established.
    - ✓ If GLOBALCONFIG EXPLICITBINDPORTRANGE is specified in a CINET environment, this message will be issued:

```
EZZ0797I EXPLICITBINDPORTRANGE HAS LIMITED SUPPORT IN A CINET ENVIRONMENT
```

- ➢ SRCIP JOBNAME support for listeners
  - The SRCIP JOBNAME rule can be used by a Listening server when it Binds to the unspecified address
  - New parameters added to the SRCIP JOBNAME rule
    - ✓ SERVER – Provides the specific IP address for a listener
    - ✓ CLIENT – Provides the existing support for an outbound connect
      - – This is the default
    - ✓ BOTH – Allows the same JOBNAME rule to be applied for both client and server applications
  - Restriction: SERVER and BOTH are not allowed on JOBNAME * rules

37

Redbooks ibm.com/redbooks

Support for DDVIPAs on the SRCIP DEST rule is allowed by a new sysplex-wide port range called the Explicit Bind Port Range. It is configured with a new parameter on the TCP/IP GLOBALCONFIG statement. There are two new GLOBABLCONFIG parameters for enabling and disabling an Explicit Bind Port Range processing. NOEXPLICITBINDPORTRANGE indicates that the stack will not participate in EBPR processing, when handling an explicit bind() of a TCP socket to an IP address of INADDR_ANY or IN6ADDR_ANY and port 0. EXPLICITBINDPORTRANGE indicates that the stack will participate in EBPR processing, when processing an explicit bind() of a TCP socket to an IP address of INADDR_ANY or IN6ADDR_ANY and port 0. It also specifies the range of ports, starting at *1st_port* for *num_ports* ports, that will define that pool. This parameter will define the range used by all stacks participating in EBPR processing throughout the sysplex. The EZBEPORTvvtt structure will coordinate port allocation across the sysplex in the new range, using list 0 of the structure. The ports are allocated from the structure in blocks of 64 ports at a time, as with Sysplexports. There is also coordination between sysplexports and the explicit bind port range. DVIPA-specific sysplexports will not be allocated from the EBPR range. Furthermore DVIPA-specific sysplexports already in use will not be assigned from the EBPR range.

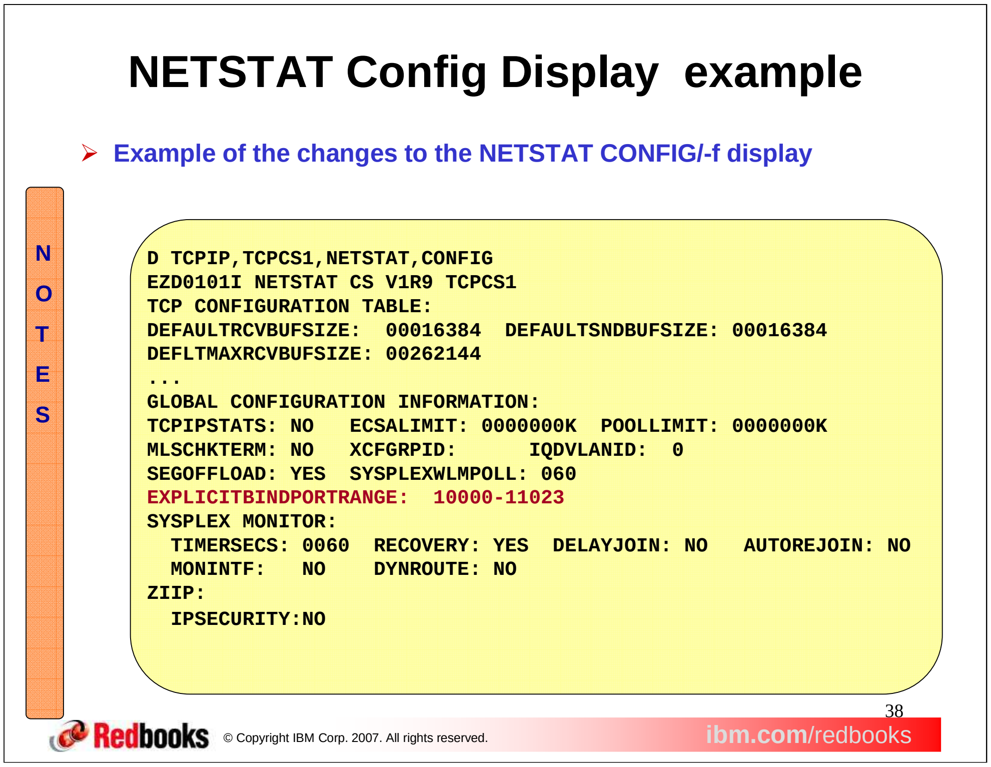
All TCP/IP stacks in the sysplex that will participate in EXPLICITBINDPORTRANGE processing should specify the same port range. This can be done by specifying the GLOBALCONFIG EXPLICITPORTRANGE statement in a file that is specified in an INCLUDE statement in the TCP profiles data sets of all the participating stacks. If stacks define different ranges, the last configuration processed defines the EBPR range for the entire sysplex (or subplex). The port range defined on the EXPLICITBINDPORTRANGE parameter should not overlap any existing port reservations of any TCP/IP stacks in the sysplex. Note that any reserved ports that are within the EXPLICITBINDPORTRANGE range will be excluded from the EXPLICITBINDPORTRANGE port pool, effectively making the pool smaller. The EXPLICITBINDPORTRANGE port range must be large enough to accommodate all applications in the sysplex that may issue explicit bind() calls for INADDR_ANY (or IN6ADDR_ANY) with port 0. If additional TCP/IP stacks or systems are introduced into the sysplex, the extent of the port range defined by EXPLICITBINDPORTRANGE should be re-evaluated. If the size of the port range defined by EXPLICITBINDPORTRANGE is too large, there will be fewer ports available for local ephemeral port allocation.

Changing the range specified on the EXPLICITBINDPORTRANGE parameter of the GLOBALCONFIG statement affects every stack in the sysplex that has configured a GLOBALCONFIG EXPLICITBINDPORTRANGE. Future port allocations for all such stacks will use the new port range. Ports in the EXPLICITBINDPORTRANGE range are usually assigned to a stack in blocks of 64 ports. When expanding the range, you should use multiples of 64 times the number of stacks using GLOBALCONFIG EXPLICITBINDPORTRANGE.

A stack with an Explicit Bind Port Range configured will attempt to set the range in the EZBEPORTvvtt structure. If the range in the structure is successfully changed to a different range, the stack setting the range will issue message EZD1291I to display the new range. If the attempt to set the range did not change the range, no message will be issued.

When operating in a CINET environment where CINET is managing more than one stack and stack affinity has not been established, CINET will substitute a port from the INADDRANYPORT port range defined in the BPXPRMxx parmlib when an application Binds to INADDR_ANY or IN6ADDR_ANY and port 0, before passing the BIND() request to the TCP/IP stack. The TCP/IP stack will not see a Bind to port 0, but will instead see a BIND specifying a specific port. Therefore, it will not assign a port from the ExplicitBindPortRange pool. Subsequently, if a SRCIP DESTINATION match selects a distributed DVIPA to be the source IP address, the connect will fail with the JRSRCIPDistDVIPA reason code, indicating the port is not from the EBPR pool.

The SRCIP JOBNAME rule is extended to allow users to specify a specific address for a Server application that Binds to INADDR(6)_ANY. Three new parameters may be specified on the JOBNAME rule. The SERVER parameter will cause a Server application with a matching jobname to have the source IP address on the matching rule used in place of INADDR(6)_ANY on the Listen(). The CLIENT parameter (which is the default), will indicate that existing behavior of substituting the source IP address for outbound connections done by an application matching the specified JOBNAME (Clients). The BOTH parameter will allow the same JOBNAME rule to be used for both client and server applications. The SERVER and BOTH parameters are not allowed on JOBNAME * rules.



NETSTAT Config Display example

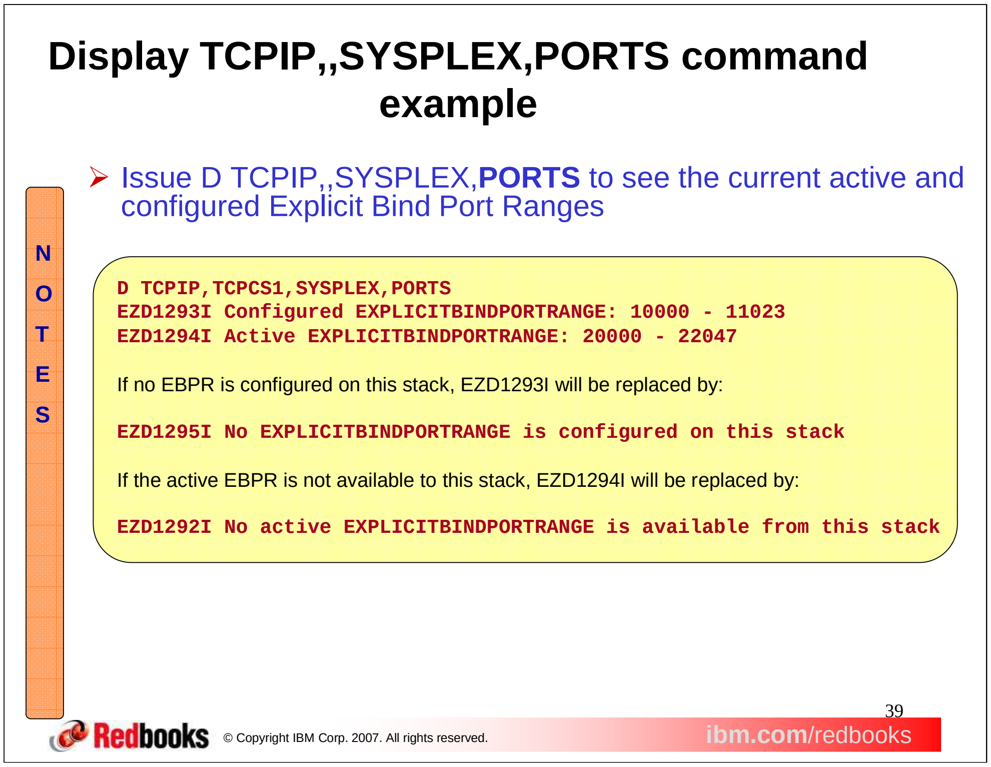
- Example of the changes to the NETSTAT CONFIG/-f display

```
D TCPIP,TCPCS1,NETSTAT,CONFIG
EZD0101I NETSTAT CS V1R9 TCPCS1
TCP CONFIGURATION TABLE:
DEFAULTRCVBUFSIZE:  00016384  DEFAULTSNDBUFSIZE: 00016384
DEFLTMAXRCVBUFSIZE: 00262144
...
GLOBAL CONFIGURATION INFORMATION:
TCPIPSTATS: NO   ECSALIMIT: 0000000K  POOLLIMIT: 0000000K
MLSCHKTERM: NO   XCFGRPID:      IQDVLANID:  0
SEGOFFLOAD: YES  SYSPLEXWLMPOLL: 060
EXPLICITBINDPORTRANGE:  10000-11023
SYSPLEX MONITOR:
  TIMERSECS: 0060  RECOVERY: YES  DELAYJOIN: NO   AUTOREJOIN: NO
  MONINTF:   NO    DYNROUTE: NO
ZIIP:
  IPSECURITY:NO
```

38

ibm.com/redbooks

The NETSTAT CONFIG/-f command will display the Explicit Bind Port Range under the Global Configuration Information section. If no Explicit Bind Port Range is configured on this stack, 00000-00000 will be displayed as the range.



Display TCPIP,,SYSPLEX,PORTS command example

- Issue D TCPIP,,SYSPLEX,**PORTS** to see the current active and configured Explicit Bind Port Ranges

```
D TCPIP,TCPCS1,SYSPLEX,PORTS
EZD1293I Configured EXPLICITBINDPORTRANGE: 10000 - 11023
EZD1294I Active EXPLICITBINDPORTRANGE: 20000 - 22047
```

If no EBPR is configured on this stack, EZD1293I will be replaced by:

```
EZD1295I No EXPLICITBINDPORTRANGE is configured on this stack
```

If the active EBPR is not available to this stack, EZD1294I will be replaced by:

```
EZD1292I No active EXPLICITBINDPORTRANGE is available from this stack
```

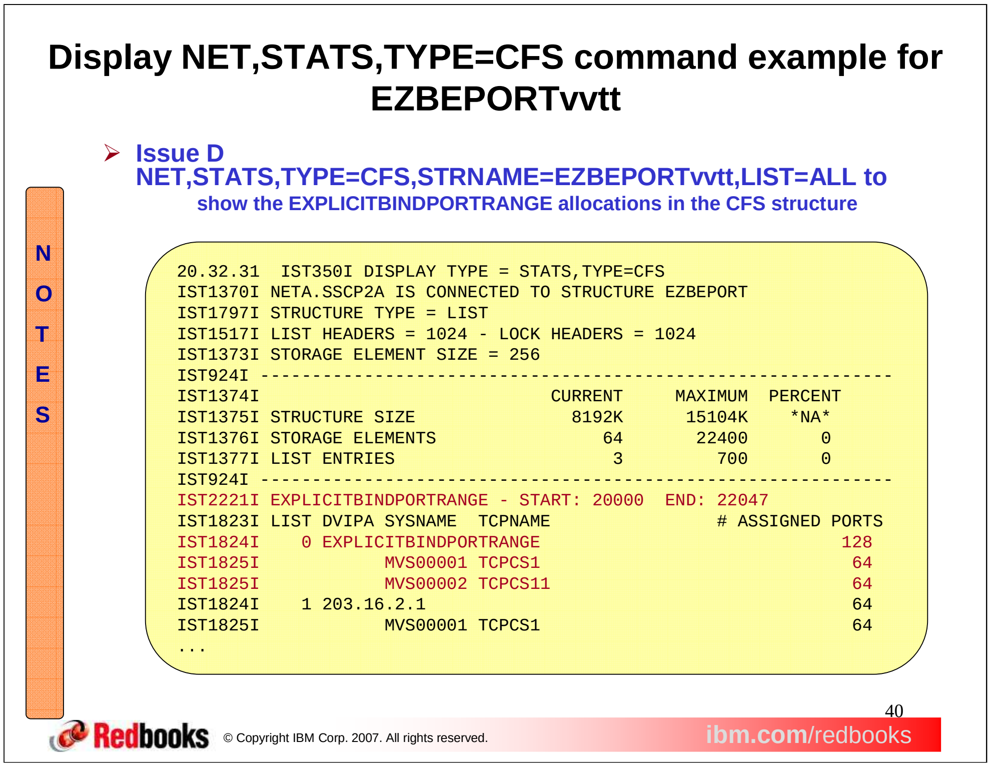
39

Redbooks

ibm.com/redbooks

The current active and configured EBPR ranges can be displayed using the new D TCPIP,,SYSPLEX,PORTS command. The configured range is the range that was defined on the specified stack. The active range is the range that is actually being used in the EZBEPORTvvtt structure.

Message EZD1292I may be issued if the stack has not yet fully completed establishing an Explicit Bind Port Range with the Coupling Facility or if access to the Coupling Facility structure has failed.

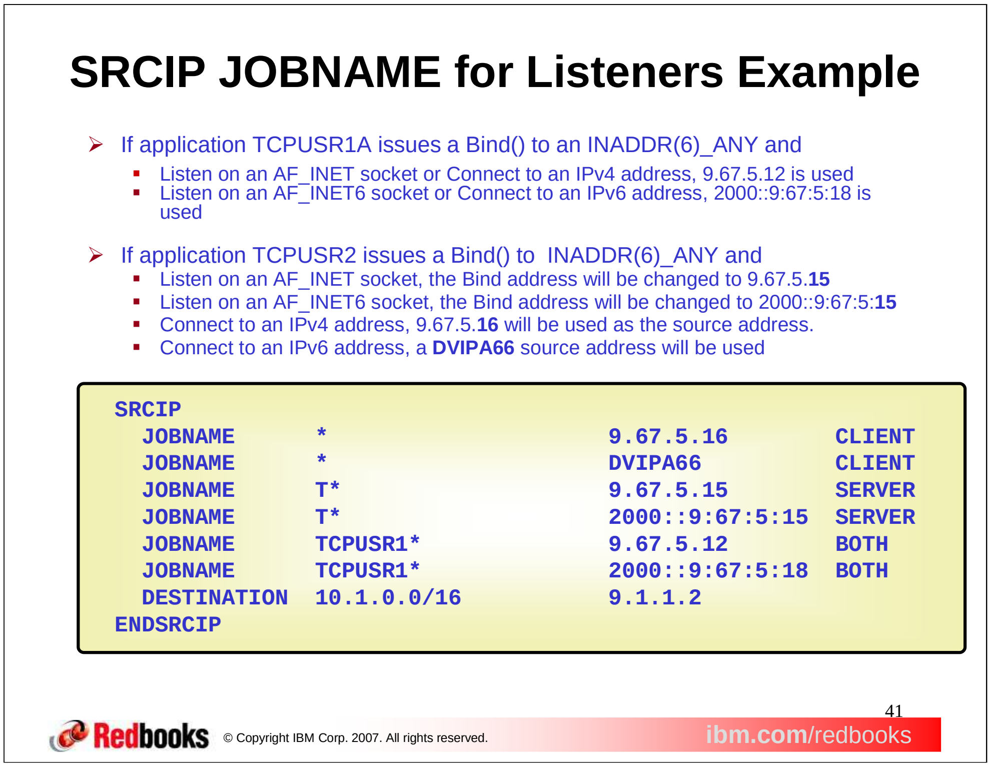


Display NET,STATS,TYPE=CFS command example for EZBEPORTvvtt

- Issue D NET,STATS,TYPE=CFS,STRNAME=EZBEPORTvvtt,LIST=ALL to show the EXPLICITBINDPORTRANGE allocations in the CFS structure

```
20.32.31  IST350I DISPLAY TYPE = STATS,TYPE=CFS
IST1370I NETA.SSCP2A IS CONNECTED TO STRUCTURE EZBEPORT
IST1797I STRUCTURE TYPE = LIST
IST1517I LIST HEADERS = 1024 - LOCK HEADERS = 1024
IST1373I STORAGE ELEMENT SIZE = 256
IST924I -------------------------------------------------------------
IST1374I                            CURRENT     MAXIMUM  PERCENT
IST1375I STRUCTURE SIZE               8192K      15104K    *NA*
IST1376I STORAGE ELEMENTS                64       22400       0
IST1377I LIST ENTRIES                     3         700       0
IST924I -------------------------------------------------------------
IST2221I EXPLICITBINDPORTRANGE - START: 20000  END: 22047
IST1823I LIST DVIPA SYSNAME  TCPNAME                # ASSIGNED PORTS
IST1824I    0 EXPLICITBINDPORTRANGE                             128
IST1825I            MVS00001 TCPCS1                              64
IST1825I            MVS00002 TCPCS11                             64
IST1824I    1 203.16.2.1                                         64
IST1825I            MVS00001 TCPCS1                              64
...
```

The active Explicit Bind Port Range can also be display from VTAM, by displaying the structure information for the EZBEPORTvvtt structure. If there is no active EBPR range, message IST2221I will not be displayed, and no information on list 0 will be displayed.



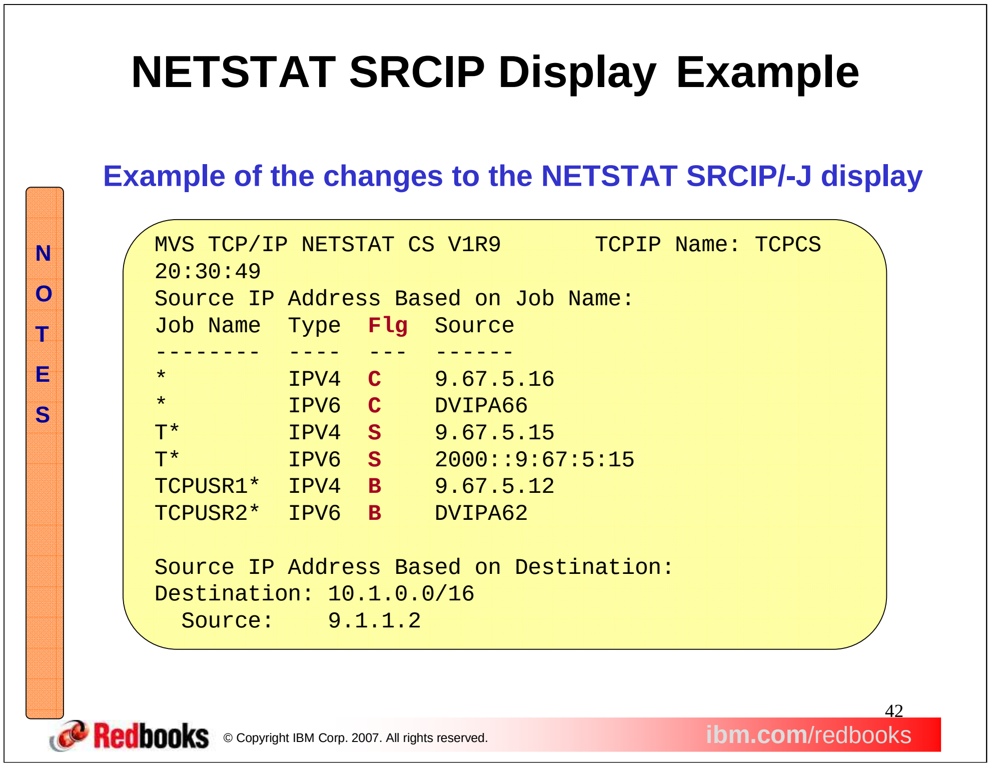
SRCIP JOBNAME for Listeners Example

- If application TCPUSR1A issues a Bind() to an INADDR(6)_ANY and
  - Listen on an AF_INET socket or Connect to an IPv4 address, 9.67.5.12 is used
  - Listen on an AF_INET6 socket or Connect to an IPv6 address, 2000::9:67:5:18 is used
- If application TCPUSR2 issues a Bind() to INADDR(6)_ANY and
  - Listen on an AF_INET socket, the Bind address will be changed to 9.67.5.**15**
  - Listen on an AF_INET6 socket, the Bind address will be changed to 2000::9:67:5:**15**
  - Connect to an IPv4 address, 9.67.5.**16** will be used as the source address.
  - Connect to an IPv6 address, a **DVIPA66** source address will be used

```
SRCIP
  JOBNAME      *                     9.67.5.16        CLIENT
  JOBNAME      *                     DVIPA66          CLIENT
  JOBNAME      T*                    9.67.5.15        SERVER
  JOBNAME      T*                    2000::9:67:5:15  SERVER
  JOBNAME      TCPUSR1*              9.67.5.12        BOTH
  JOBNAME      TCPUSR1*              2000::9:67:5:18  BOTH
  DESTINATION  10.1.0.0/16           9.1.1.2
ENDSRCIP
```

This slide illustrates the use of the new SRCIP JOBNAME rule parameters.

Note that if the application issues a Bind to INADDR_ANY on an AF_INET6 socket, an IPv4 source address will be chosen over an IPv6 address since the application is intending to use the socket to receive mapped IPv4 packets. So in the example above, if application TCPUSR2 issues a Bind to INADDR_ANY & a Listen on an AF_INET6 socket, the Bind address would be changed to 9.67.5.15.



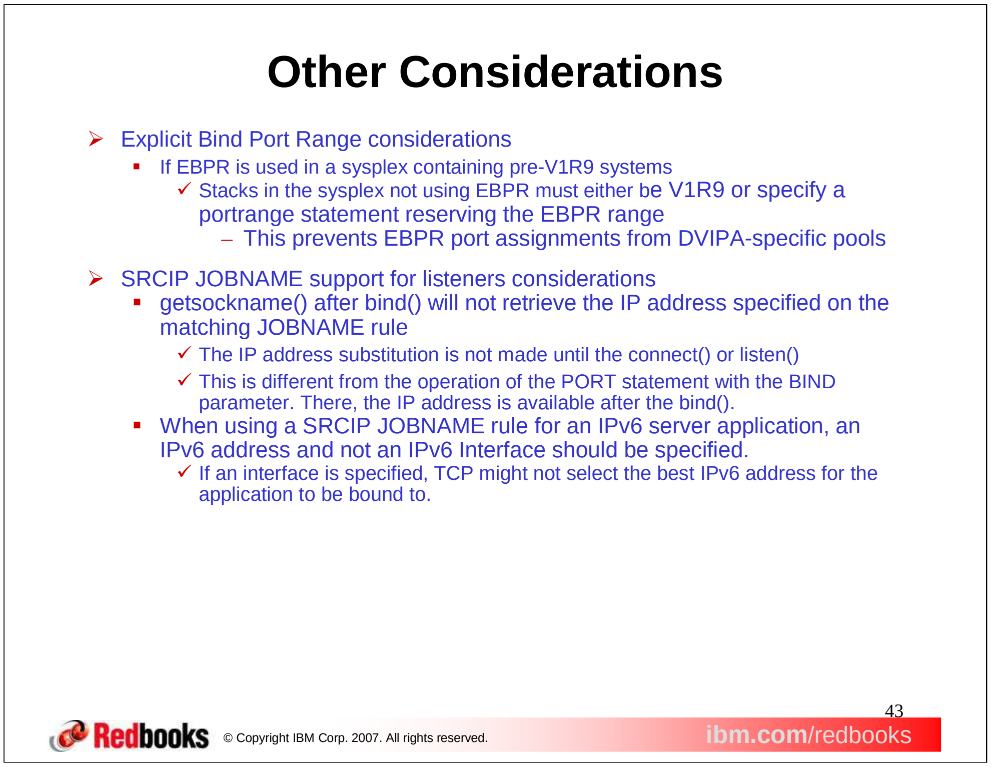
NETSTAT SRCIP Display Example

N O T E S

Example of the changes to the NETSTAT SRCIP/-J display

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS
20:30:49
Source IP Address Based on Job Name:
Job Name  Type  Flg  Source
--------  ----  ---  ------
*         IPV4  C    9.67.5.16
*         IPV6  C    DVIPA66
T*        IPV4  S    9.67.5.15
T*        IPV6  S    2000::9:67:5:15
TCPUSR1*  IPV4  B    9.67.5.12
TCPUSR2*  IPV6  B    DVIPA62

Source IP Address Based on Destination:
Destination: 10.1.0.0/16
  Source:    9.1.1.2
```

The NETSTAT SRCIP/-J display shows the new JOBNAME parameters by adding a flag column, FLG. In that column, C is for CLIENT, S is for SERVER, and B is for BOTH.



Other Considerations

- ➢ Explicit Bind Port Range considerations
  - ▪ If EBPR is used in a sysplex containing pre-V1R9 systems
    - ✓ Stacks in the sysplex not using EBPR must either be V1R9 or specify a portrange statement reserving the EBPR range
      - – This prevents EBPR port assignments from DVIPA-specific pools
- ➢ SRCIP JOBNAME support for listeners considerations
  - ▪ getsockname() after bind() will not retrieve the IP address specified on the matching JOBNAME rule
    - ✓ The IP address substitution is not made until the connect() or listen()
    - ✓ This is different from the operation of the PORT statement with the BIND parameter. There, the IP address is available after the bind().
  - ▪ When using a SRCIP JOBNAME rule for an IPv6 server application, an IPv6 address and not an IPv6 Interface should be specified.
    - ✓ If an interface is specified, TCP might not select the best IPv6 address for the application to be bound to.

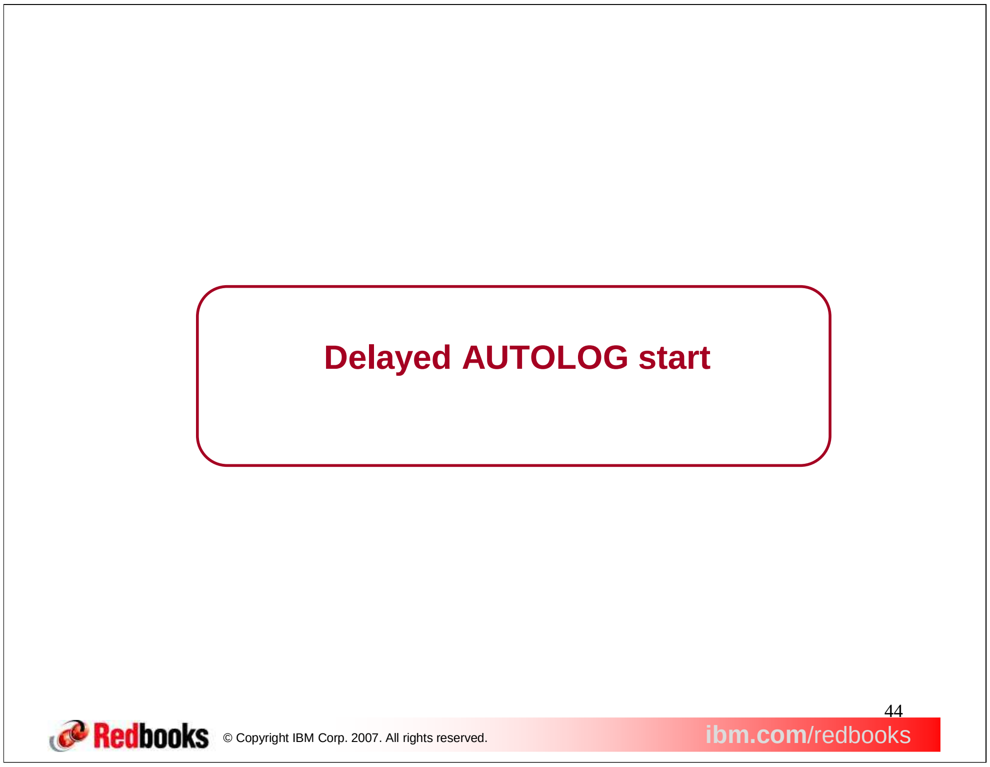
Redbooks ibm.com/redbooks

This slide describes how to handle mixed configurations, that include V1R9 systems and pre-V1R9 systems when using the explicit bind port range function. Also, in a CINET environment if there is no stack affinity and multiple stacks, CINET will assign a port before the stack even receives the Bind. Therefore the explicit bind port range processing will not occur for the request.

Applications that bind to INADDR(6)_ANY, and match on a SRCIP JOBNAME or DESTINATION statement, will not have the designated IP address as its source address upon completion of the bind() call. The source address will not be set to the designated address until completion of the subsequent connect() (client applications) or listen() (server applications) call.

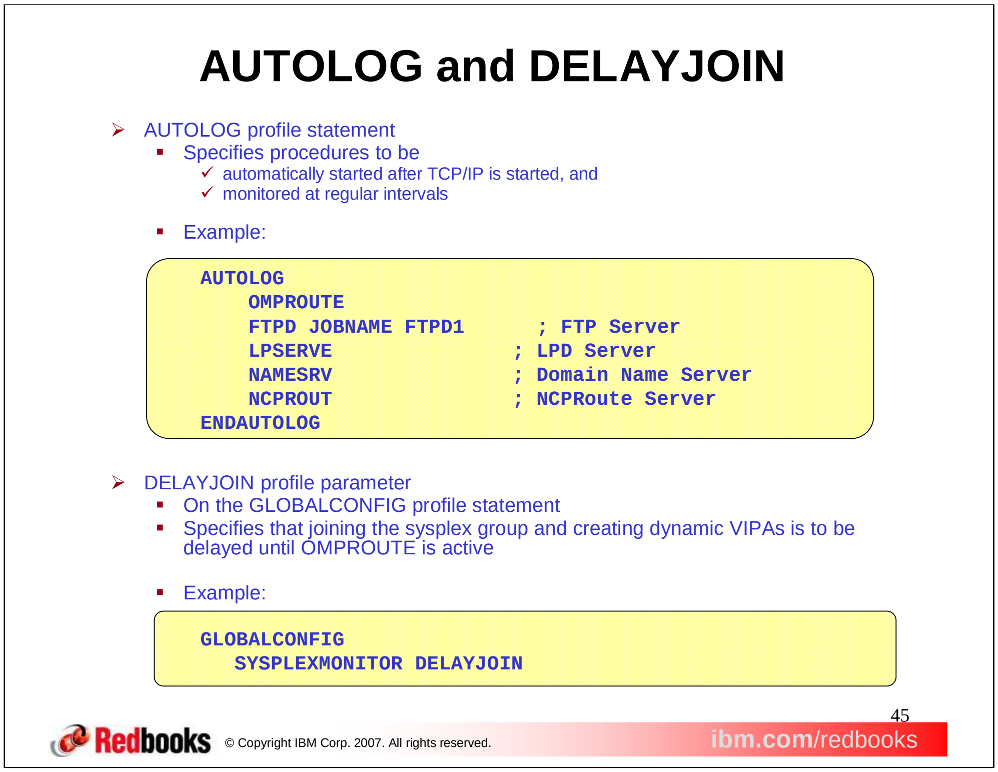
When using the BIND parameter on the PORT statement, the designated IP address will be set upon completion of the bind() call.

If an IPv6 server application matches a JOBNAME rule specifying an IPv6 Interface, rather than an IPv6 address, TCP will choose the first IPv6 address in the interface as the Bind() address. This may not be the most appropriate IP address for the inbound connections.



Delayed AUTOLOG start

The enhancement for AUTOLOG is concerned with the timing of when AUTOLOG starts a procedure.



AUTOLOG and DELAYJOIN

- AUTOLOG profile statement
  - Specifies procedures to be
    - ✓ automatically started after TCP/IP is started, and
    - ✓ monitored at regular intervals
  - Example:

```
AUTOLOG
    OMPROUTE
    FTPD JOBNAME FTPD1       ; FTP Server
    LPSERVE                ; LPD Server
    NAMESRV                ; Domain Name Server
    NCPROUT                ; NCPRoute Server
ENDAUTOLOG
```

- DELAYJOIN profile parameter
  - On the GLOBALCONFIG profile statement
  - Specifies that joining the sysplex group and creating dynamic VIPAs is to be delayed until OMPROUTE is active
  - Example:

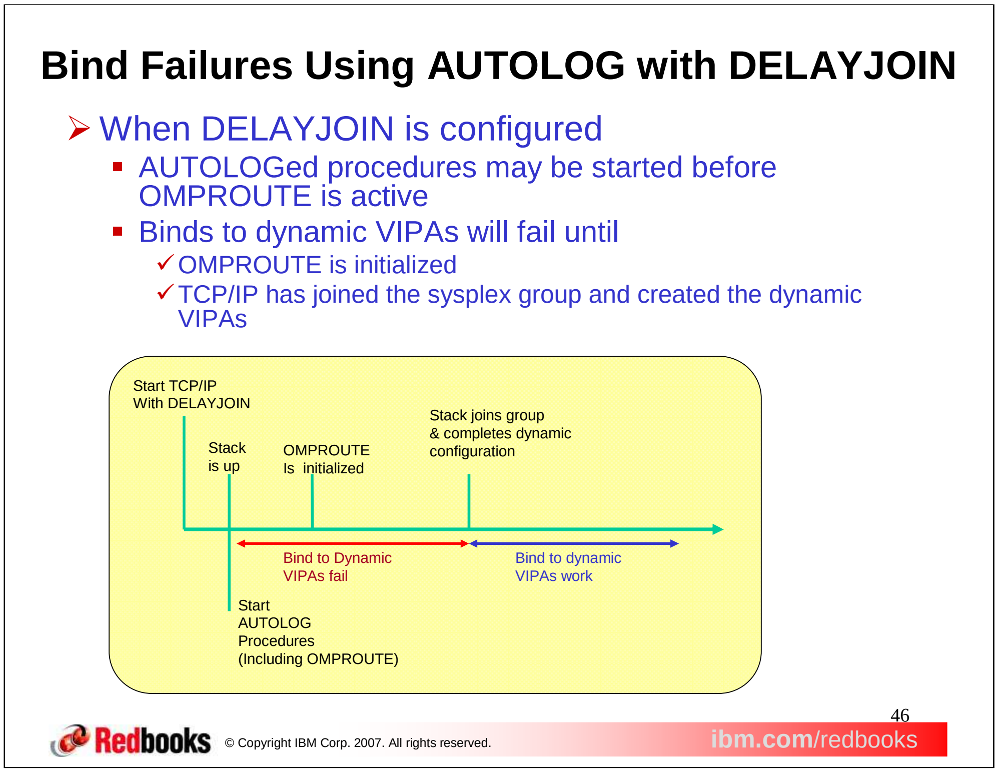
```
GLOBALCONFIG
   SYSPLEXMONITOR DELAYJOIN
```

45

ibm.com/redbooks

The procedures to be automatically started as soon as TCP/IP has been initialized are specified on the AUTOLOG profile statement. The example shown here will cause AUTOLOG to start the five specified procedures as soon as TCP/IP initialization is completed.

DELAYJOIN is another configuration parameter.   It is specified on the GLOBALCONFIG profile statement, as shown in the example on this slide.   When DELAYJOIN is specified, TCP/IP will not join the sysplex group until OMPROUTE is active . Since the stack’s dynamic VIPA configuration is not processed until after the stack has joined the sysplex group, this delay in joining the sysplex group ensures that OMPROUTE will be active and ready to advertise dynamic VIPAs when they are created on this stack.



Bind Failures Using AUTOLOG with DELAYJOIN

- When DELAYJOIN is configured
  - AUTOLOGed procedures may be started before OMPROUTE is active
  - Binds to dynamic VIPAs will fail until
    - ✓ OMPROUTE is initialized
    - ✓ TCP/IP has joined the sysplex group and created the dynamic VIPAs

Start TCP/IP
With DELAYJOIN

Stack
is up

OMPROUTE
Is initialized

Stack joins group
& completes dynamic
configuration

Bind to Dynamic
VIPAs fail

Bind to dynamic
VIPAs work

Start
AUTOLOG
Procedures
(Including OMPROUTE)

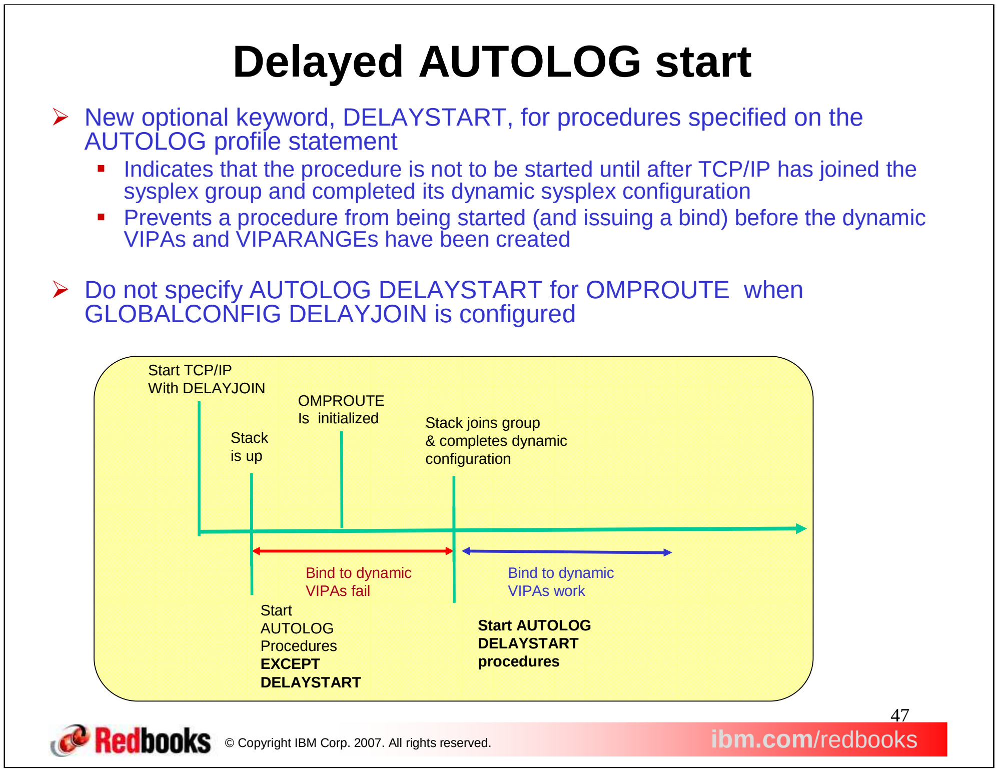
46

When DELAYJOIN is configured, the stack will not join the sysplex group and create dynamic VIPAs until OMPROUTE signals that it is ready to advertise them. OMPROUTE and the other AUTOLOGed procedures will be started at the same time. AUTOLOGed procedures that bind to dynamic VIPAs may fail due to the delay in creating these DVIPAs.

This slide contains a time line showing what happens to bind requests to dynamic VIPAs when TCP/IP is started with DELAYJOIN specified.

When the TCP/IP stack is started with DELAYJOIN configured,

1. the stack completes its basic initialization (the stack is up) but it is not in the sysplex group yet.
2. At this point, AUTOLOG will start the specified procedures. In our example, this includes OMPROUTE.
3. When OMPROUTE has completed its initialization and is active, it notifies the stack.
4. Now the stack will join the sysplex group and then process its dynamic configuration, which includes creating its dynamic VIPAs. Until the stack has completed its dynamic configuration processing, any bind request to a dynamic VIPA will fail.



Delayed AUTOLOG start

- New optional keyword, DELAYSTART, for procedures specified on the AUTOLOG profile statement
  - Indicates that the procedure is not to be started until after TCP/IP has joined the sysplex group and completed its dynamic sysplex configuration
  - Prevents a procedure from being started (and issuing a bind) before the dynamic VIPAs and VIPARANGEs have been created
- Do not specify AUTOLOG DELAYSTART for OMPROUTE when GLOBALCONFIG DELAYJOIN is configured

Start TCP/IP
With DELAYJOIN

Stack
is up

OMPROUTE
Is initialized

Stack joins group
& completes dynamic
configuration

Bind to dynamic
VIPAs fail

Bind to dynamic
VIPAs work

Start
AUTOLOG
Procedures
EXCEPT
DELAYSTART

Start AUTOLOG
DELAYSTART
procedures

47

Redbooks

ibm.com/redbooks

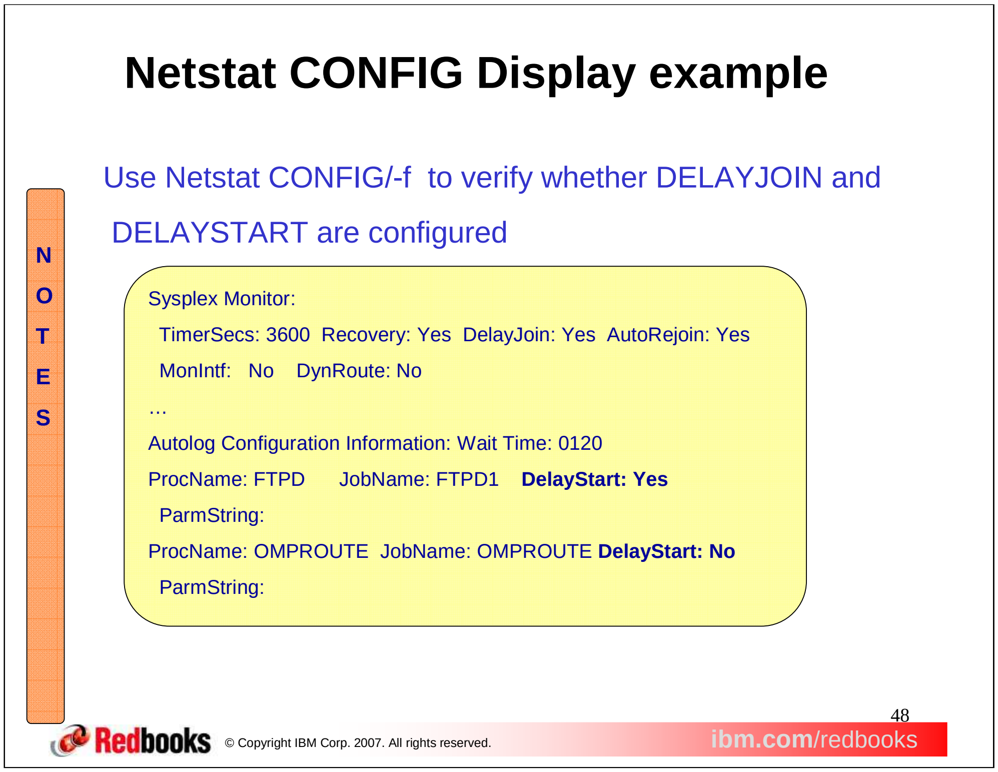
V1R9 provides a new optional keyword, DELAYSTART, for procedures specified on the AUTOLOG profile statement. DELAYSTART is used to identify procedures that should not be automatically started until after the stack has joined the sysplex group and its dynamic sysplex configuration is completed. At that point, bind requests to dynamic VIPAs can succeed.

One word of caution: when DELAYJOIN is configured, do not specify DELAYSTART for your OMPROUTE procedure. If you do, the stack will complete its initialization but OMPROUTE will never be started (because AUTOLOG is waiting for the stack to join the sysplex group) and the stack will not join the group and create its dynamic VIPAs because the stack is waiting for OMPROUTE to be active.

The slide contains a time line showing AUTOLOG processing when DELAYSTART is specified for some procedures and not specified for other procedures.

As we saw earlier, when the TCP/IP stack is started with DELAYJOIN configured,

1. the stack completes its basic initialization (the stack is up) but it is not in the sysplex group yet.
2. At this point, AUTOLOG will start the specified procedures. However, it will ignore procedures that have DELAYSTART specified.
3. When OMPROUTE has completed its initialization and is active, it notifies the stack.
4. Now the stack will join the sysplex group and then process its dynamic configuration, which includes creating its dynamic VIPAs. Until the stack has completed its dynamic configuration processing, any bind request to a dynamic VIPA will fail.
5. If DELAYSTART has been configured for any AUTOLOGed procedure, the stack notifes AUTOLOG when the dynamic configuration is complete and AUTOLOG then starts all procedures with DELAYSTART specified.



Netstat CONFIG Display example

NOTES

Use Netstat CONFIG/-f to verify whether DELAYJOIN and DELAYSTART are configured

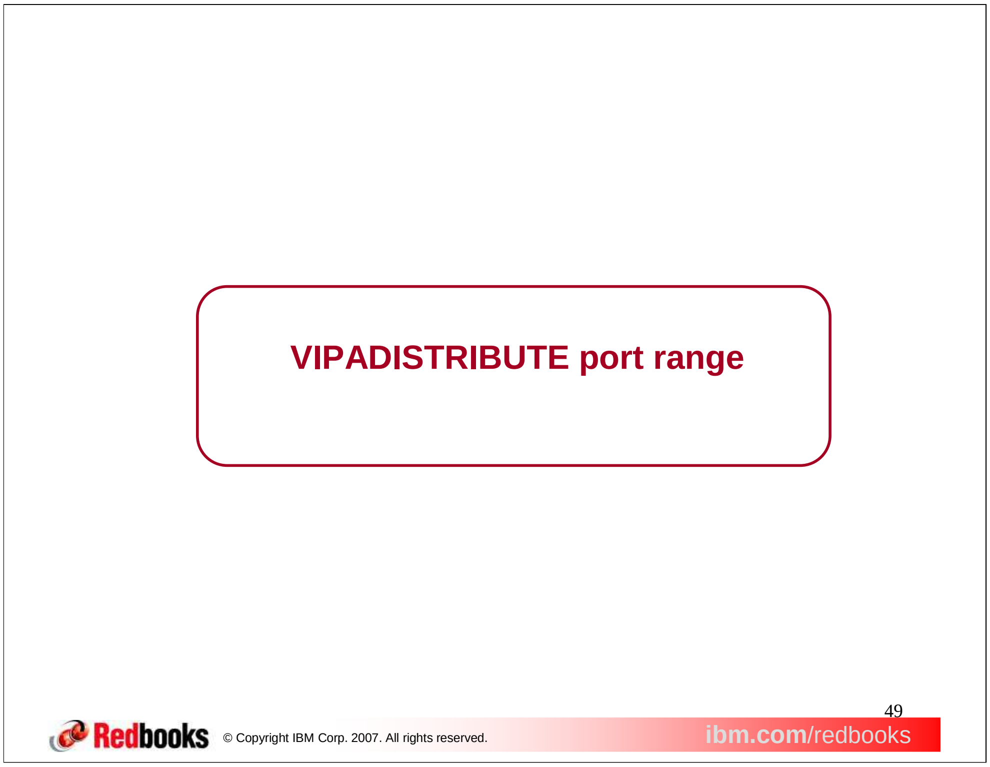
```
Sysplex Monitor:
  TimerSecs: 3600  Recovery: Yes  DelayJoin: Yes  AutoRejoin: Yes
  MonIntf:   No    DynRoute: No
…
Autolog Configuration Information: Wait Time: 0120
ProcName: FTPD      JobName: FTPD1    DelayStart: Yes
  ParmString:
ProcName: OMPROUTE  JobName: OMPROUTE DelayStart: No
  ParmString:
```

48

Redbooks

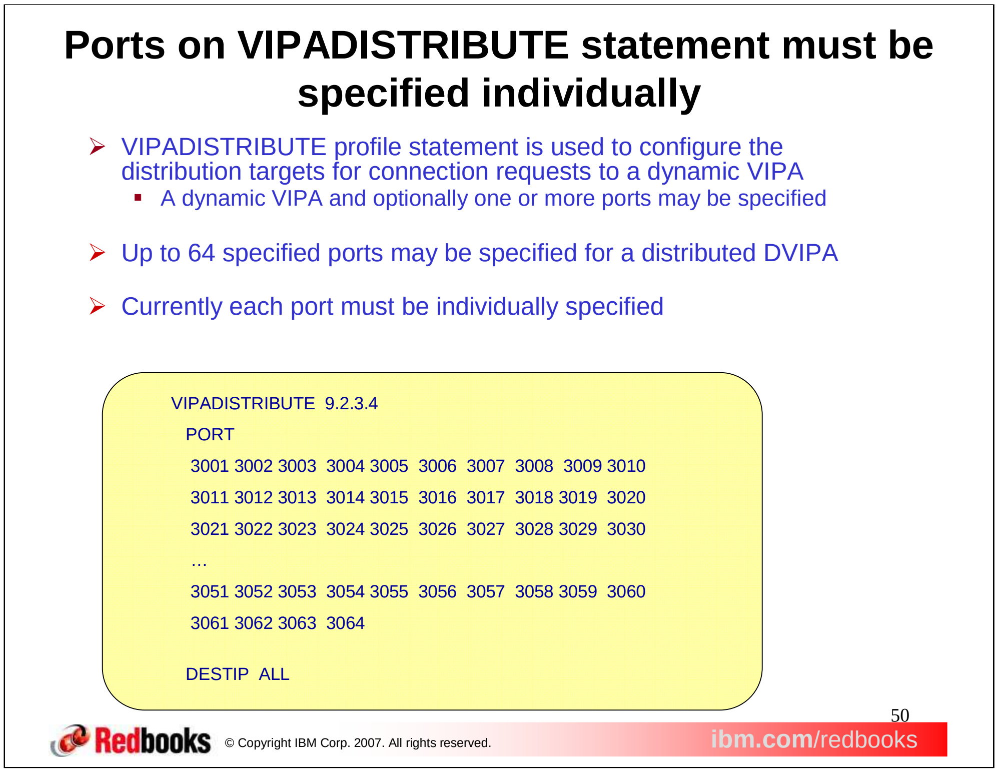
ibm.com/redbooks

The Netstat CONFIG display command can be used to verify the new AUTOLOG DELAYSTART values which are shown in bold type in this example.   The same Netstat command also displays the Sysplex Monitor DELAYJOIN value.



VIPADISTRIBUTE port range

With z/OS V1R9, the VIPADISTRIBUTE profile statement is enhanced to allow you to specify a range of ports.



Ports on VIPADISTRIBUTE statement must be specified individually

- VIPADISTRIBUTE profile statement is used to configure the distribution targets for connection requests to a dynamic VIPA
  - A dynamic VIPA and optionally one or more ports may be specified
- Up to 64 specified ports may be specified for a distributed DVIPA
- Currently each port must be individually specified

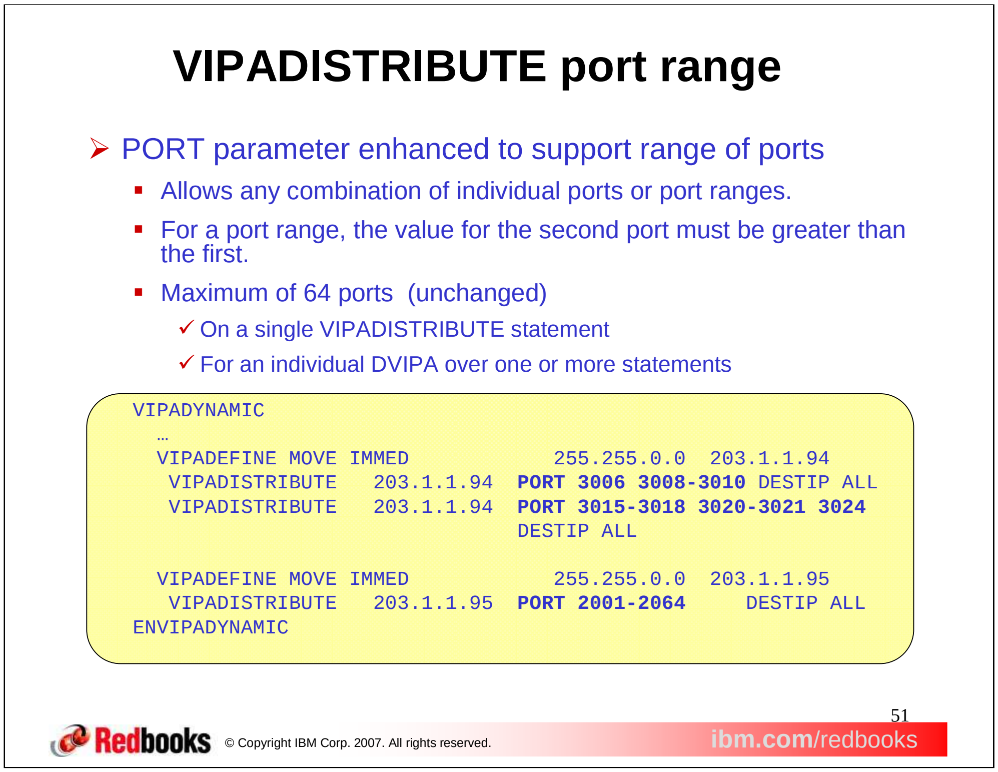
```
VIPADISTRIBUTE  9.2.3.4
  PORT
   3001 3002 3003  3004 3005  3006  3007  3008  3009 3010
   3011 3012 3013  3014 3015  3016  3017  3018 3019  3020
   3021 3022 3023  3024 3025  3026  3027  3028 3029  3030
   …
   3051 3052 3053  3054 3055  3056  3057  3058 3059  3060
   3061 3062 3063  3064

  DESTIP  ALL
```

Redbooks

ibm.com/redbooks

The VIPADISTRIBUTE profile statement specifies how inbound connection requests to a dynamic VIPA are to be distributed among the stacks in the sysplex group. The example on this slide specifies that new connection requests can be forwarded to any stack in the sysplex group which has a socket listening on the dynamic VIPA 9.2.3.4 or inaddr_any, and one of the ports (3000, 3001, 3002) .

On a VIPADISTRIBUTE statement, you can specify up to 64 ports, each of which must be individually specified. The example on this slide shows a VIPADISTRIBUTE statement that specifies ports 3001 through 3064.



VIPADISTRIBUTE port range

- PORT parameter enhanced to support range of ports
  - Allows any combination of individual ports or port ranges.
  - For a port range, the value for the second port must be greater than the first.
  - Maximum of 64 ports (unchanged)
    - ✓ On a single VIPADISTRIBUTE statement
    - ✓ For an individual DVIPA over one or more statements

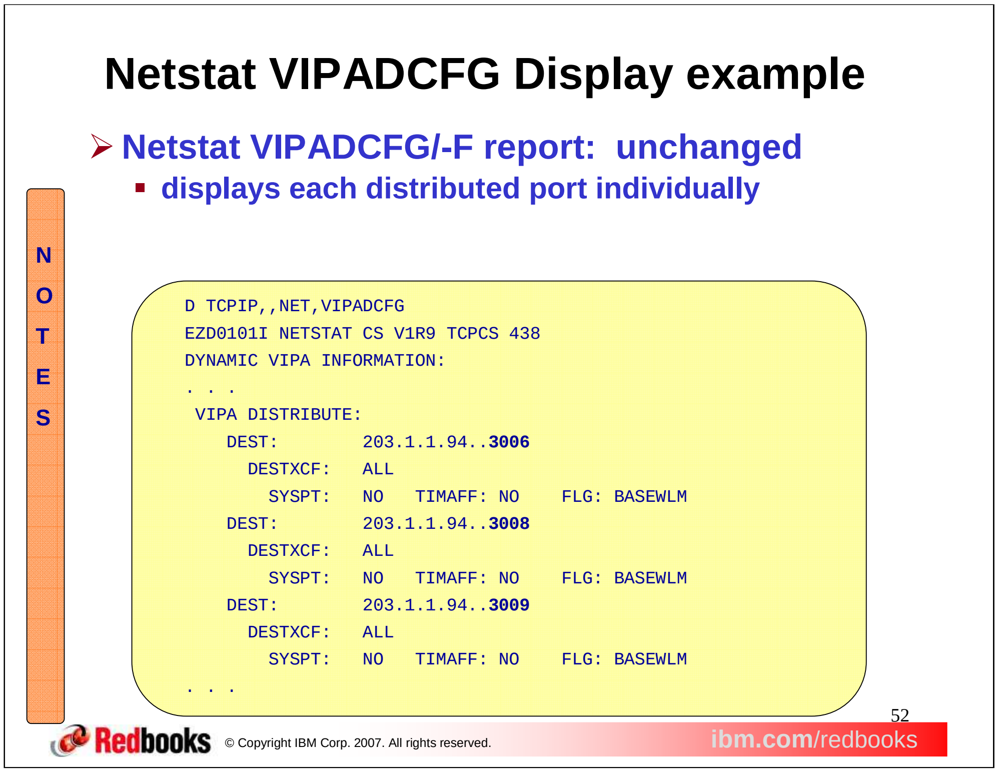
```
VIPADYNAMIC
  …
  VIPADEFINE MOVE IMMED            255.255.0.0  203.1.1.94
   VIPADISTRIBUTE   203.1.1.94  PORT 3006 3008-3010 DESTIP ALL
   VIPADISTRIBUTE   203.1.1.94  PORT 3015-3018 3020-3021 3024
                                DESTIP ALL

  VIPADEFINE MOVE IMMED            255.255.0.0  203.1.1.95
   VIPADISTRIBUTE   203.1.1.95  PORT 2001-2064     DESTIP ALL
ENVIPADYNAMIC
```

With z/OS V1R9, the VIPADISTRIBUTE statement syntax is more flexible. It will now accept individual port numbers, a range of port numbers, or a combination of individual ports and ranges for the PORT keyword. The maximum number of ports that can be specified on a VIPADISTRIBUTE statement or for a DVIPA over multiple statements remains 64.

The example on this slide contains a VIPADYNAMIC block containing several VIPADISTRIBUTE statements that demonstrate the new syntax for the PORT values.

Connections to DVIPA 203.1.1.94 are to be distributed for ports 3006, 3008, 3009, and 3010 (because of the first VIPADISTRIBUTE statement) and ports 3015, 3016, 3017, 3018, 3020, 3021, and 3024 (because of the second VIPADISTRIBUTE statement).

Connections to DVIPA 203.1.1.95 are to be distributed for the full range of 64 ports from 2001 through 2064



Netstat VIPADCFG Display example

- Netstat VIPADCFG/-F report: unchanged
  - displays each distributed port individually

NOTES

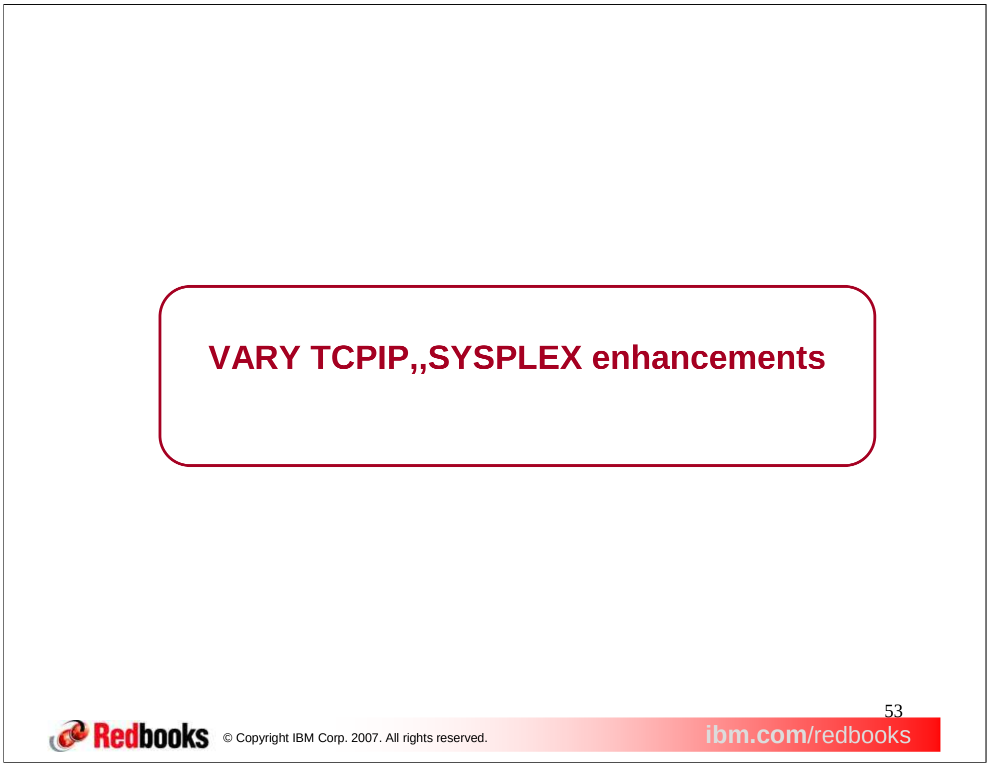
```
D TCPIP,,NET,VIPADCFG
EZD0101I NETSTAT CS V1R9 TCPCS 438
DYNAMIC VIPA INFORMATION:
. . .
 VIPA DISTRIBUTE:
    DEST:        203.1.1.94..3006
      DESTXCF:   ALL
        SYSPT:   NO   TIMAFF: NO    FLG: BASEWLM
    DEST:        203.1.1.94..3008
      DESTXCF:   ALL
        SYSPT:   NO   TIMAFF: NO    FLG: BASEWLM
    DEST:        203.1.1.94..3009
      DESTXCF:   ALL
        SYSPT:   NO   TIMAFF: NO    FLG: BASEWLM
. . .
```

52

Redbooks

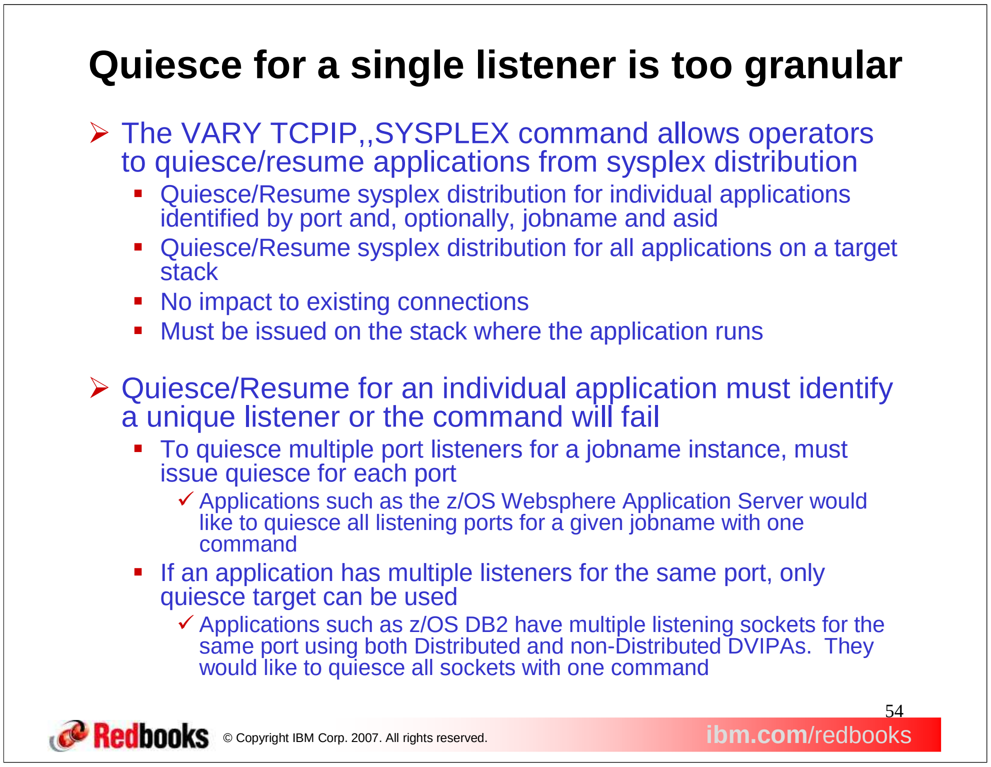
ibm.com/redbooks

The Netstat VIPADCFG command displays the distribution specifications that are configured for dynamic VIPAs on this stack. Although you can now specify ranges of port numbers on the VIPADISTRIBUTE statement, this Netstat report has not changed. As in previous releases, it displays distribution specifications for each individual DVIPA-port combination.



VARY TCPIP,,SYSPLEX enhancements

The next slides describe an enhancement to the Sysplex distributor quiesce/resume command



Quiesce for a single listener is too granular

- The VARY TCPIP,,SYSPLEX command allows operators to quiesce/resume applications from sysplex distribution
  - Quiesce/Resume sysplex distribution for individual applications identified by port and, optionally, jobname and asid
  - Quiesce/Resume sysplex distribution for all applications on a target stack
  - No impact to existing connections
  - Must be issued on the stack where the application runs
- Quiesce/Resume for an individual application must identify a unique listener or the command will fail
  - To quiesce multiple port listeners for a jobname instance, must issue quiesce for each port
    - ✓ Applications such as the z/OS Websphere Application Server would like to quiesce all listening ports for a given jobname with one command
  - If an application has multiple listeners for the same port, only quiesce target can be used
    - ✓ Applications such as z/OS DB2 have multiple listening sockets for the same port using both Distributed and non-Distributed DVIPAs. They would like to quiesce all sockets with one command

54

Redbooks

ibm.com/redbooks

The existing VARY TCPIP,,SYSPLEX command allows operators to quiesce or resume applications from sysplex distribution. These operations do not impact the existing connections. Furthermore the command must be issued on the stack where the application is running.

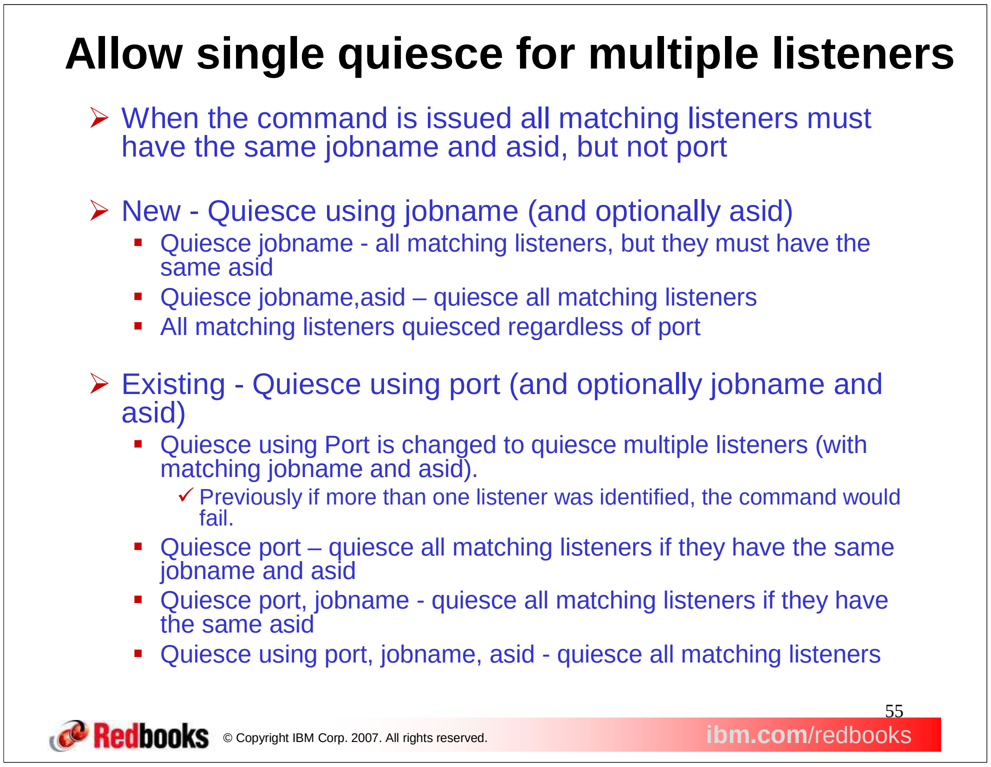
In prior releases to quiesce multiple port listeners for a jobname instance, an operator must issue the quiesce flavor of the command for each port. An application instance may have listeners on different ports. Each of them must be quiesced with a separate command. For example:

```
Quiesce Port=50001,Jobname=jobxyz

Quiesce Port=50002,Jobname=jobxyz

        …

Quiesce Port=50025,Jobname=jobxyz
```

In prior releases, if an application instance has multiple listeners for the same port, only quiesce target can be used. An application may have multiple listeners on port 50001 with the same asid (71). This command will fail since a unique listener is not identified:

```
Quiesce Port=50001,jobname=jobxyz,asid=71
```

The only option is Quiesce Target, but this will quiesce **all** target applications on this stack from sysplex distribution.



Allow single quiesce for multiple listeners

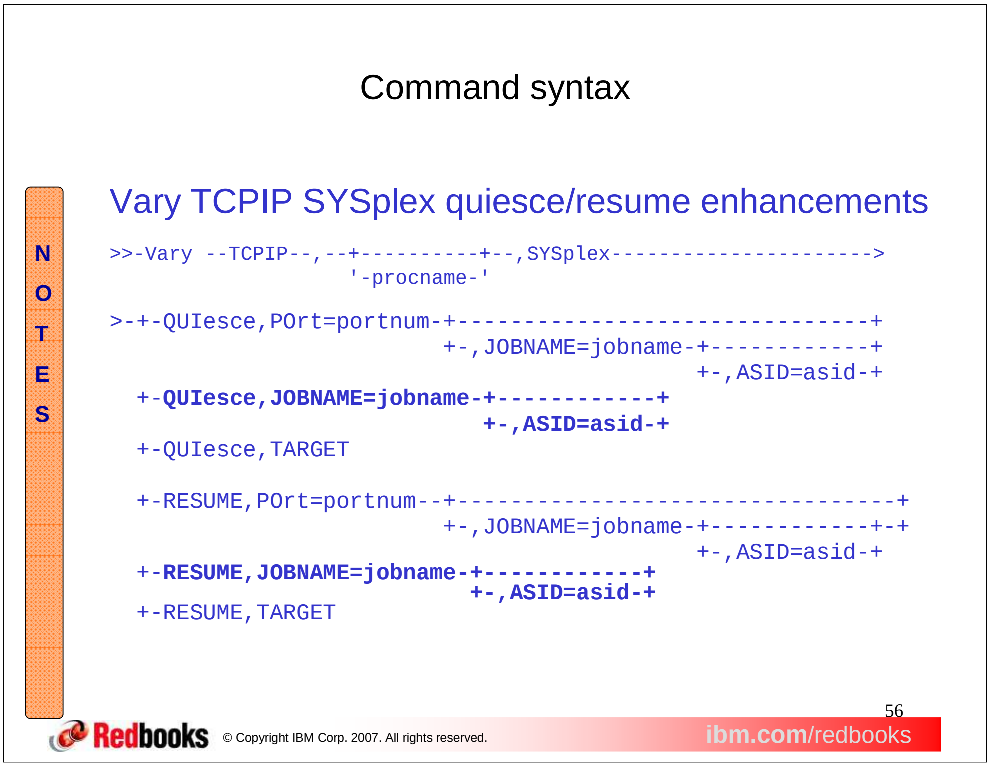
- When the command is issued all matching listeners must have the same jobname and asid, but not port
- New - Quiesce using jobname (and optionally asid)
  - Quiesce jobname - all matching listeners, but they must have the same asid
  - Quiesce jobname,asid – quiesce all matching listeners
  - All matching listeners quiesced regardless of port
- Existing - Quiesce using port (and optionally jobname and asid)
  - Quiesce using Port is changed to quiesce multiple listeners (with matching jobname and asid).
    - ✓ Previously if more than one listener was identified, the command would fail.
  - Quiesce port – quiesce all matching listeners if they have the same jobname and asid
  - Quiesce port, jobname - quiesce all matching listeners if they have the same asid
  - Quiesce using port, jobname, asid - quiesce all matching listeners

Redbooks

ibm.com/redbooks

A new flavor of the Quiesce and Resume Sysplex command is now allowed. A user can now quiesce and resume applications using the jobname and optionally the associated asid. When jobname is used and all listeners identified by jobname do not have the same asid, the command will fail. The listeners will not be quiesced or resumed.

The existing Quiesce/Resume, port or Quiesce/Resume, port,jobname command is changed in z/OS V1R9. When either of these flavors of the command is used, if all listeners do not have the same jobname and asid, the command will fail. The listeners will not be quiesced or resumed. In previous releases, if a unique listener was not identified, the command would fail.



Command syntax

NOTES

Vary TCPIP SYSplex quiesce/resume enhancements

```
>>-Vary --TCPIP--,--+----------+--,SYSplex---------------------->
                    '-procname-'

>-+-QUIesce,POrt=portnum-+------------------------------+
                         +-,JOBNAME=jobname-+-----------+
                                            +-,ASID=asid-+
  +-QUIesce,JOBNAME=jobname-+------------+
                            +-,ASID=asid-+
  +-QUIesce,TARGET

  +-RESUME,POrt=portnum--+--------------------------------+
                         +-,JOBNAME=jobname-+-----------+-+
                                            +-,ASID=asid-+
  +-RESUME,JOBNAME=jobname-+------------+
                          +-,ASID=asid-+
  +-RESUME,TARGET
```

56

ibm.com/redbooks

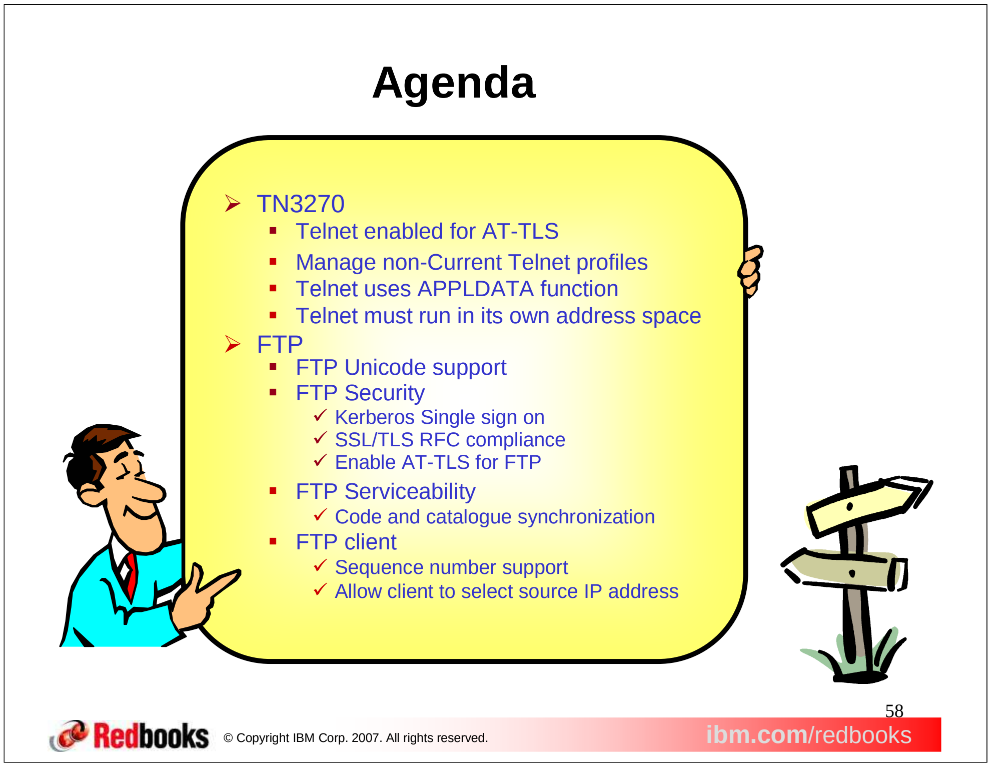
This slide shows the various ways the quiesce/resume command can now be issued. Vary TCPIP Sysplex also supports Leavegroup, Joingroup, Deactivate, and Reactivate; these are not shown here.

ibm.com

Standard TCP/IP applications
TN3270 and FTP

International Technical Support Organization

This presentation describes the changes to TN3270 and FTP in z/OS V1R9.

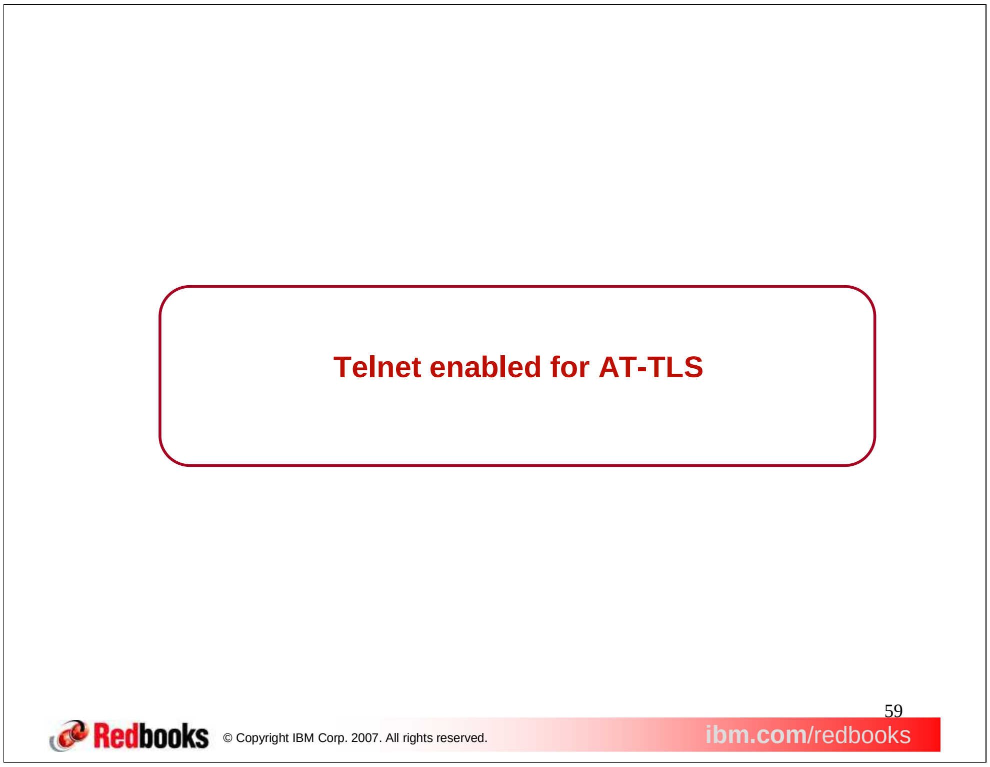


Agenda

- ➢ TN3270
  - Telnet enabled for AT-TLS
  - Manage non-Current Telnet profiles
  - Telnet uses APPLDATA function
  - Telnet must run in its own address space
- ➢ FTP
  - FTP Unicode support
  - FTP Security
    - ✓ Kerberos Single sign on
    - ✓ SSL/TLS RFC compliance
    - ✓ Enable AT-TLS for FTP
  - FTP Serviceability
    - ✓ Code and catalogue synchronization
  - FTP client
    - ✓ Sequence number support
    - ✓ Allow client to select source IP address

In this presentation, we will explain how Telnet has been enabled for Application Transparent – Transport Layer Security, AT-TLS, exploitation, how storage savings are realized by managing non-Current profiles, how Telnet uses the new APPLDATA function, and explain why Telnet must run in its own address space.

For FTP, we will address the more complex line items first – Unicode support and FTP security enhancements. Then we will describe a serviceability enhancement to FTP. We will conclude with enhancements that affect the FTP client only.

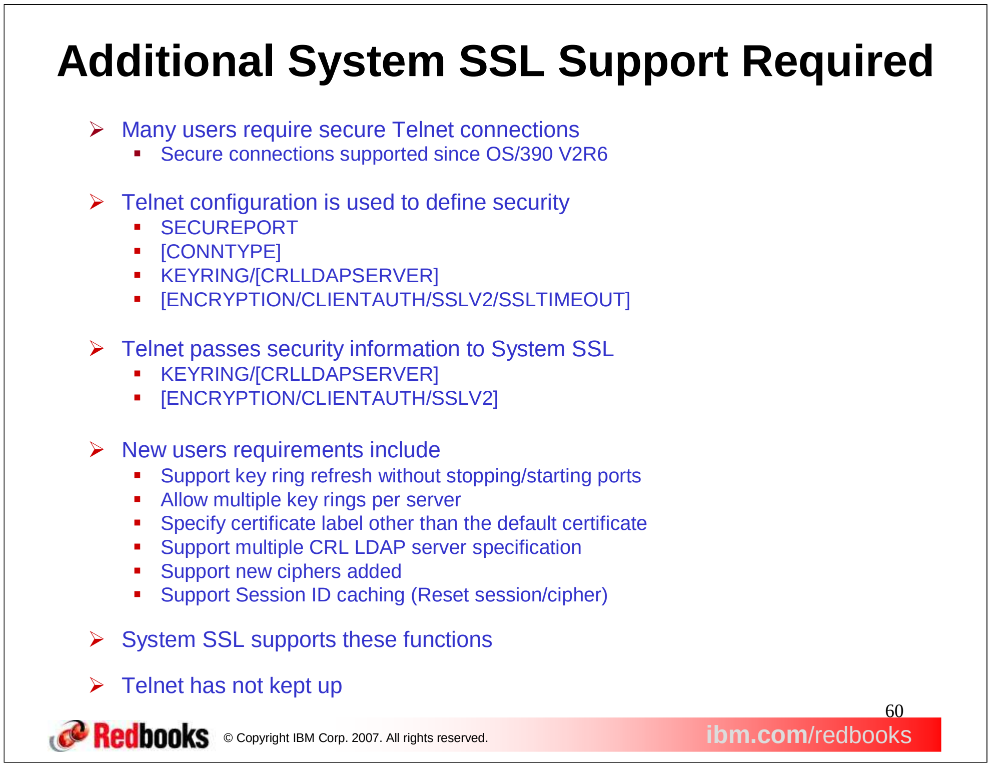


Telnet enabled for AT-TLS

59

ibm.com/redbooks

Telnet has been enabled to use Application Transparent - Transport Layer Security, AT-TLS, the strategic z/OS Communications Server security solution.



Additional System SSL Support Required

- ➢ Many users require secure Telnet connections
  - Secure connections supported since OS/390 V2R6
- ➢ Telnet configuration is used to define security
  - SECUREPORT
  - [CONNTYPE]
  - KEYRING/[CRLLDAPSERVER]
  - [ENCRYPTION/CLIENTAUTH/SSLV2/SSLTIMEOUT]
- ➢ Telnet passes security information to System SSL
  - KEYRING/[CRLLDAPSERVER]
  - [ENCRYPTION/CLIENTAUTH/SSLV2]
- ➢ New users requirements include
  - Support key ring refresh without stopping/starting ports
  - Allow multiple key rings per server
  - Specify certificate label other than the default certificate
  - Support multiple CRL LDAP server specification
  - Support new ciphers added
  - Support Session ID caching (Reset session/cipher)
- ➢ System SSL supports these functions
- ➢ Telnet has not kept up

60

Redbooks

ibm.com/redbooks

For several years, users have required that their Telnet connections be secure. Telnet first implemented secure connections in OS/390 V2R6 which was Generally available in September 1998.

Telnet was developed with a direct interface to the System Secure Sockets Layer, SSL, component. Telnet configuration is used to define the parameters needed for System SSL to set up its environment to support secure connections. SECUREPORT designates that the connections to the Telnet port will be secure and KEYRING specifies the key ring name System SSL will use. Additional, optional parameters are shown in brackets.

In a later release, OS/390 V2R10, the Conntype statement was added to allow both secure and non-secure connections on the same port.

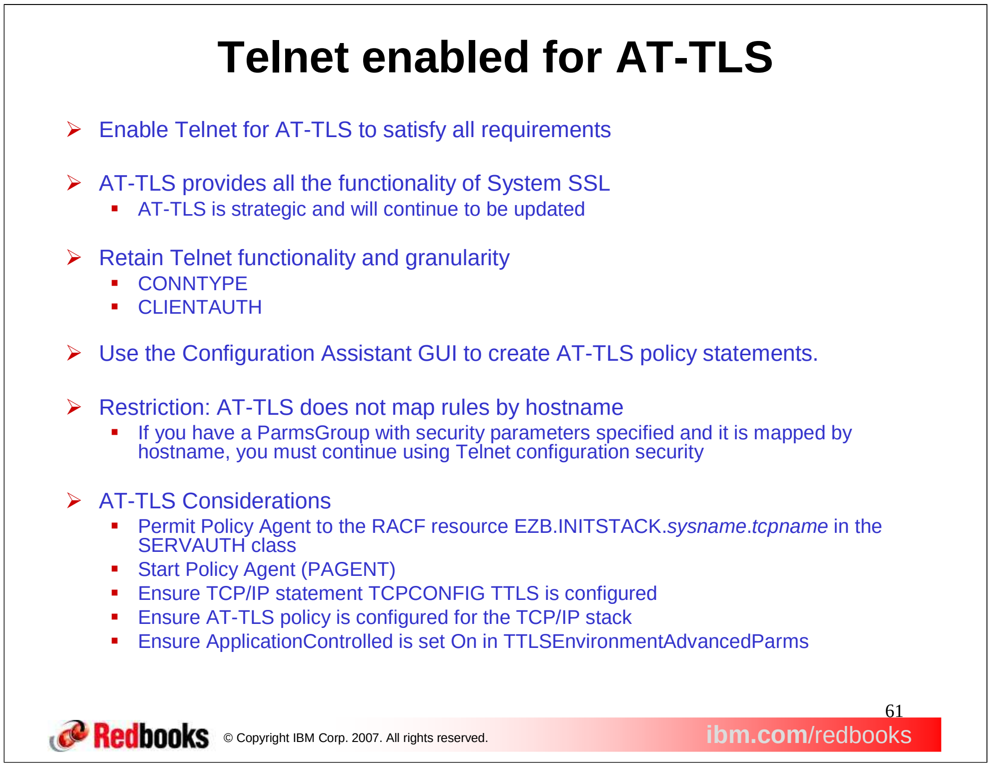
System SSL needs a Key ring to properly set up the System SSL environment used by Telnet. Optionally, CRLLDAPServer, Encryption, ClientAuth, and SSLv2 values can be specified to further customize the System SSL environment.

When Telnet first implemented secure connections on OS/390 V2R6, System SSL was not as robust as it is today. System SSL allowed only one active environment to support telnet connections. Telnet security setup was developed around that assumption and others based on System SSL capability at the time. For example, because only one System SSL environment could be activated, Telnet allows only one key ring name for all ports.

Users have asked for Telnet to support different key rings on different ports and even different key rings on the same port. Users have a need to be able to refresh security parameters without having to stop/restart the secure ports. This is particularly useful when the default certificate expires and must be replaced. Some users have backup Certificate Revocation List Lightweight Directory Access Protocol, CRL LDAP, servers and would like to specify these backups. Users would like to quickly use new ciphers that are periodically added. Users have client emulators that support session ID caching and renegotiation of a cipher key during an active secure session. Users also want to specify a certificate label to be used instead of the default key ring certificate.

System SSL has continued to improve and now supports these functions.

Telnet configuration has not been enhanced to take advantage of the new System SSL function. We did recently add two AES ciphers to Telnet but Telnet still does not support all the ciphers available for use by System SSL.



Telnet enabled for AT-TLS

- Enable Telnet for AT-TLS to satisfy all requirements
- AT-TLS provides all the functionality of System SSL
  - AT-TLS is strategic and will continue to be updated
- Retain Telnet functionality and granularity
  - CONNTYPE
  - CLIENTAUTH
- Use the Configuration Assistant GUI to create AT-TLS policy statements.
- Restriction: AT-TLS does not map rules by hostname
  - If you have a ParmsGroup with security parameters specified and it is mapped by hostname, you must continue using Telnet configuration security
- AT-TLS Considerations
  - Permit Policy Agent to the RACF resource EZB.INITSTACK.*sysname.tcpname* in the SERVAUTH class
  - Start Policy Agent (PAGENT)
  - Ensure TCP/IP statement TCPCONFIG TTLS is configured
  - Ensure AT-TLS policy is configured for the TCP/IP stack
  - Ensure ApplicationControlled is set On in TTLSEnvironmentAdvancedParms

61

Redbooks ibm.com/redbooks

Application Transparent Transport Layer Security (AT-TLS) was introduced in z/OS V1R7 and supports all of the new functions in System SSL. AT-TLS is the z/OS Communications server strategic application security option and will continue to be updated as new System SSL functions become available.

To satisfy existing Telnet security requirements, we could either make additional updates to Telnet configuration to make use of the new System SSL function or enable Telnet to fully utilize AT-TLS. Because AT-TLS is strategic and provides System SSL functions beyond the current requirements, we chose to enable Telnet for AT-TLS. With AT-TLS the user will be able to specify multiple key rings for different ports or the same port, change key rings without stopping ports, specify up to five CRL LDAP servers, specify new ciphers immediately, cache session IDs, manage session IDs and cipher renegotiation, and use a certificate other than the default certificate during the SSL negotiations.

Telnet provides the user a great deal of flexibility through its current configuration options. That flexibility had to be retained while moving to AT-TLS. Being able to specify Conntype and client authentication at very granular levels is a popular Telnet feature that must be retained.

There are no migration issues with AT-TLS for Telnet. You can continue to define secure ports with the SECUREPORT statement, but you will have access to many more System SSL functions if you use TTLSPORT.

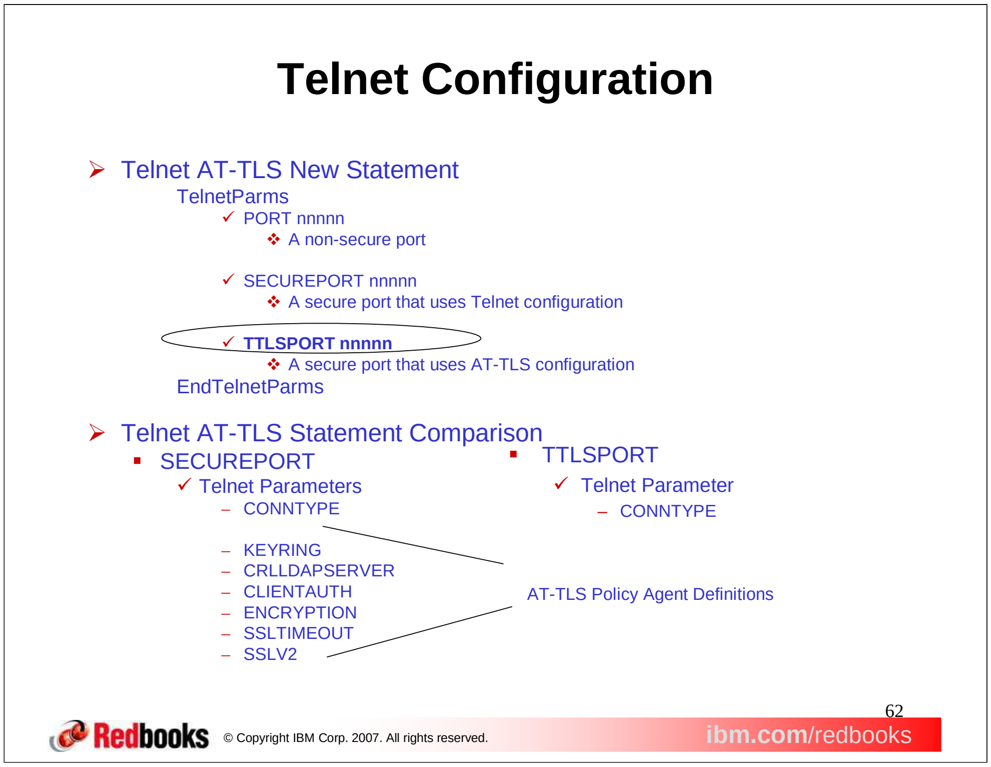
In some cases the creation of AT-TLS policy files for Policy Agent can be difficult. The Configuration Assistant GUI will, through a series of wizards and online help panels, generate AT-TLS configuration files for any number of z/OS images with any number of TCPIP stacks per image.

TTLSRULE is, in some cases, a replacement for the ParmsGroup and ParmsMap statements. In Telnet, you can define a client identifier as a hostname or a hostname group and map a ParmsGroup to that client identifier. That ParmsGroup may contain security parameters. TTLSRULE does not support the concept of a connection matching a specified hostname. If you use hostnames as a client identifier to differentiate security variables you will have to continue to use the Telnet configuration for security.

To install the AT-TLS policy you need to permit the Policy Agent to the INITSTACK Resource Access Control Facility, RACF, resource in the SERVAUTH class and start the Policy Agent. If Telnet connections fail with error code 100B, “Unexpected SSL handshake encountered.”, AT-TLS is probably not enabled in the stack. All connections are considered non-secure but the client is sending a secure handshake. You need to ensure that TCPCONFIG TTLS has been configured in the TCP/IP profile.

You also need to ensure there is a TCPImage statement for the TCP/IP stack used by Telnet and ensure the TCPImage policy file has a TTLSCONFIG statement pointing to the configuration file that defines the Telnet AT-TLS rules.

If Telnet connections fail with error code 1035, “Policy is invalid for the conntype specified.”, either there is no policy configured for the connection or the policy did not specify that for Telnet, AT-TLS must be application controlled.



Telnet Configuration

- ➢ Telnet AT-TLS New Statement
  - TelnetParms
    - ✓ PORT nnnnn
      - ❖ A non-secure port
    - ✓ SECUREPORT nnnnn
      - ❖ A secure port that uses Telnet configuration
    - ✓ **TTLSPORT nnnnn**
      - ❖ A secure port that uses AT-TLS configuration
  - EndTelnetParms
- ➢ Telnet AT-TLS Statement Comparison
  - ▪ SECUREPORT
    - ✓ Telnet Parameters
      - – CONNTYPE
      - – KEYRING
      - – CRLLDAPSERVER
      - – CLIENTAUTH
      - – ENCRYPTION
      - – SSLTIMEOUT
      - – SSLV2
  - ▪ TTLSPORT
    - ✓ Telnet Parameter
      - – CONNTYPE

AT-TLS Policy Agent Definitions

62

Redbooks ibm.com/redbooks

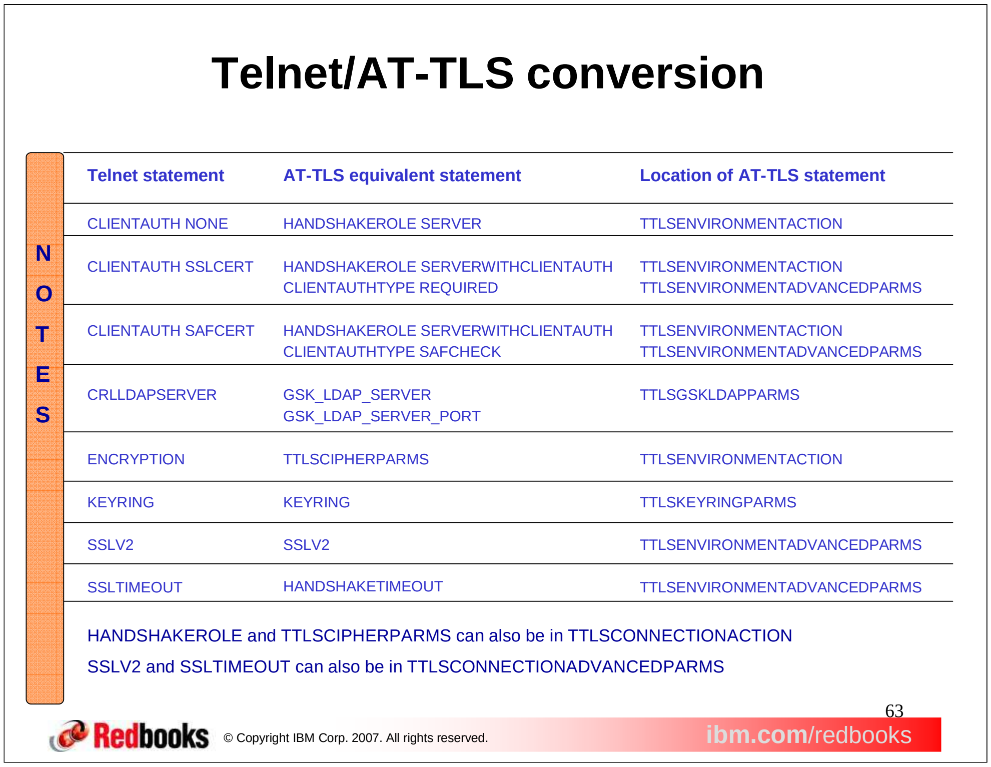
Ports are defined as either non-secure basic ports or secure ports by specifying either PORT nnnnn or SECUREPORT nnnnn. A basic port can support only non-secure connections. A SECUREPORT interacts with System SSL to create a single System SSL environment used by the port. A SECUREPORT can support secure or non-secure connections depending on the Conntype value. Security-related parameters can be specified only for a SECUREPORT.

A new port definition, TTLSPORT, is created to define a Telnet port that uses AT-TLS security configuration instead of using Telnet configuration. To ease migration to TTLSPORT, the SECUREPORT option is retained at the current level of functionality.

Several security configuration statements used by SECUREPORT to create a System SSL environment are moved to AT-TLS when a TTLSPORT is defined. AT-TLS configuration is managed using the Policy Agent. For either port type, a System SSL environment must be created. If SECUREPORT is used, Telnet configuration defines the values used to create the environment. If TTLSPORT is used, AT-TLS configuration defines the values used to create the environment.

Notice the Conntype statement remains for both port types. Conntype is not a System SSL environment variable. Conntype is used by Telnet to decide if System SSL should be used to set up security for a particular connection.

The notes page that follows, titled “Telnet/AT-TLS Conversion”, gives the conversion for all Telnet to AT-TLS statements and a detailed discussion of how to implement client authentication in AT-TLS. Following the notes page is an example showing both the Telnet configuration and the AT-TLS configuration.



Telnet/AT-TLS conversion

NOTES

Telnet statement	AT-TLS equivalent statement	Location of AT-TLS statement
CLIENTAUTH NONE	HANDSHAKEROLE SERVER	TTLSENVIRONMENTACTION
CLIENTAUTH SSLCERT	HANDSHAKEROLE SERVERWITHCLIENTAUTH CLIENTAUTHTYPE REQUIRED	TTLSENVIRONMENTACTION TTLSENVIRONMENTADVANCEDPARMS
CLIENTAUTH SAFCERT	HANDSHAKEROLE SERVERWITHCLIENTAUTH CLIENTAUTHTYPE SAFCHECK	TTLSENVIRONMENTACTION TTLSENVIRONMENTADVANCEDPARMS
CRLLDAPSERVER	GSK_LDAP_SERVER GSK_LDAP_SERVER_PORT	TTLSGSKLDAPPARMS
ENCRYPTION	TTLSCIPHERPARMS	TTLSENVIRONMENTACTION
KEYRING	KEYRING	TTLSKEYRINGPARMS
SSLV2	SSLV2	TTLSENVIRONMENTADVANCEDPARMS
SSLTIMEOUT	HANDSHAKETIMEOUT	TTLSENVIRONMENTADVANCEDPARMS

HANDSHAKEROLE and TTLSCIPHERPARMS can also be in TTLSCONNECTIONACTION

SSLV2 and SSLTIMEOUT can also be in TTLSCONNECTIONADVANCEDPARMS

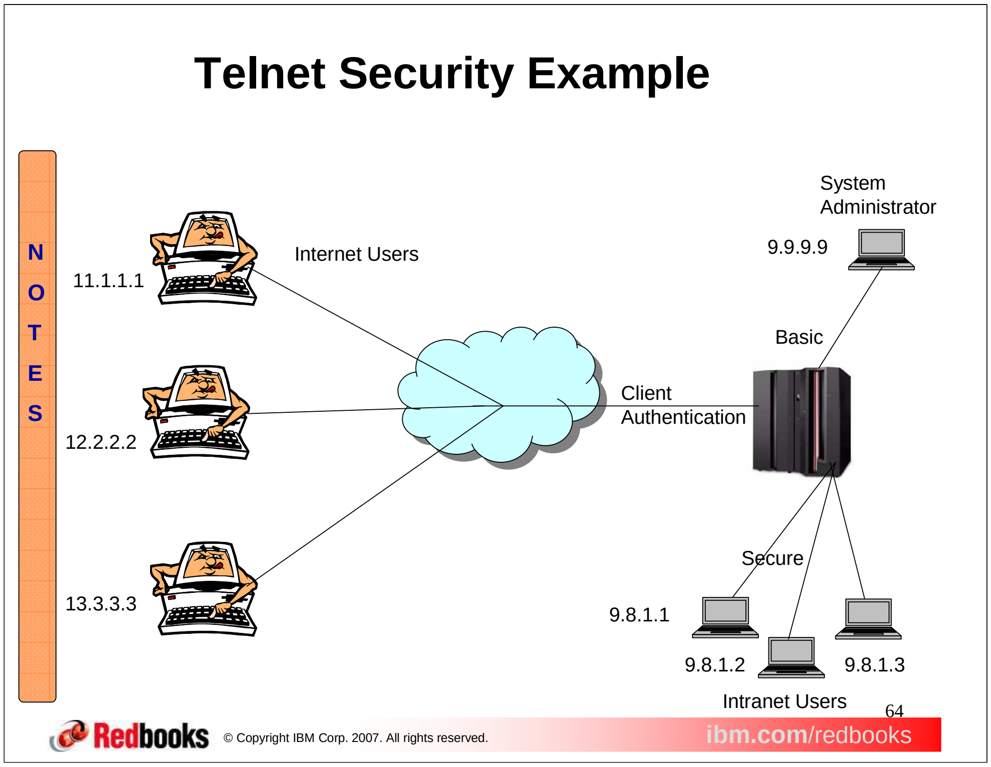
This table shows the AT-TLS equivalent statements for the Telnet security statements.

There are many variations possible with the Telnet profile statement CLIENTAUTH. In AT-TLS, whether or not client authentication is done is controlled by the HandshakeRole parameter on either the TTLSEnvironmentAction or TTLSConnectionAction statements. If the connection needs client authentication, the level of authentication is controlled with the ClientAuthType parameter on the TTLSEnvironmentAdvancedParms statement.

If you have both CLIENTAUTH SSLCERT and CLIENTAUTH SAFCERT in different ParmsGroup statements in your Telnet configuration, you need two TTLSEnvironmentAction statements; one TTLSEnvironmentAction statement for ClientAuthType Required and one TTLSEnvironmentAction statement for ClientAuthType SAFCheck. Two TTLSRule statements, each referencing a different TTLSEnvironmentAction statement in AT-TLS, replace the two PARMSMAP statements in the Telnet profile.

If you have a mixture of CLIENTAUTH NONE and CLIENTAUTH SAFCERT, you need a TTLSEnvironmentAction statement with HandshakeRole ServerWithClientAuth, and a TTLSConnectionAction statement with HandshakeRole Server. Two TTLSRule statements in AT-TLS (one with the TTLSConnectionAction statement and one without) replace the two PARMSMAP statements in the Telnet profile. You could instead create a second TTLSEnvironmentAction statement with HandshakeRole Server, but many more resources are associated with a TTLSEnvironmentAction statement compared to a TTLSConnectionAction

System SSL defines client authentication type only at the environment level and controls whether or not client authentication is performed by the handshake role which can be specified at the environment or connection level.



Telnet Security Example

NOTES

Internet Users

11.1.1.1

12.2.2.2

13.3.3.3

System Administrator

9.9.9.9

Basic

Client Authentication

Secure

9.8.1.1

9.8.1.2

9.8.1.3

Intranet Users

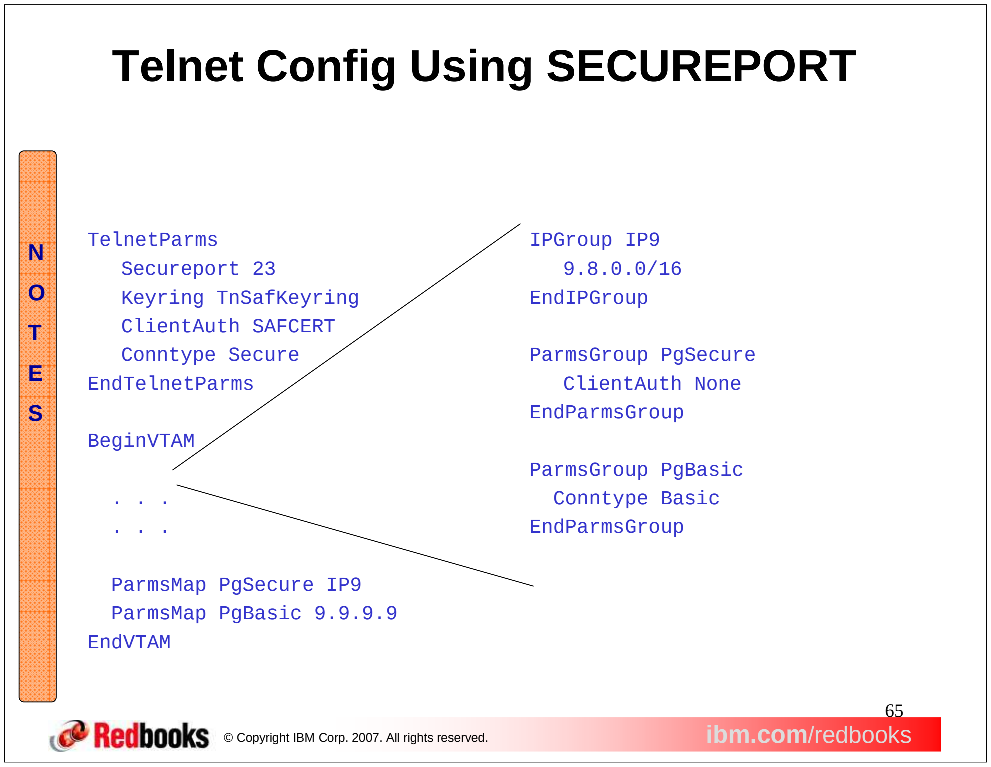
64

Redbooks

ibm.com/redbooks

The easiest way to show the difference between SECUREPORT and TTLSPORT configuration is through an example.

Assume the following environment. We have several end users in the internet who require secure connections with server authentication and a client authentication level that requires a mapping of the client certificate to a Security server user ID. Several more end users are on the company intranet where secure connections with only server authentication is required. Finally, the system administrator has a fixed IP address and does not require any security.



Telnet Config Using SECUREPORT

```
TelnetParms
   Secureport 23
   Keyring TnSafKeyring
   ClientAuth SAFCERT
   Conntype Secure
EndTelnetParms

BeginVTAM

  . . .
  . . .

  ParmsMap PgSecure IP9
  ParmsMap PgBasic 9.9.9.9
EndVTAM
```

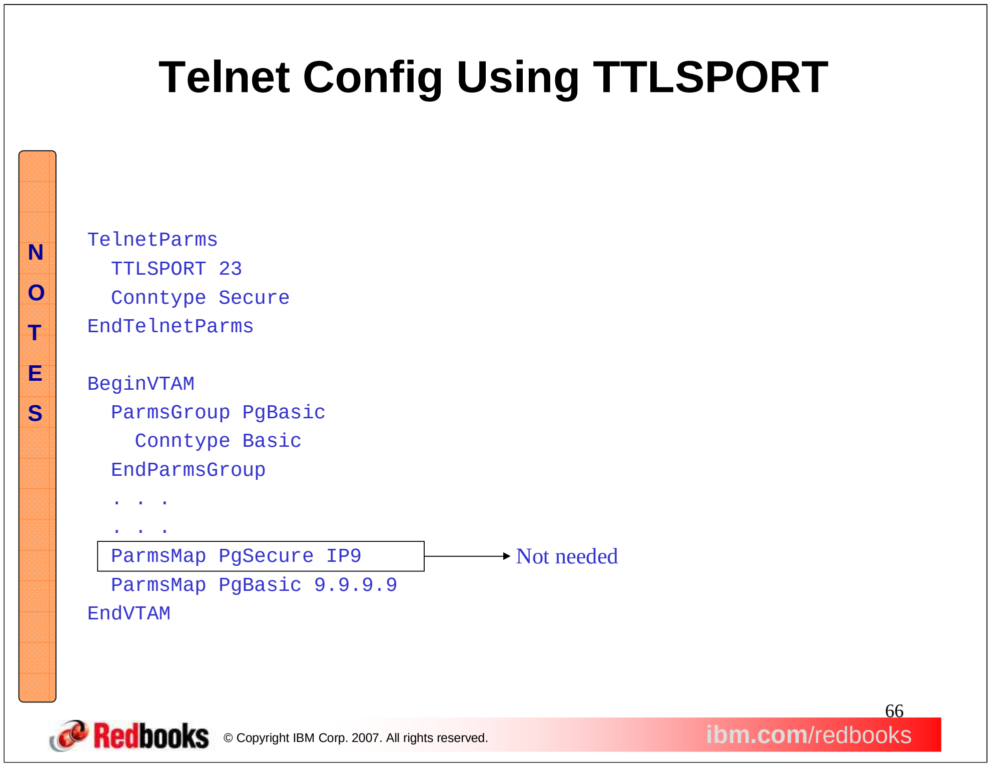
```
IPGroup IP9
   9.8.0.0/16
EndIPGroup

ParmsGroup PgSecure
   ClientAuth None
EndParmsGroup

ParmsGroup PgBasic
  Conntype Basic
EndParmsGroup
```

First we'll look at the security statements needed when SECUREPORT is used to define port security for the example from the previous slide.

Within TelnetParms the key ring name (TNSafKeyring), client authentication level (SAFCert), and connection type (Secure) are specified. Conntype secure is the default but is shown for completeness. The key ring and client authentication levels are passed to System SSL to create the System SSL environment used for securing connections at the port level. If nothing else were coded, all connections would be secure and require client authentication, satisfying the internet requirement on the previous slide.

An IPGroup, IP9, defines a subset of connections that represent all intranet company users. A Parmsgroup, PgSecure, defines internal security with client authentication off. The ParmsMap statement is used to map PgSecure to IP9. The result is any connection IP address starting with 9.8 will not be required to provide client authentication, satisfying the intranet requirement on the previous slide.

Another Parmsgroup, PgBasic, defines a connection type as basic (non-secure). This Parmsgroup is mapped to the system administrator's IP address directly giving the system administrator a basic connection, satisfying the administrator requirement on the previous slide.



Telnet Config Using TTLSPORT

NOTES

```
TelnetParms
  TTLSPORT 23
  Conntype Secure
EndTelnetParms

BeginVTAM
  ParmsGroup PgBasic
    Conntype Basic
  EndParmsGroup
  . . .
  . . .
  ParmsMap PgSecure IP9
  ParmsMap PgBasic 9.9.9.9
EndVTAM
```

Not needed

Redbooks

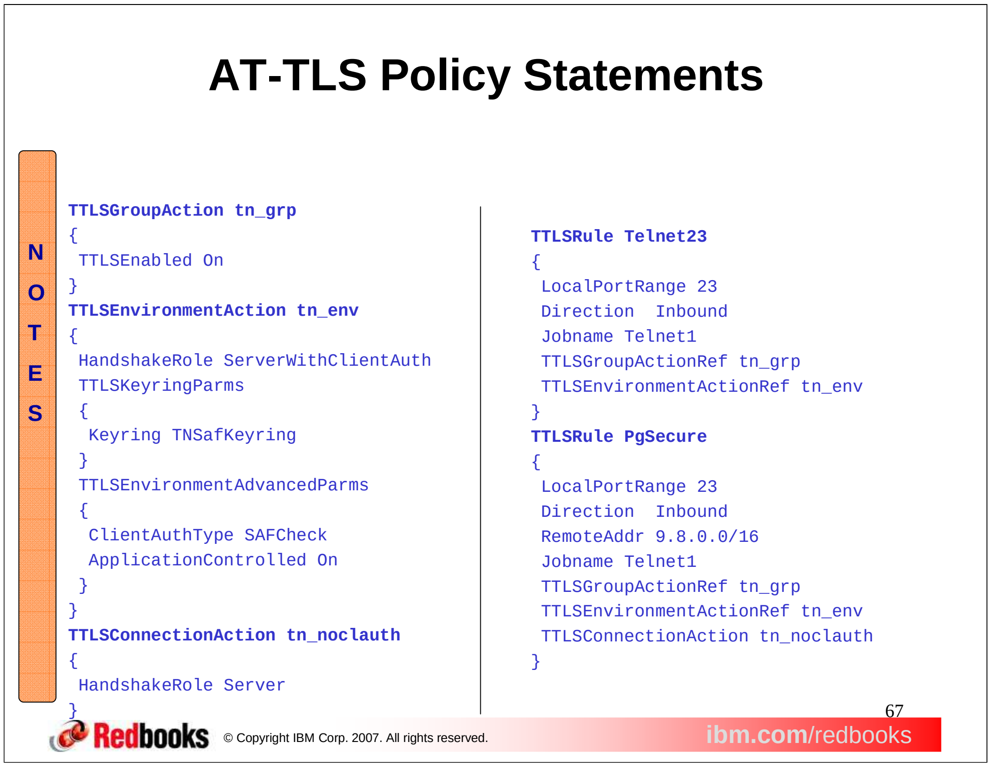
ibm.com/redbooks

Telnet security statements are not needed when TTLSPORT is used to define the secure port.

TelnetParms uses the TTLSPORT statement and does not need the Key ring and client authentication statements that are now configured in AT-TLS policy in the Policy Agent. Connection type is controlled by Telnet and must still be specified. Conntype secure is the default but is shown for completeness. Combined with the AT-TLS statements on the next slide the internet requirements from the example are satisfied.

Because client authentication control is done by AT-TLS the IP9 IPGroup and PgSecure Parmsgroup are not needed in Telnet. Combined with the AT-TLS PgSecure rule on the next slide the intranet requirements from the example are satisfied.

Because Telnet controls whether or not security is initiated, the PgBasic Parmsgroup remains in Telnet and continues to be mapped to IP address 9.9.9.9 to give the system administrator a basic, non-secure, connection which satisfies the administrator requirement from the example.



AT-TLS Policy Statements

```
TTLSGroupAction tn_grp
{
 TTLSEnabled On
}
TTLSEnvironmentAction tn_env
{
 HandshakeRole ServerWithClientAuth
 TTLSKeyringParms
 {
  Keyring TNSafKeyring
 }
 TTLSEnvironmentAdvancedParms
 {
  ClientAuthType SAFCheck
  ApplicationControlled On
 }
}
TTLSConnectionAction tn_noclauth
{
 HandshakeRole Server
}
```

```
TTLSRule Telnet23
{
 LocalPortRange 23
 Direction  Inbound
 Jobname Telnet1
 TTLSGroupActionRef tn_grp
 TTLSEnvironmentActionRef tn_env
}
TTLSRule PgSecure
{
 LocalPortRange 23
 Direction  Inbound
 RemoteAddr 9.8.0.0/16
 Jobname Telnet1
 TTLSGroupActionRef tn_grp
 TTLSEnvironmentActionRef tn_env
 TTLSConnectionAction tn_noclauth
}
```

67

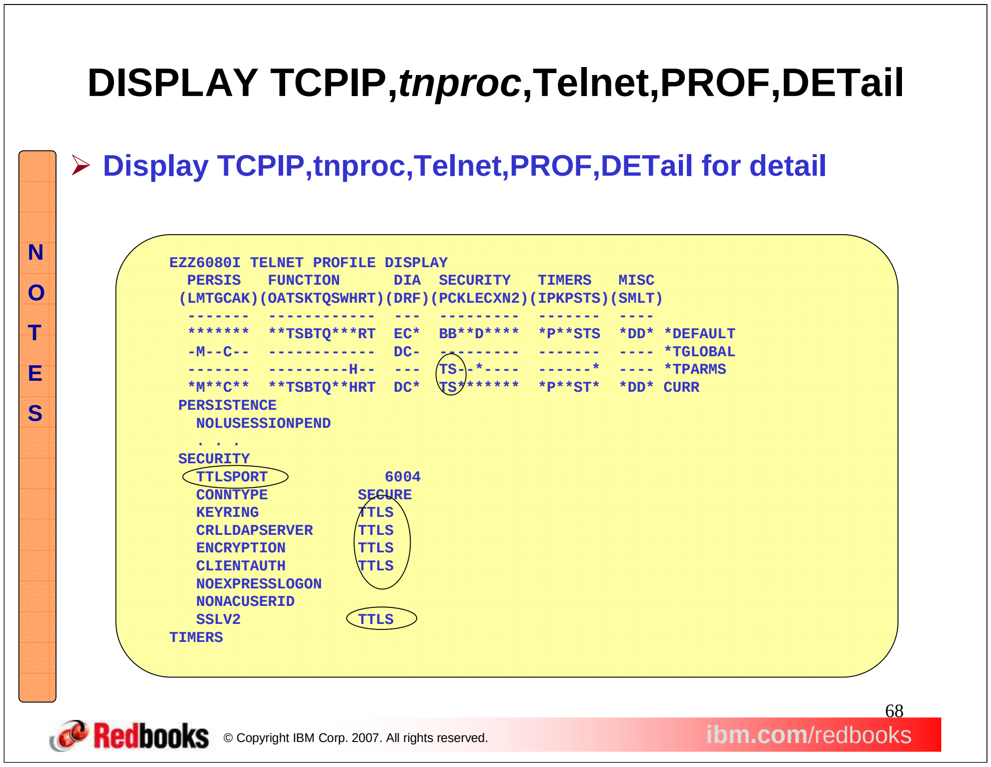
ibm.com/redbooks

TTLSRULEs require group and environment actions. Connection actions are optional. The group action, TTLSGroupAction, defines high level settings such as if AT-TLS is enabled. The Environment action, TTLSEnvironmentAction, defines the System SSL environment and is where key ring and client authentication are defined. The connection action, TTLSConnectionAction, is used to override certain environment settings at the connection level.

TTLSRULE Telnet23 is similar to the TelnetParms statement defining security for the entire port and will define the security for the internet users from the example on slide 11. The group action enables AT-TLS. The environment action specifies that key ring TNSafKeyring should be used and the client authentication level is SAFCheck. The AT-TLS parameter SAFCheck is equivalent to the Telnet parameter SAFCert. Combined with the Telnet statements on the previous slide the internet requirements from the example on slide 11 are satisfied.

TTLSRULE PgSecure uses the RemoteAddr statement to specify the internal subset of port 23 connections. The connection action is also specified. The connection action turns off client authentication by setting handshake role to Server instead of ServerWithClientAuth. The result is all internal clients will have secure connections without client authentication. This TTLSRULE essentially replaces the PgSecure ParmsGroup, IP9 IPGroup, and the ParmsMap statement in Telnet. Combined with the Telnet statements on the previous slide the intranet requirements from the example on slide 11 are satisfied.

AT-TLS is not referenced by the system administrator connection based on the Telnet Conntype Basic statement.



DISPLAY TCPIP,*tnproc*,Telnet,PROF,DETail

N O T E S

- Display TCPIP,tnproc,Telnet,PROF,DETail for detail

```
EZZ6080I TELNET PROFILE DISPLAY
  PERSIS   FUNCTION      DIA  SECURITY   TIMERS   MISC
 (LMTGCAK)(OATSKTQSWHRT)(DRF)(PCKLECXN2)(IPKPSTS)(SMLT)
  -------  ------------  ---  ---------  -------  ----
  *******  **TSBTQ***RT  EC*  BB**D****  *P**STS  *DD* *DEFAULT
  -M--C--  ------------  DC-  ---------  -------  ---- *TGLOBAL
  -------  ---------H--  ---  TS--*----  ------*  ---- *TPARMS
  *M**C**  **TSBTQ**HRT  DC*  TS*******  *P**ST*  *DD* CURR
 PERSISTENCE
   NOLUSESSIONPEND
   . . .
 SECURITY
   TTLSPORT              6004
   CONNTYPE           SECURE
   KEYRING            TTLS
   CRLLDAPSERVER      TTLS
   ENCRYPTION         TTLS
   CLIENTAUTH         TTLS
   NOEXPRESSLOGON
   NONACUSERID
   SSLV2              TTLS
TIMERS
```

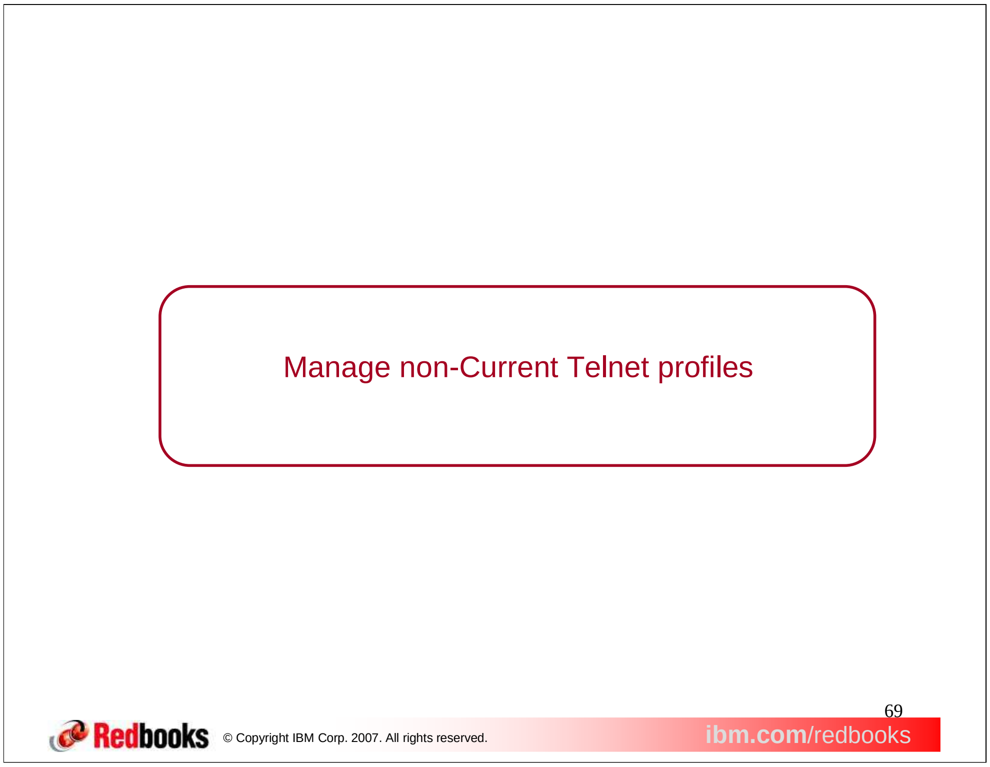
68

Redbooks ibm.com/redbooks

The Telnet profile detail display shows the values of all parameters in the profile. With TTLSPORT, most configuration variables have moved from Telnet configuration to AT-TLS configuration which is represented by the TTLS values.

TS indicates a Secure port defined by the TTLSPORT statement with AT-TLS configuration and a Connection type of Secure.

TTLS indicates the variable is defined in the AT-TLS policy and not in Telnet.

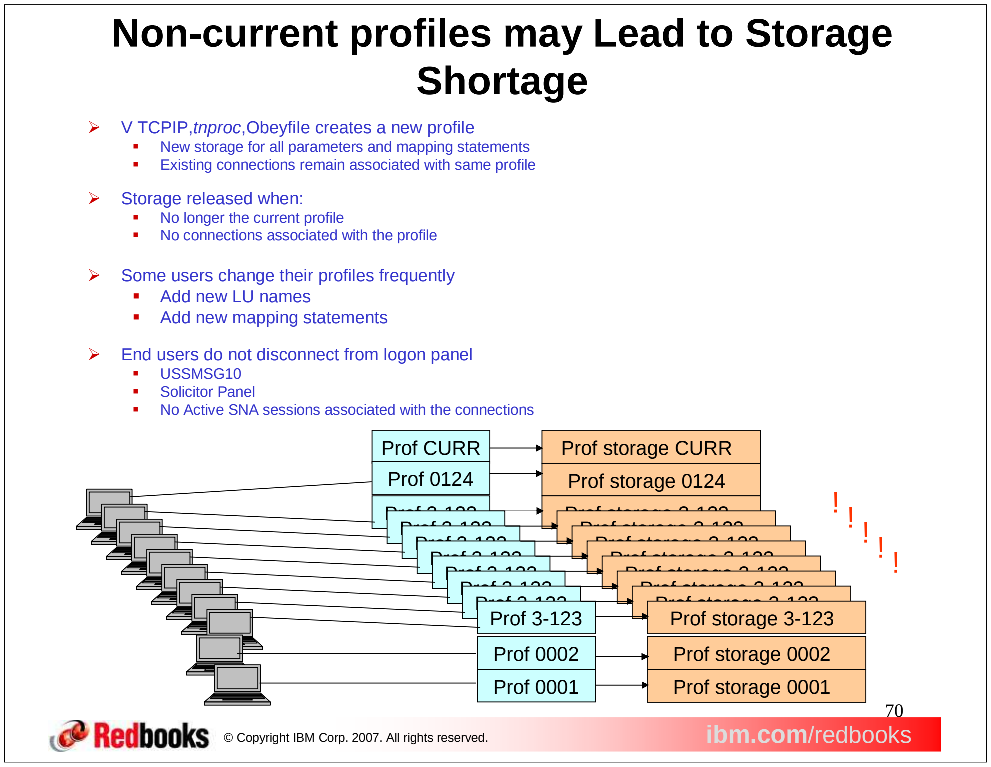


Manage non-Current Telnet profiles

69

ibm.com/redbooks

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



Non-current profiles may Lead to Storage Shortage

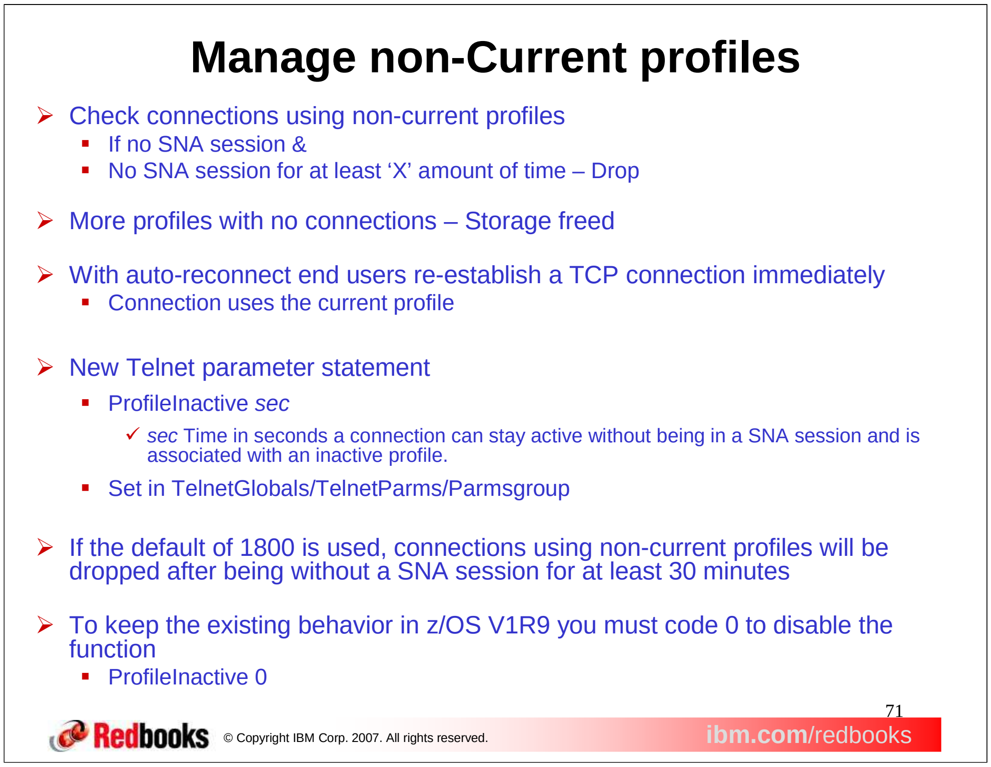
- V TCPIP,*tnproc*,Obeyfile creates a new profile
  - New storage for all parameters and mapping statements
  - Existing connections remain associated with same profile
- Storage released when:
  - No longer the current profile
  - No connections associated with the profile
- Some users change their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

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.

Some users 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.



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
- 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.
  - Set in TelnetGlobals/TelnetParms/Parmsgroup
- If the default of 1800 is used, connections using non-current profiles will be dropped after being without a SNA session for at least 30 minutes
- To keep the existing behavior in z/OS V1R9 you must code 0 to disable the function
  - ProfileInactive 0

71

Redbooks ibm.com/redbooks

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.

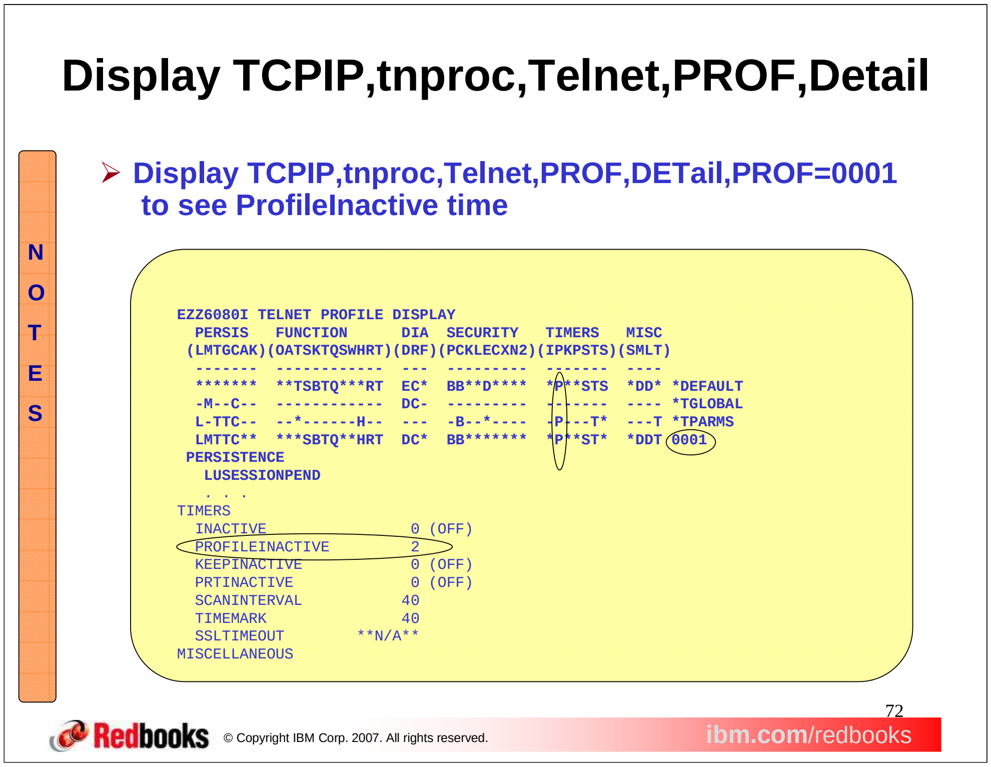
When the connection is dropped, that is one less connection associated with the old profile. When the connection count goes to zero, the profile parameter and mapping structures can be released, freeing potentially large amounts of storage.

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

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 desired.

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,PrtInactive, and KeepInactive, the connection will be dropped sometime soon after 30 minutes, but probably not at precisely 30 minutes.

Remember that the default in z/OS V1R9 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.



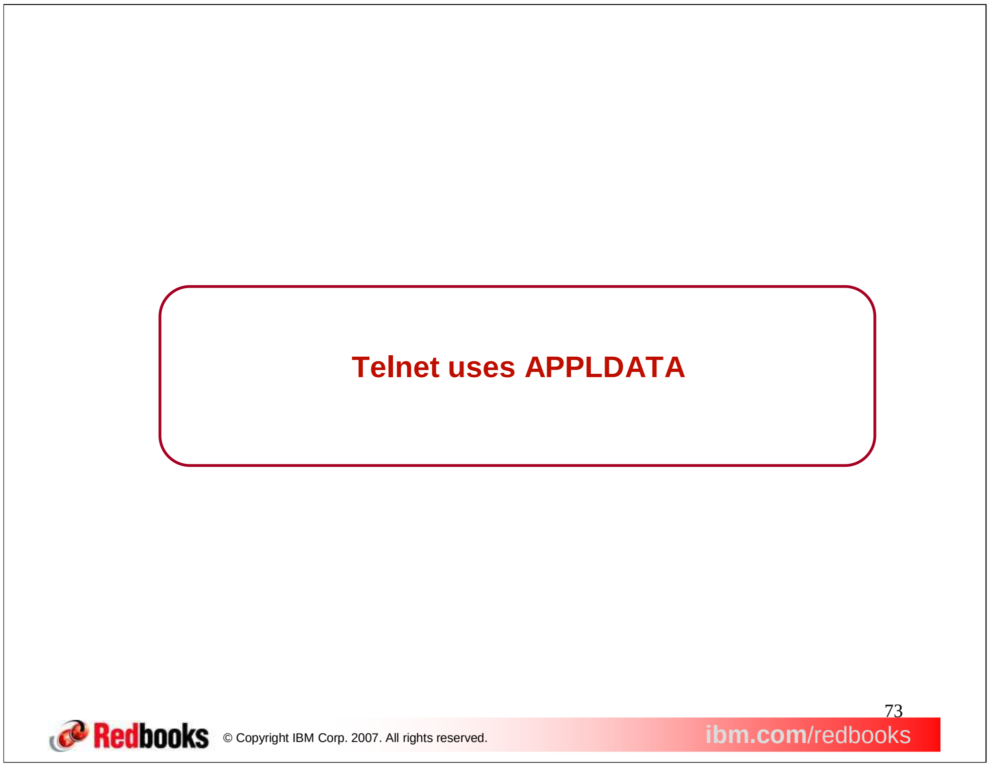
Display TCPIP,tnproc,Telnet,PROF,Detail

- 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)(PCKLECXN2)(IPKPSTS)(SMLT)
  -------  ------------  ---  ---------  -------  ----
  *******  **TSBTQ***RT  EC*  BB**D****  *P**STS  *DD* *DEFAULT
  -M--C--  ------------  DC-  ---------  -------  ---- *TGLOBAL
  L-TTC--  --*------H--  ---  -B--*----  -P---T*  ---T *TPARMS
  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
```

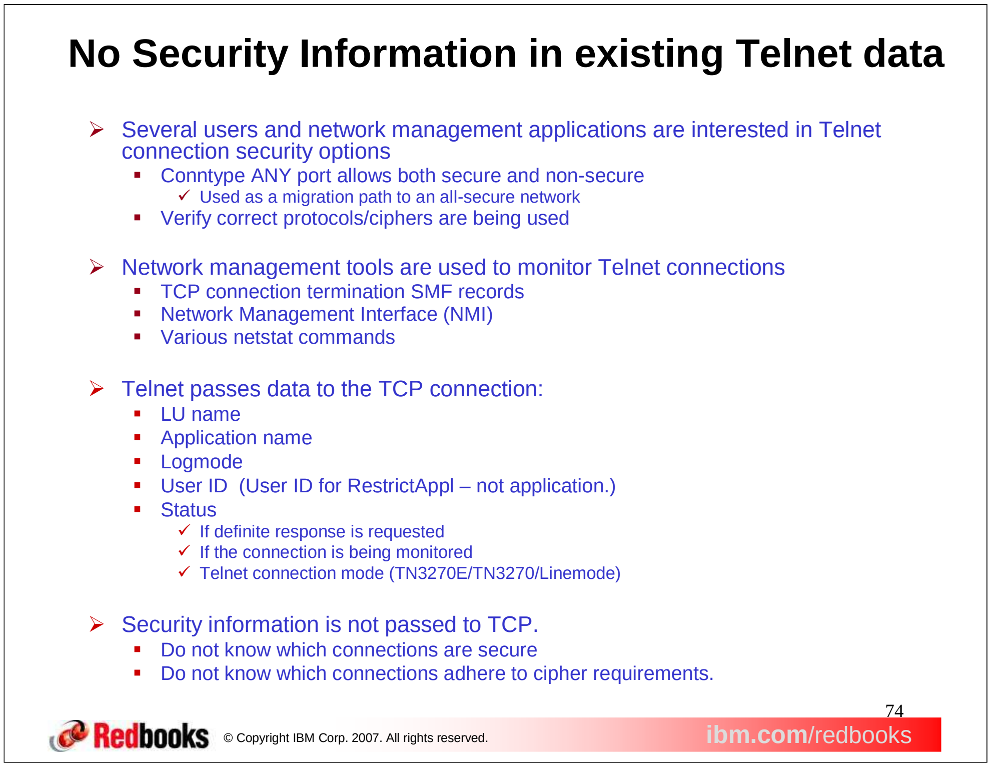
72

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.



Telnet uses APPLDATA

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



No Security Information in existing Telnet data

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

74

Redbooks

ibm.com/redbooks

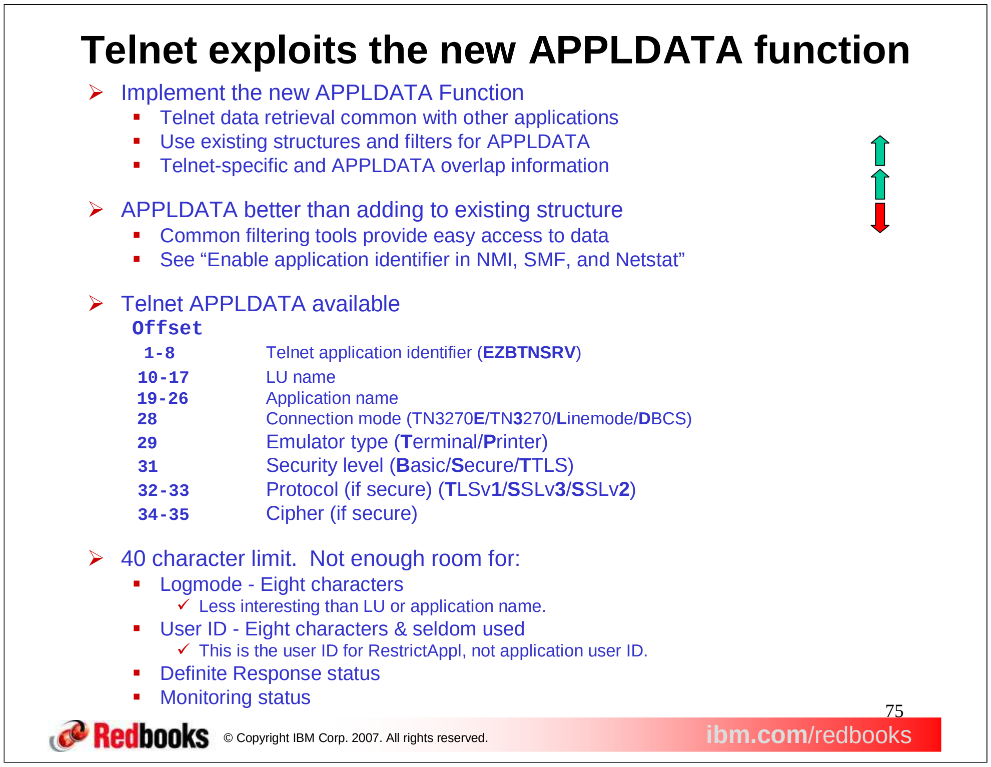
As security becomes more prevalent, many administrators are using the Conntype ANY option in Telnet to slowly migrate their end users to secure connections. With Conntype ANY, a single port can be used for both secure and non-secure connections depending on the client connection mode choice. These administrators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

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.



Telnet exploits the new APPLDATA function

- 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 (**EZBTNSRV**)
10-17	LU name
19-26	Application name
28	Connection mode (TN3270**E**/TN**3**270/**L**inemode/**D**BCS)
29	Emulator type (**T**erminal/**P**rinter)
31	Security level (**B**asic/**S**ecure/**T**TLS)
32-33	Protocol (if secure) (**T**LSv**1**/**S**SLv**3**/**S**SLv**2**)
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

75

© Copyright IBM Corp. 2007. All rights reserved.

ibm.com/redbooks

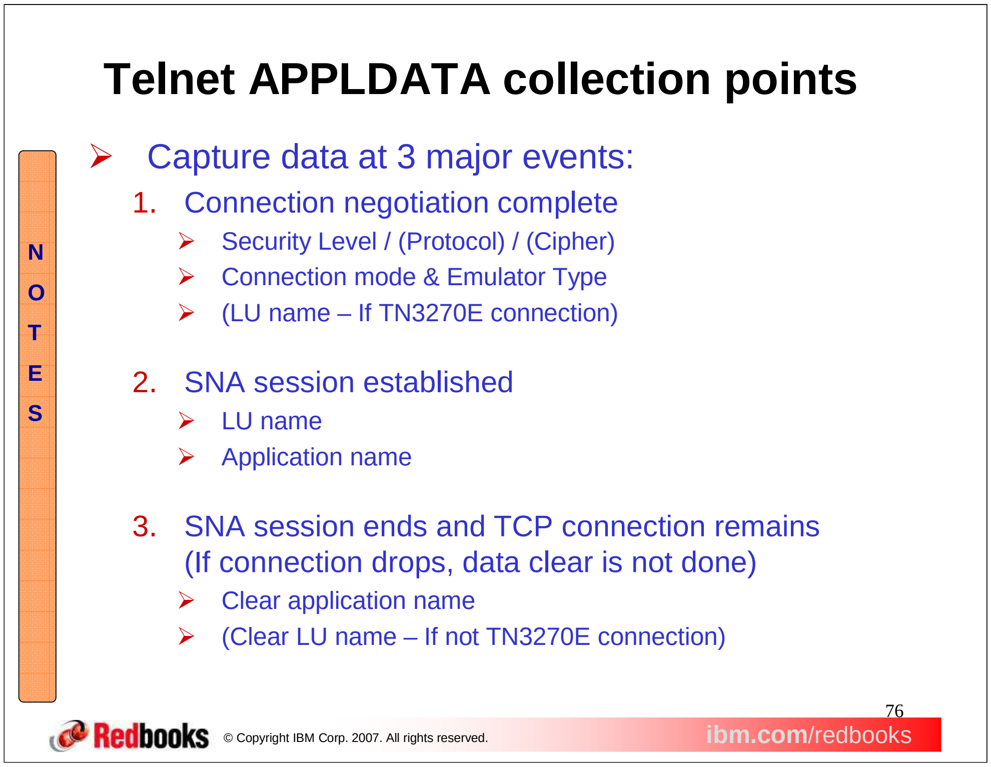
The new APPLDATA area provides a strategic location for Telnet application data passed to the TCP connection. There are several advantages to using the APPLDATA area. APPLDATA is a common area for all applications allowing Telnet queries to use standard filters instead of requiring unique tools. An NMI 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 user will be able to immediately generate summary and detail reports documenting the security level of Telnet connections.

We considered 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 we are adding new security information. 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. We do not know of any requirements to filter on Definite Response status or Monitoring status. These flags could be added if a requirement exists.



Telnet APPLDATA collection points

NOTES

- Capture data at 3 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)

76

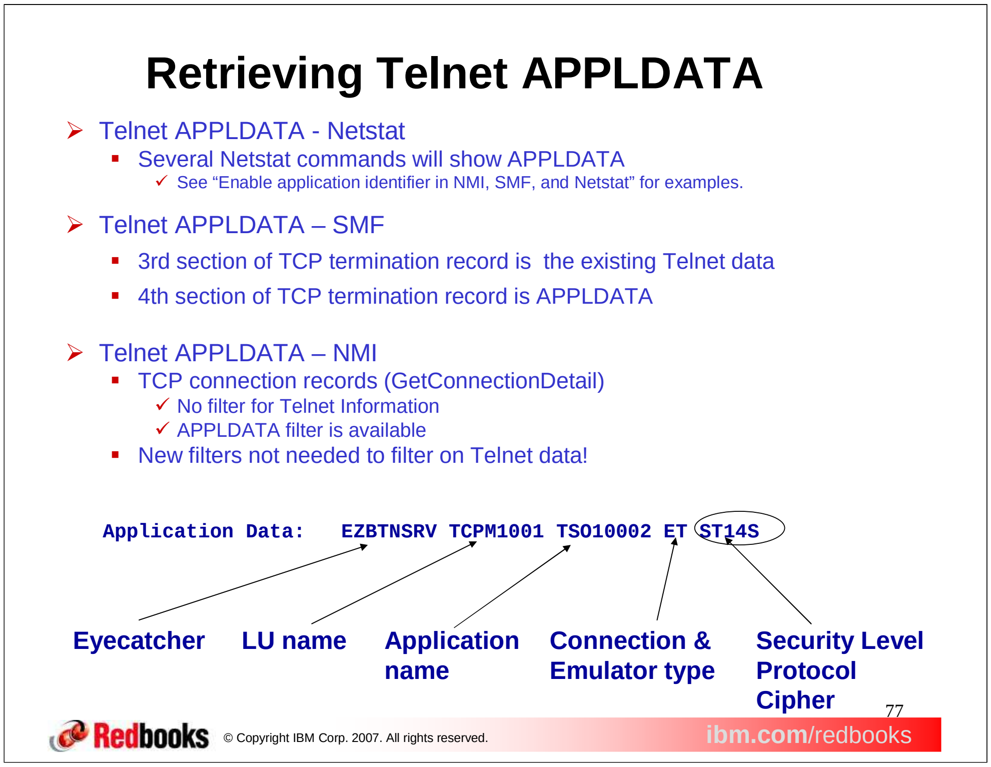
Redbooks ibm.com/redbooks

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 we have completed System SSL handshake and Telnet negotiations. We know what the security level, protocol, and cipher are and we know what the connection mode and emulator type are. If the connection mode is TN3270E we also know the LU name assigned to the connection. 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!

```
Application Data:   EZBTNSRV TCPM1001 TSO10002 ET ST14S
```

77

ibm.com/redbooks

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.

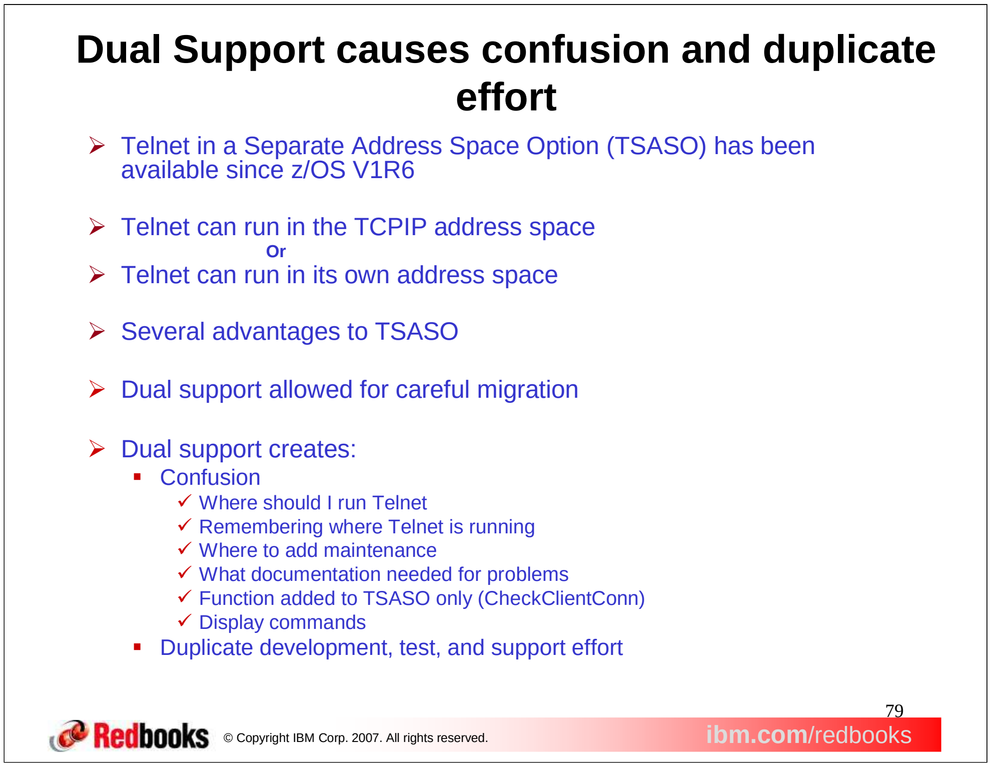
The TCP connection termination SMF record reports the Telnet-unique data section and the new APPLDATA section. The 3rd section is the Telnet-unique data and the 4th section is the APPLDATA section. The 4th section includes the same information as the 3rd section with the exception of logmode, user ID,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

Telnet must run in its own address space

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



Dual Support causes confusion and duplicate effort

- ➢ 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
- ➢ Dual support creates:
  - Confusion
    - ✓ Where should I 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

79

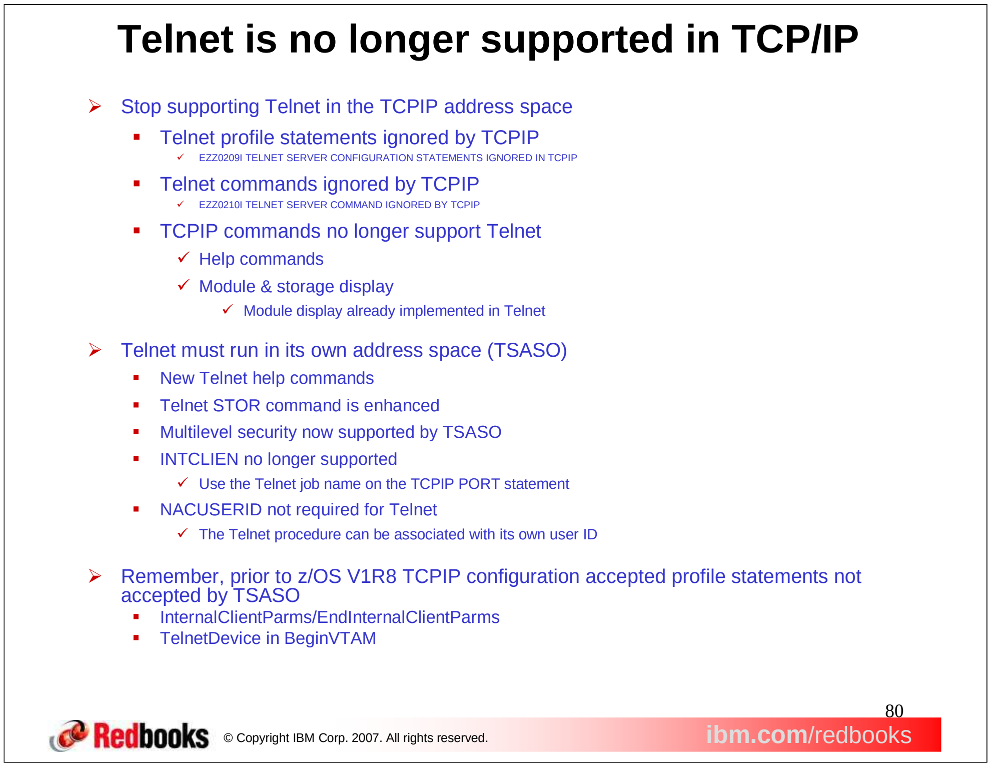
Redbooks ibm.com/redbooks

Telnet has been able to run in its own address space since z/OS V1R6 which was generally available in September, 2004. Since that time, users 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 up to 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 users will move Telnet to realize the TSASO advantages.

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.



Telnet is no longer supported in TCP/IP

- Stop supporting Telnet in the TCPIP address space
  - 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
- Telnet must run in its own address space (TSASO)
  - New Telnet help commands
  - Telnet STOR command is enhanced
  - Multilevel security now supported by TSASO
  - 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
- Remember, prior to z/OS V1R8 TCPIP configuration accepted profile statements not accepted by TSASO
  - InternalClientParms/EndInternalClientParms
  - TelnetDevice in BeginVTAM

80

Redbooks ibm.com/redbooks

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 users who use Telnet heavily have already switched to TSASO to isolate Telnet from TCP/IP. 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.

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.

If you haven't migrated yet see, z/OS Communications Server IP Configuration Guide, Accessing Remote hosts using Telnet - Telnet in its own address space, for setup details.

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.

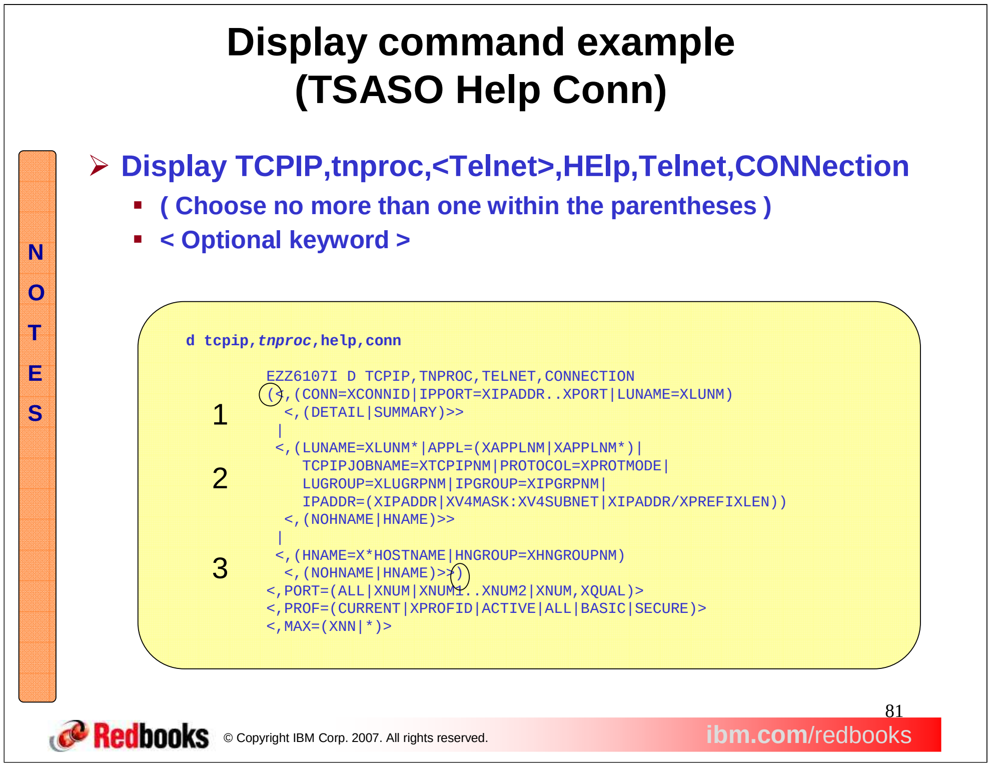
The TCPIP STOR command has two functions. It can display the load module name that contains the specified module name along with the module address and the first 48 bytes of storage. If no module name is specified, the command displays the current and maximum storage usage for the TCPIP address space 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 z/OS V1R9.

Multilevel security for Telnet required that Telnet run in the TCPIP address space because Telnet, in its own address space, 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 Comm Svr: Configuration Guide under "Accessing Remote Hosts using Telnet".

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, prior to 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.



Display command example (TSASO Help Conn)

NOTES

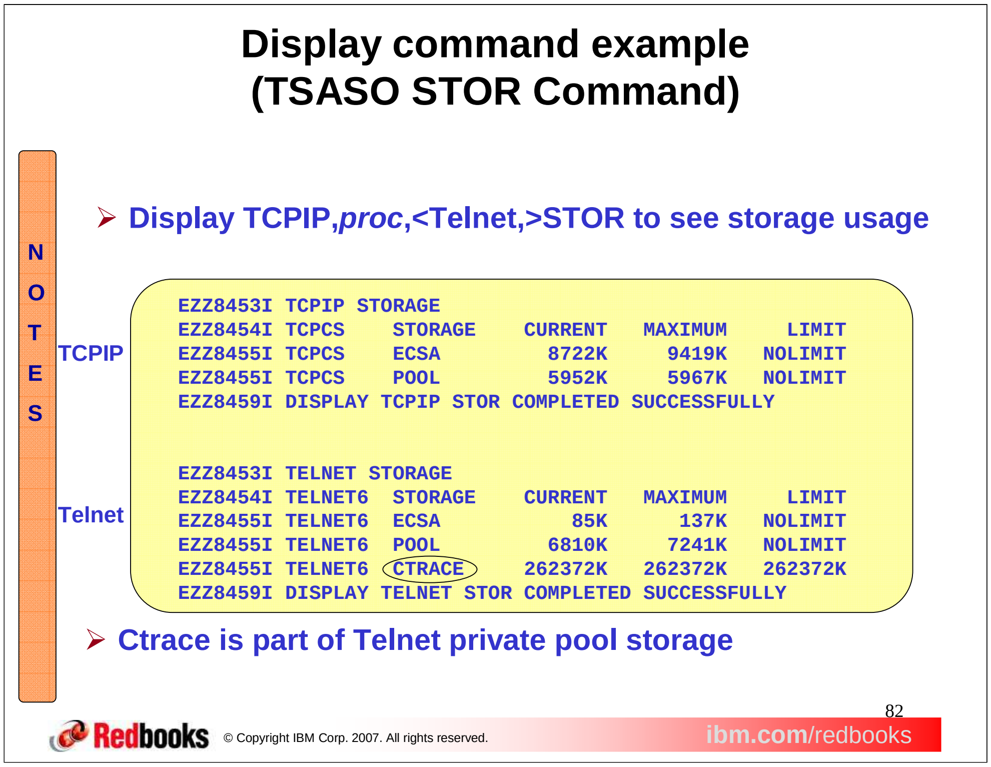
- **Display TCPIP,tnproc,<Telnet>,HElp,Telnet,CONNection**
  - **(Choose no more than one within the parentheses)**
  - **< Optional keyword >**

```
d tcpip,tnproc,help,conn

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

81

Redbooks ibm.com/redbooks

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

```
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
```

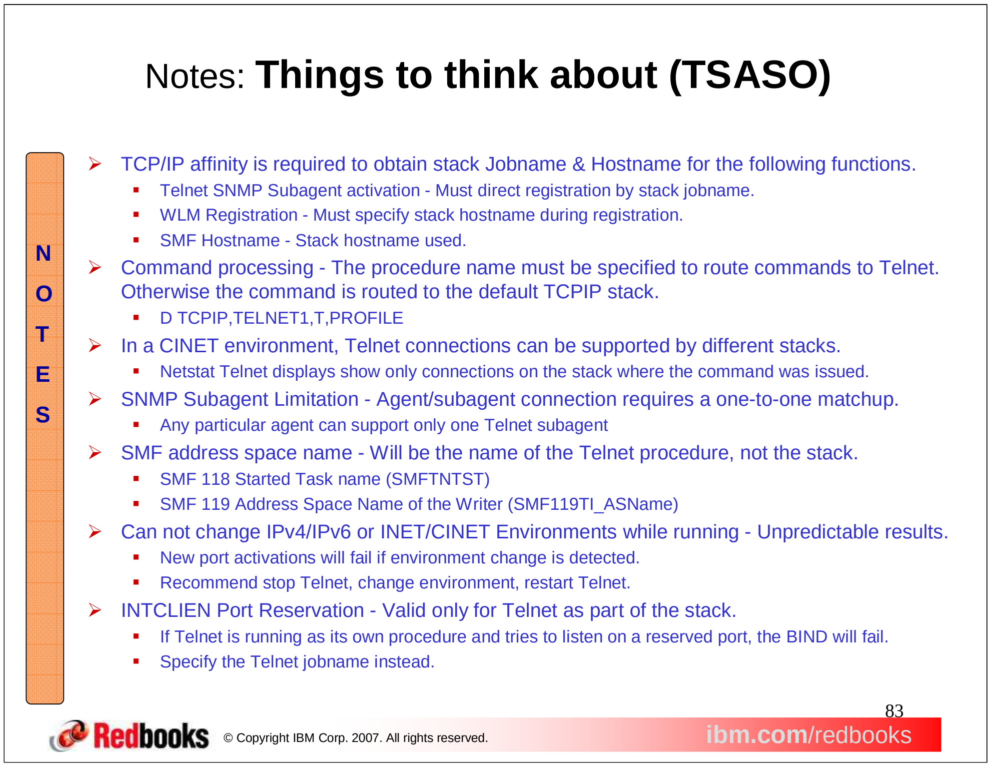
Telnet

```
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
```

- 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.

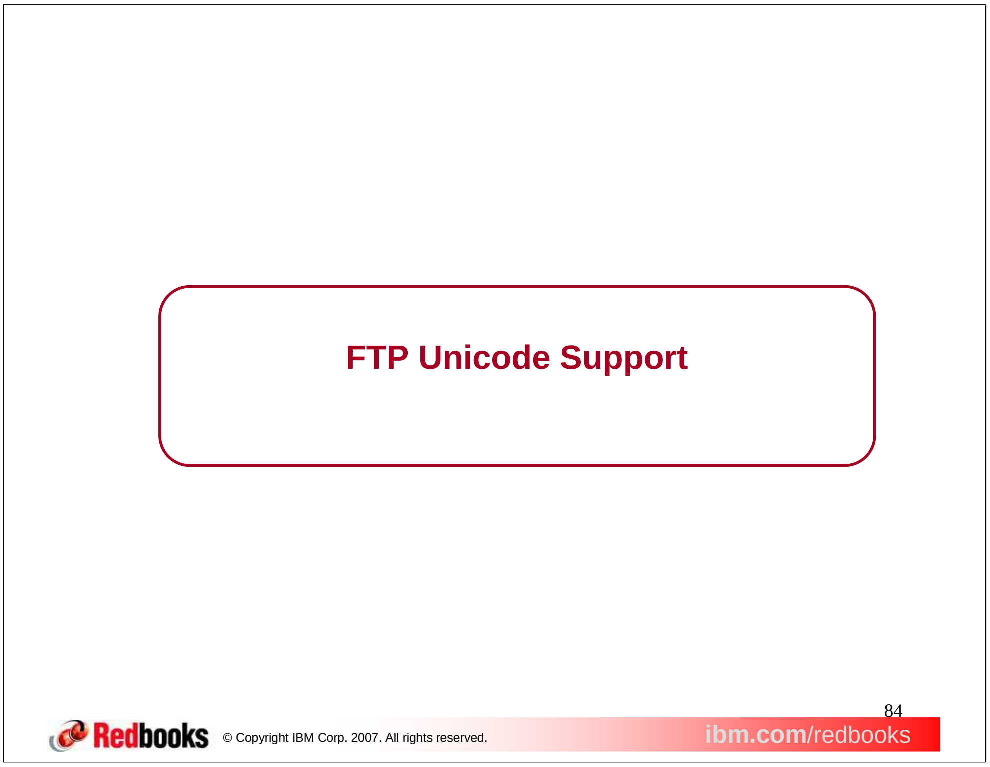


Notes: **Things to think about (TSASO)**

N O T E S

- TCP/IP affinity is required to obtain stack Jobname & Hostname for the following functions.
  - Telnet SNMP Subagent activation - Must direct registration by stack jobname.
  - WLM Registration - Must specify stack hostname during registration.
  - SMF Hostname - Stack hostname 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 matchup.
  - 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 Comm Svr: Configuration Guide, “Accessing Remote Hosts using Telnet”.

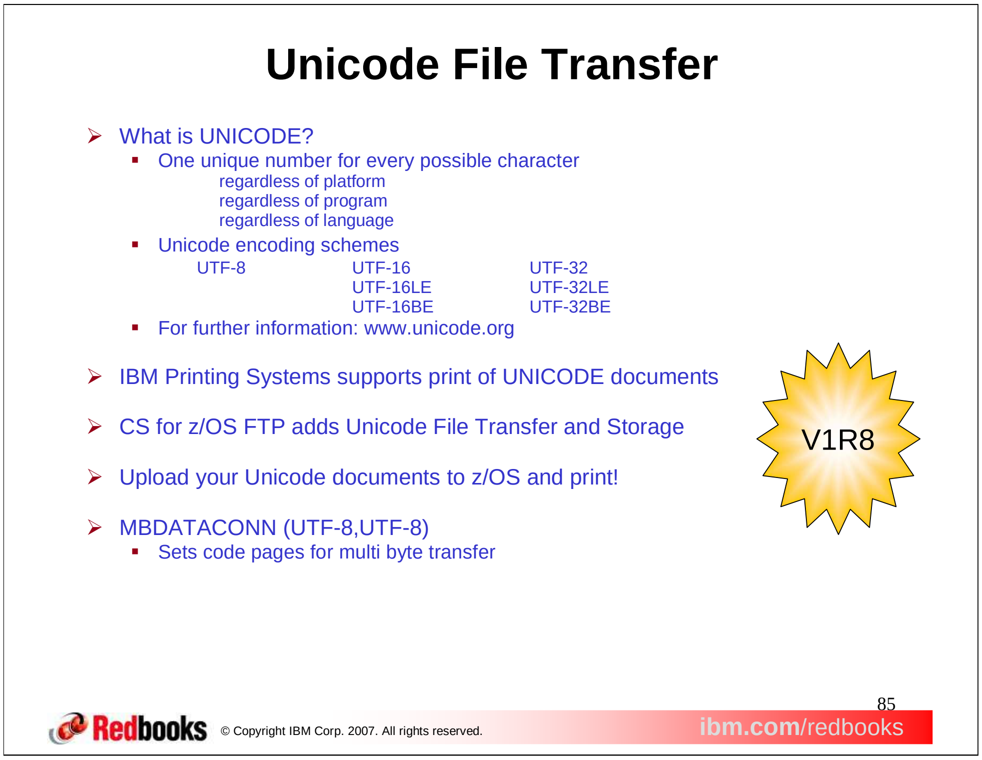


FTP Unicode Support

84

ibm.com/redbooks

This section describes the enhancements made in z/OS V1R9 to z/OS FTP Unicode file transfer and storage.



Unicode File Transfer

- What is UNICODE?
  - One unique number for every possible character
    - regardless of platform
    - regardless of program
    - regardless of language
  - Unicode encoding schemes

UTF-8	UTF-16	UTF-32
	UTF-16LE	UTF-32LE
	UTF-16BE	UTF-32BE

  - For further information: www.unicode.org
- IBM Printing Systems supports print of UNICODE documents
- CS for z/OS FTP adds Unicode File Transfer and Storage
- Upload your Unicode documents to z/OS and print!
- MBDATACONN (UTF-8,UTF-8)
  - Sets code pages for multi byte transfer

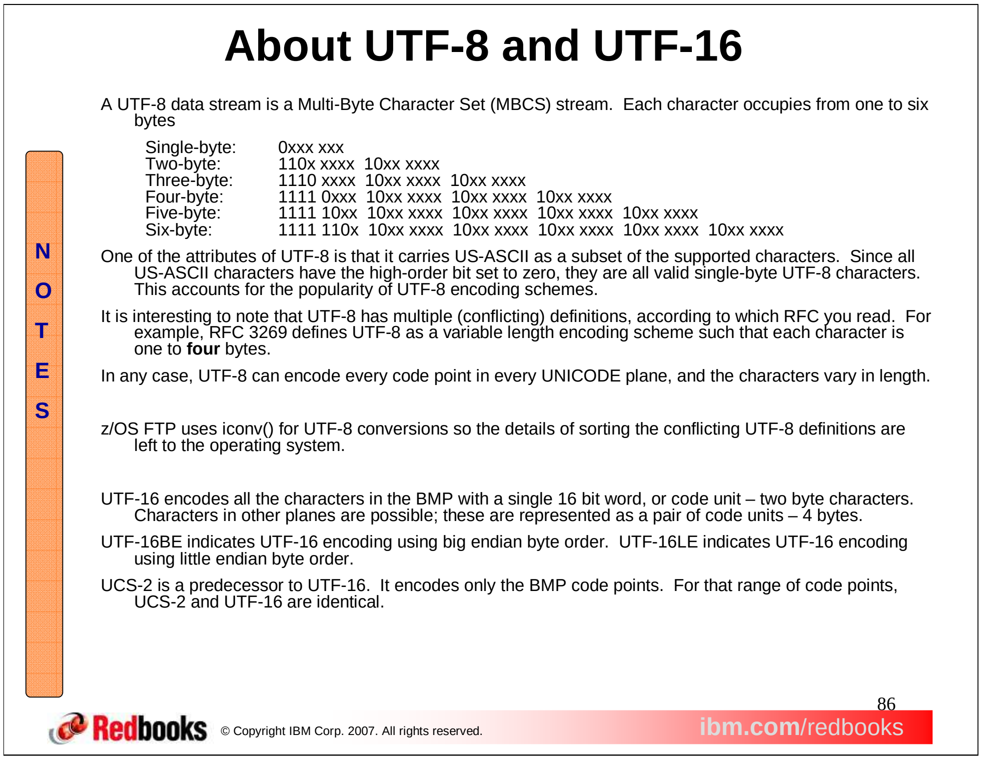
85

Unicode is defined by the Unicode Consortium. Their goal is to is to define encoding schemes that have the ability to encode every possible character in the universe. The URL in this slide is the web page of the Unicode Consortium. It is an excellent resource for learning about Unicode.

In V1R8, to support IBM Printing System's new support for UNICODE documents, CS for z/OS added Unicode File Transfer and storage. You can now move UNICODE documents to a z/OS host to store and to print.

In V1R8, z/OS FTP enhanced the configuration option, MBDATACONN, to support Unicode file transfer and storage. The MBDATACONN configuration option is used to specify which code pages to use for multi byte transfer. The first code page is the file storage code page; the second is the network transfer code page. These code pages are supported in pairs; the *IP Configuration Reference*, MBDATACONN statement, lists the code page pairs that are supported for multi byte transfer. In V1R8 we added the pair (UTF-8,UTF-8). This means that you can transfer and store a UTF-8 file. UTF-8 is one of the Unicode encodings.

Your notes have some details about the different Unicode encoding schemes.



About UTF-8 and UTF-16

NOTES

A UTF-8 data stream is a Multi-Byte Character Set (MBCS) stream. Each character occupies from one to six bytes

Single-byte:	0xxx xxx
Two-byte:	110x xxxx 10xx xxxx
Three-byte:	1110 xxxx 10xx xxxx 10xx xxxx
Four-byte:	1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx
Five-byte:	1111 10xx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx
Six-byte:	1111 110x 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx

One of the attributes of UTF-8 is that it carries US-ASCII as a subset of the supported characters. Since all US-ASCII characters have the high-order bit set to zero, they are all valid single-byte UTF-8 characters. This accounts for the popularity of UTF-8 encoding schemes.

It is interesting to note that UTF-8 has multiple (conflicting) definitions, according to which RFC you read. For example, RFC 3269 defines UTF-8 as a variable length encoding scheme such that each character is one to **four** bytes.

In any case, UTF-8 can encode every code point in every UNICODE plane, and the characters vary in length.

z/OS FTP uses iconv() for UTF-8 conversions so the details of sorting the conflicting UTF-8 definitions are left to the operating system.

UTF-16 encodes all the characters in the BMP with a single 16 bit word, or code unit – two byte characters. Characters in other planes are possible; these are represented as a pair of code units – 4 bytes.

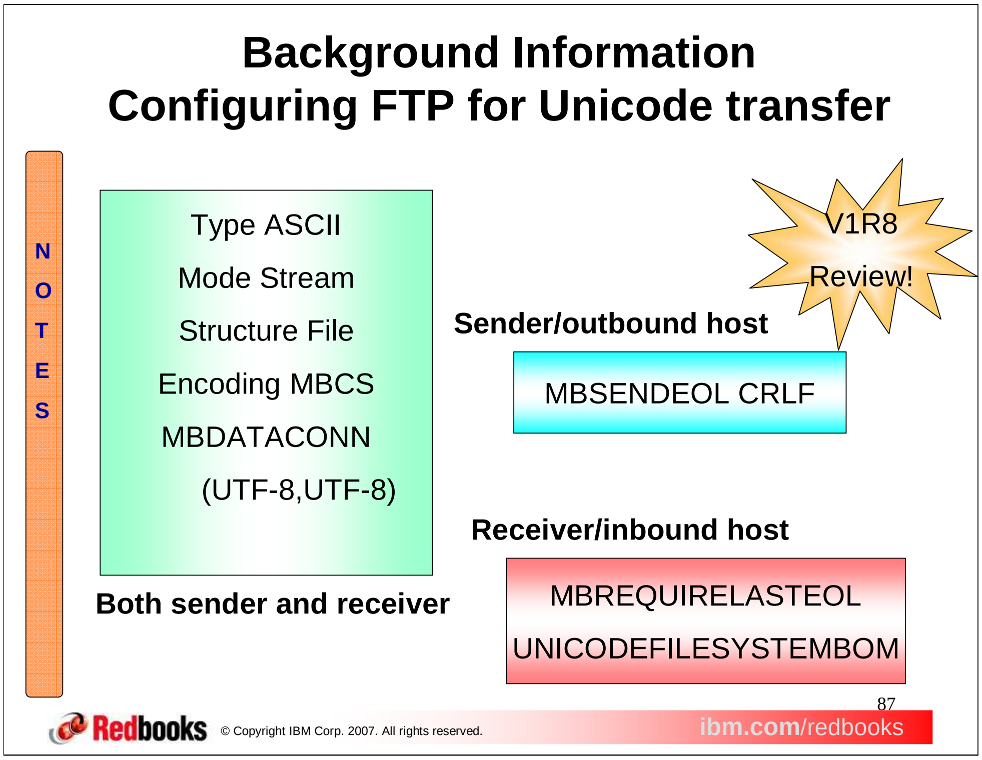
UTF-16BE indicates UTF-16 encoding using big endian byte order. UTF-16LE indicates UTF-16 encoding using little endian byte order.

UCS-2 is a predecessor to UTF-16. It encodes only the BMP code points. For that range of code points, UCS-2 and UTF-16 are identical.

86

Redbooks ibm.com/redbooks

These notes provide more information about the UTF-8 and UTF-16 encodings.



Background Information
Configuring FTP for Unicode transfer

N
O
T
E
S

Type ASCII

Mode Stream

Structure File

Encoding MBCS

MBDATACONN

(UTF-8,UTF-8)

Both sender and receiver

V1R8
Review!

Sender/outbound host

MBSENDEOL CRLF

Receiver/inbound host

MBREQUIRELASTEOL

UNICODEFILESYSTEMBOM

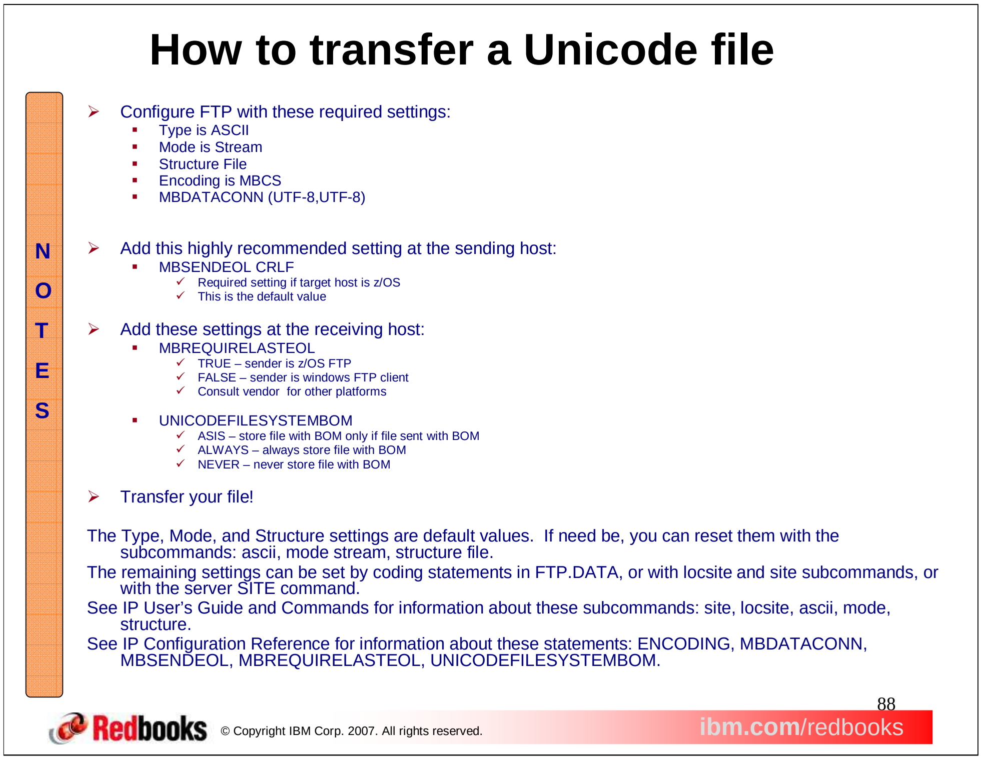
Here's a quick reminder of how we learned to configure FTP to transfer a Unicode file using the V1R8 FTP support for Unicode File Transfer and Storage.

The box on the left in green shows the configuration settings that must be set on both the sending and receiving hosts; the cyan box on the upper right shows the configuration settings that must be set at the sending host; the red box on the lower right shows the configuration options that must be set at the receiving host.

MBSENDEOL was introduced in V1R7, and applies only to outbound multi byte transfer. Although alternate settings for MBSENDEOL are available, most users should stick with CRLF, the default value and the value specified in the FTP standard, RFC 959. This is the only setting that works when sending to a z/OS host.

MBREQUIRELASTEOL and UNICODEFILESYSTEMBOM were introduced in V1R8, and apply only to inbound multi byte file transfer. You have to understand the sending host's FTP implementation to set MBREQUIRELASTEOL correctly; the file transfer will fail if you pick the wrong value. That's the bad news; the good news is that you can toggle to the other setting and try again if you get it wrong, or you can consult the notes on the next page for suggested values.

You can set UNICODEFILESYSTEMBOM to whatever you would like. The setting won't affect the success or failure of the file transfer, but may impact the end user of the file.



How to transfer a Unicode file

N O T E S

- Configure FTP with these required settings:
  - Type is ASCII
  - Mode is Stream
  - Structure File
  - Encoding is MBCS
  - MBDATACONN (UTF-8,UTF-8)
- Add this highly recommended setting at the sending host:
  - MBSENDEOL CRLF
    - ✓ Required setting if target host is z/OS
    - ✓ This is the default value
- Add these settings at the receiving host:
  - MBREQUIRELASTEOL
    - ✓ TRUE – sender is z/OS FTP
    - ✓ FALSE – sender is windows FTP client
    - ✓ Consult vendor for other platforms
  - UNICODEFILESYSTEMBOM
    - ✓ ASIS – store file with BOM only if file sent with BOM
    - ✓ ALWAYS – always store file with BOM
    - ✓ NEVER – never store file with BOM
- Transfer your file!

The Type, Mode, and Structure settings are default values. If need be, you can reset them with the subcommands: ascii, mode stream, structure file.

The remaining settings can be set by coding statements in FTP.DATA, or with locsite and site subcommands, or with the server SITE command.

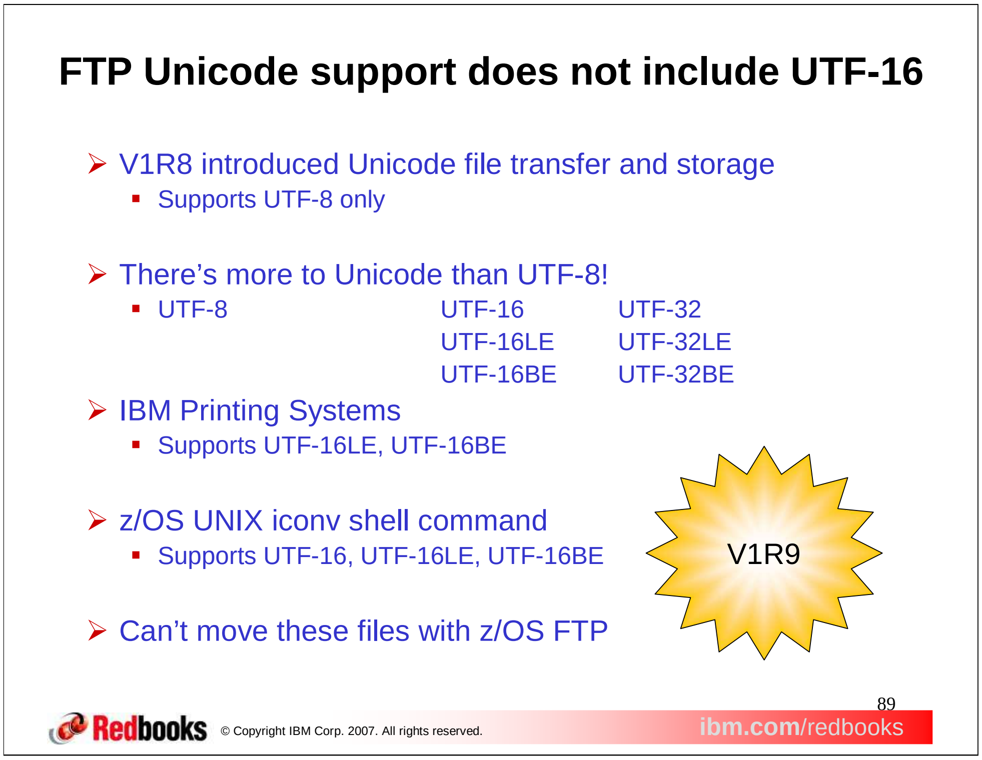
See IP User's Guide and Commands for information about these subcommands: site, locsite, ascii, mode, structure.

See IP Configuration Reference for information about these statements: ENCODING, MBDATACONN, MBSENDEOL, MBREQUIRELASTEOL, UNICODEFILESYSTEMBOM.

88

Redbooks ibm.com/redbooks

These notes provide a checklist for transferring a Unicode file, and resources for more information.



FTP Unicode support does not include UTF-16

- V1R8 introduced Unicode file transfer and storage
  - Supports UTF-8 only
- There’s more to Unicode than UTF-8!

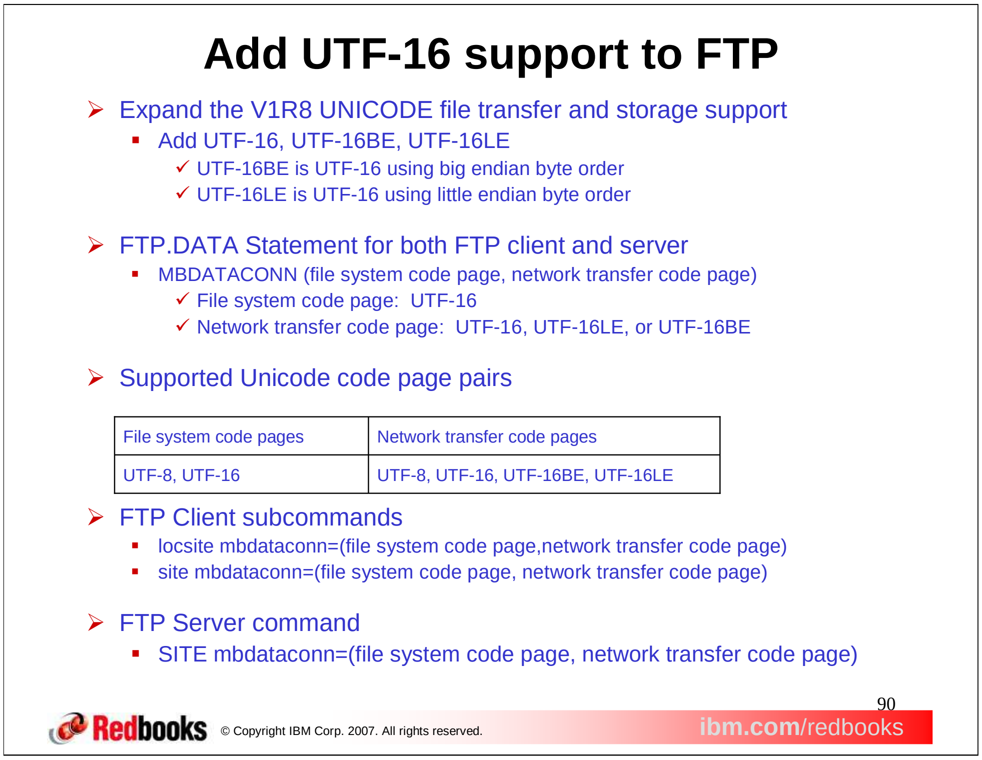
UTF-8	UTF-16	UTF-32
	UTF-16LE	UTF-32LE
	UTF-16BE	UTF-32BE

- IBM Printing Systems
  - Supports UTF-16LE, UTF-16BE
- z/OS UNIX iconv shell command
  - Supports UTF-16, UTF-16LE, UTF-16BE
- Can’t move these files with z/OS FTP

In V1R8, z/OS FTP supported Unicode file transfer and storage, but the only Unicode encoding supported was UTF-8. However, there is more to Unicode than UTF-8! The Unicode Consortium has defined the encodings listed here.

The z/OS platform has started making use of the UTF-16 class of encodings. Here we list two z/OS exploiters of UTF-16 encodings.

The problem is that you can’t move these Unicode files with z/OS FTP.



Add UTF-16 support to FTP

- Expand the V1R8 UNICODE file transfer and storage support
  - Add UTF-16, UTF-16BE, UTF-16LE
    - ✓ UTF-16BE is UTF-16 using big endian byte order
    - ✓ UTF-16LE is UTF-16 using little endian byte order
- FTP.DATA Statement for both FTP client and server
  - MBDATACONN (file system code page, network transfer code page)
    - ✓ File system code page: UTF-16
    - ✓ Network transfer code page: UTF-16, UTF-16LE, or UTF-16BE
- Supported Unicode code page pairs

File system code pages	Network transfer code pages
UTF-8, UTF-16	UTF-8, UTF-16, UTF-16BE, UTF-16LE

- FTP Client subcommands
  - locsite mbdataconn=(file system code page,network transfer code page)
  - site mbdataconn=(file system code page, network transfer code page)
- FTP Server command
  - SITE mbdataconn=(file system code page, network transfer code page)

90

Redbooks ibm.com/redbooks

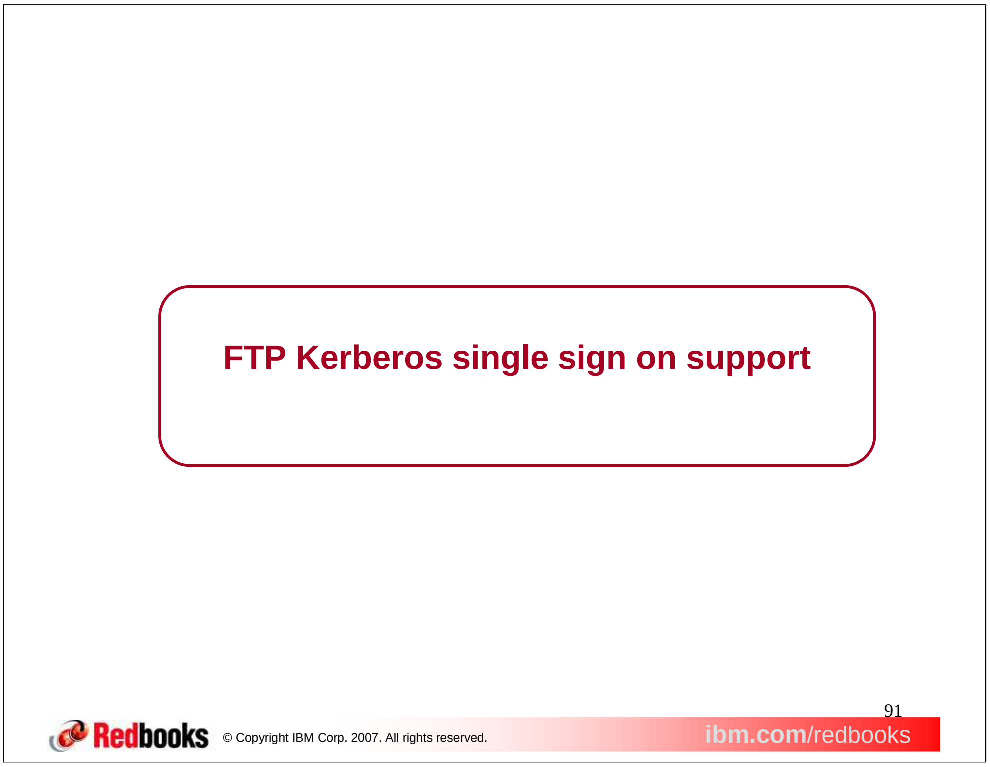
V1R9 builds upon the UNICODE support added in V1R8 by adding support for UTF-16. For practical purposes, UTF-16 uses two bytes per character (your notes discuss exceptions). A two byte character must use either little endian byte order or big endian byte order; therefore, UTF-16 is always either UTF-16BE or UTF-16LE. By definition, UTF-16 is UTF-16BE by default unless a BOM is present. Recall that MBDATACONN statement defines the code pages to use for multi byte file transfer, and can be specified for both the FTP client and server. For the file system code page, you can now specify UTF-16. FTP will always use UTF-16BE in the z/OS file system.

For the network transfer, you can now specify the UTF-16 encoding schemes above. FTP must be able to support either little endian or big endian encodings on the network because some platforms support only little endian encoding. FTP for z/OS will always assume that UTF-16 is equivalent to UTF-16BE (big endian UTF-16).

The chart summarizes all the MBDATACONN code page combinations supported for Unicode file transfer as of V1R9. Any choice from the **File system code pages** column can be specified with any choice from the **Network transfer code pages** column.

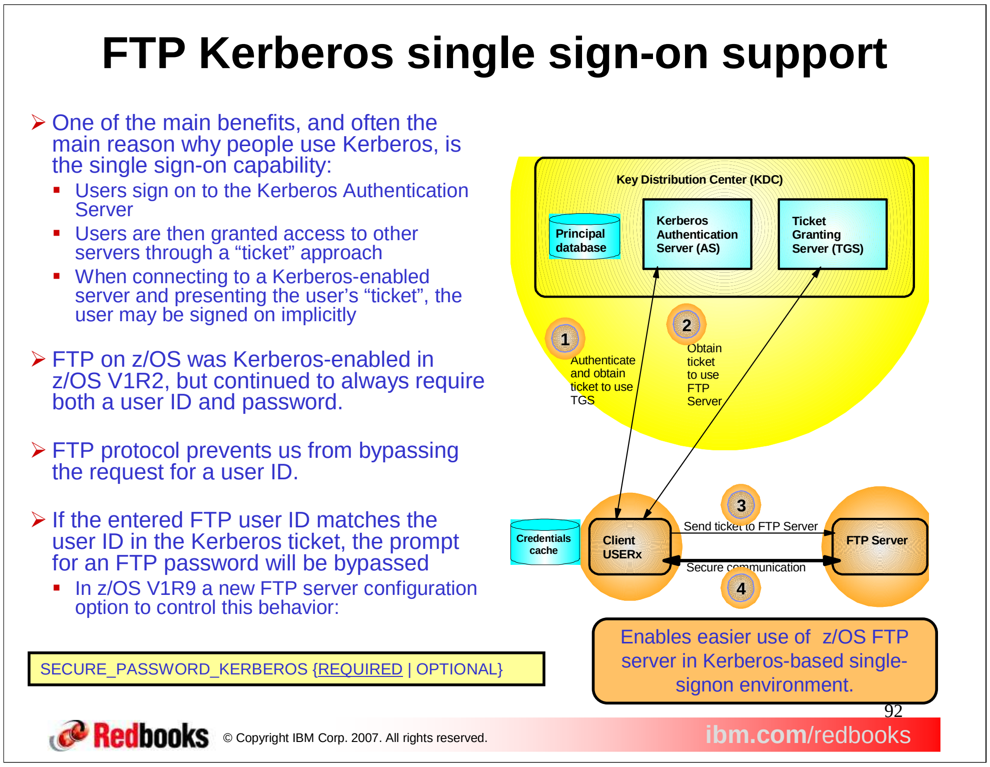
As well as using the FTP.DATA statement MBDATACONN, you can use locsite and site subcommands to configure the multibyte code pages. Code pages valid for the MBDATACONN FTP.DATA statement are valid for site and locsite subcommands. As a reminder, the locsite subcommand configures the FTP client; the site subcommand configures the FTP server.

The server SITE command parameter MBDATACONN defines code pages for multibyte transfer to the server, for the current login session. The MBDATACONN parameter is enhanced to accept the new UTF-16 code pages. The z/OS client sends a SITE command to the server for you when you use the site subcommand. If you log into the server using a different FTP client such as the windows client, you may have to use that client’s QUOTE subcommand to send a SITE command to the server.



FTP Kerberos single sign on support

This section covers an extension to the Kerberos support which enables connections to the z/OS FTP server using Kerberos without requiring the FTP client to supply the user’s password.



FTP Kerberos single sign-on support

- One of the main benefits, and often the main reason why people use Kerberos, is the single sign-on capability:
  - Users sign on to the Kerberos Authentication Server
  - Users are then granted access to other servers through a “ticket” approach
  - When connecting to a Kerberos-enabled server and presenting the user’s “ticket”, the user may be signed on implicitly
- FTP on z/OS was Kerberos-enabled in z/OS V1R2, but continued to always require both a user ID and password.
- FTP protocol prevents us from bypassing the request for a user ID.
- If the entered FTP user ID matches the user ID in the Kerberos ticket, the prompt for an FTP password will be bypassed
  - In z/OS V1R9 a new FTP server configuration option to control this behavior:

```
SECURE_PASSWORD_KERBEROS {REQUIRED | OPTIONAL}
```

Key Distribution Center (KDC)
Principal database
Kerberos Authentication Server (AS)
Ticket Granting Server (TGS)
1
Authenticate and obtain ticket to use TGS
2
Obtain ticket to use FTP Server
Credentials cache
Client USERx
3
Send ticket to FTP Server
FTP Server
Secure communication
4

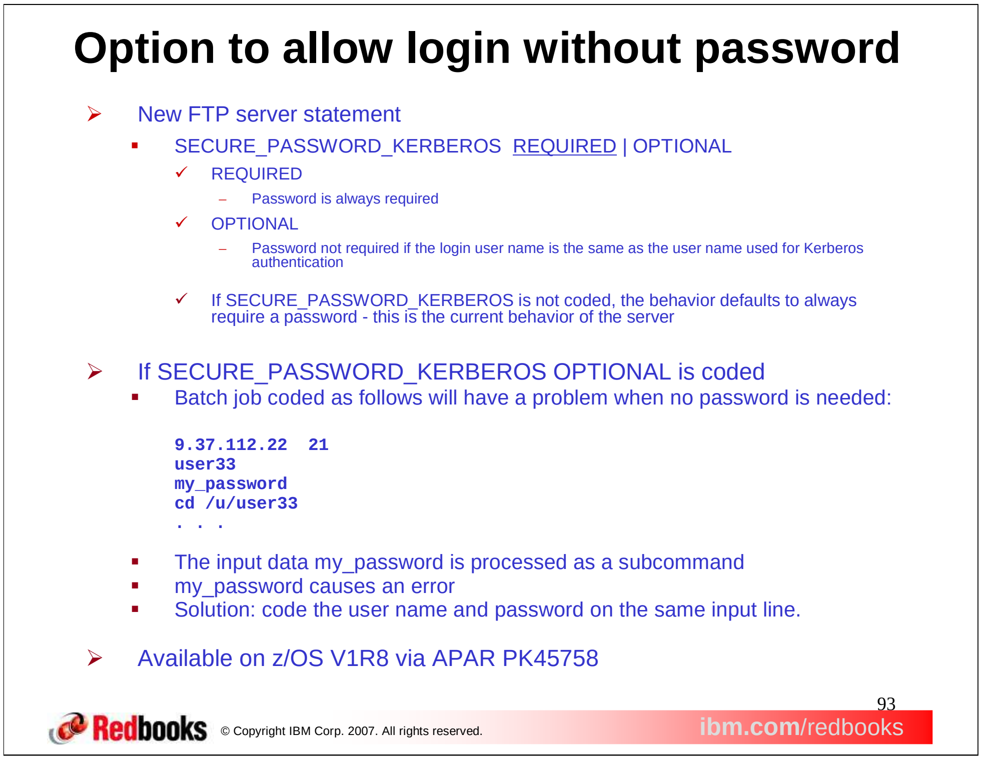
Enables easier use of z/OS FTP server in Kerberos-based single-signon environment.

92

In a Kerberos environment, users must authenticate to the Kerberos Key Distribution Center (KDC) by supplying their user name and password. Users are also accustomed to using single sign-on support. The user authenticates once to the Kerberos KDC and then should be able to access and be authenticated by other services without having to enter their password again. However, if they then login to a Kerberos enabled z/OS FTP server, they must enter their user name and password again.

The solution to the problem is to allow users to login to the z/OS FTP server without having to re-enter the password. First, the user must authenticate to the Kerberos KDC. Then the user starts the FTP client and connects to the z/OS FTP server using GSSAPI authentication. GSSAPI, or Generic Security Service Application Programming Interface, is the authentication method used by the FTP protocol to allow connections between Kerberos enabled clients and servers.

The FTP protocol still requires that the client supply a user name to the FTP server. If the user name supplied to the z/OS FTP server is the same user name used to authenticate to the Kerberos KDC, the z/OS FTP server will not prompt for the password.



Option to allow login without password

- ➢ New FTP server statement
  - ▪ SECURE_PASSWORD_KERBEROS REQUIRED | OPTIONAL
    - ✓ REQUIRED
      - – Password is always required
    - ✓ OPTIONAL
      - – Password not required if the login user name is the same as the user name used for Kerberos authentication
    - ✓ If SECURE_PASSWORD_KERBEROS is not coded, the behavior defaults to always require a password - this is the current behavior of the server
- ➢ If SECURE_PASSWORD_KERBEROS OPTIONAL is coded
  - ▪ Batch job coded as follows will have a problem when no password is needed:

```
9.37.112.22  21
user33
my_password
cd /u/user33
. . .
```

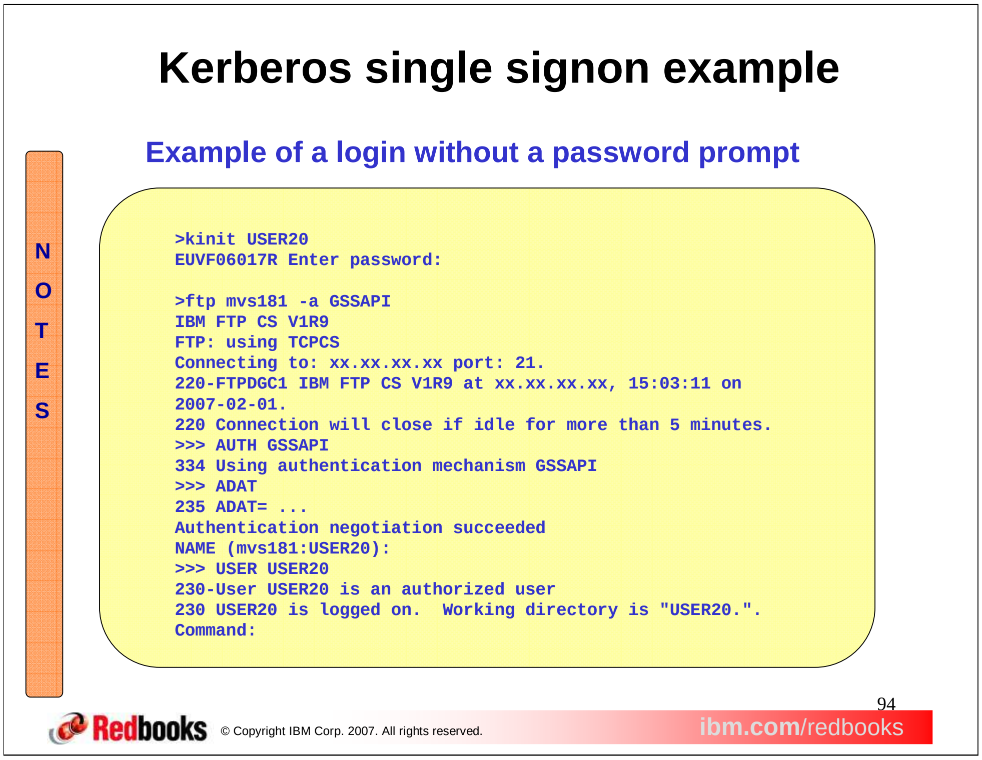
  - ▪ The input data my_password is processed as a subcommand
  - ▪ my_password causes an error
  - ▪ Solution: code the user name and password on the same input line.
- ➢ Available on z/OS V1R8 via APAR PK45758

93

Redbooks ibm.com/redbooks

A new statement, SECURE_PASSWORD_KERBEROS, has been added to the FTP.DATA file. It has two values, REQUIRED and OPTIONAL. If REQUIRED is coded, the z/OS FTP server will always prompt for the user's password. If OPTIONAL is coded and the user name used to log into the z/OS FTP server is the same as the user name used to authenticate to the Kerberos KDC, the z/OS FTP server will not require the user's password and the user will be logged in. If OPTIONAL is coded and the user name used to log into the z/OS FTP server is not the same as the user name used to authenticate to the Kerberos KDC, the z/OS FTP server will require the user's password before the user can be logged in. The default value is REQUIRED.

Batch jobs may have to be updated if this function is enabled. If the batch job specifies the user name and the password on separate lines and SECURE_PASSWORD_KERBEROS OPTIONAL is coded, the batch job may incorrectly supply the password when the server does not prompt for the password. The server will reject the password since the user will already be logged in. To avoid this problem, the batch job can be changed to code the user name and password on the same line.



Kerberos single signon example

Example of a login without a password prompt

N O T E S

```
>kinit USER20
EUVF06017R Enter password:

>ftp mvs181 -a GSSAPI
IBM FTP CS V1R9
FTP: using TCPCS
Connecting to: xx.xx.xx.xx port: 21.
220-FTPDGC1 IBM FTP CS V1R9 at xx.xx.xx.xx, 15:03:11 on
2007-02-01.
220 Connection will close if idle for more than 5 minutes.
>>> AUTH GSSAPI
334 Using authentication mechanism GSSAPI
>>> ADAT
235 ADAT= ...
Authentication negotiation succeeded
NAME (mvs181:USER20):
>>> USER USER20
230-User USER20 is an authorized user
230 USER20 is logged on.  Working directory is "USER20.".
Command:
```

94

Redbooks ibm.com/redbooks

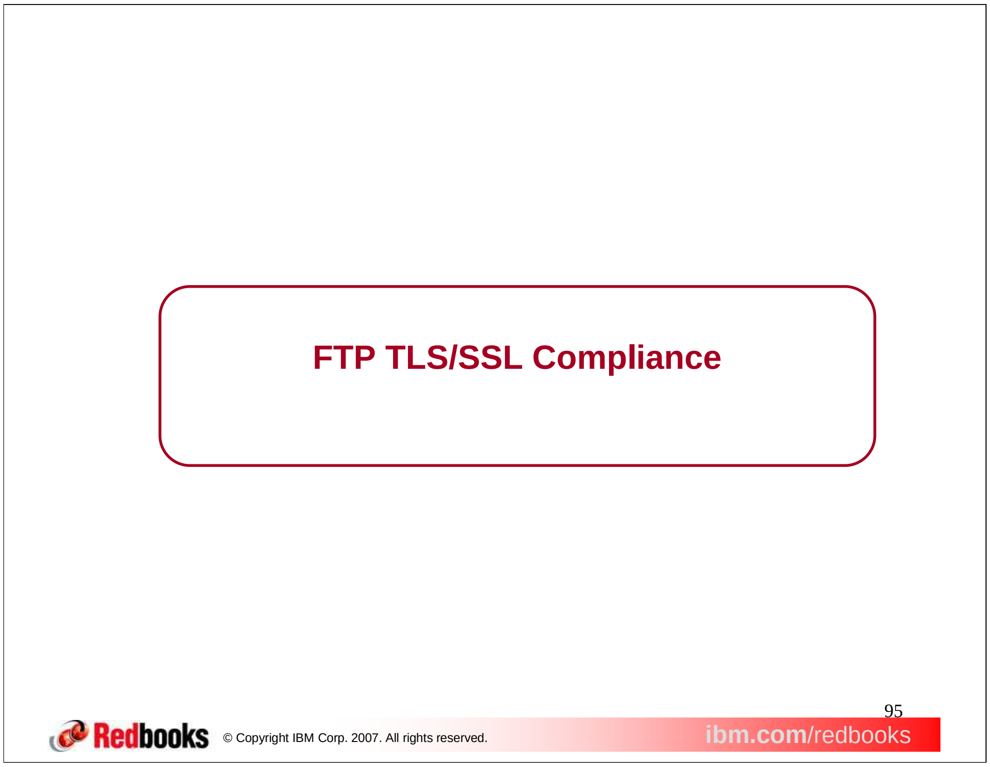
This is an example of an FTP client connecting to a z/OS FTP server which has enabled single sign on support by specifying SECURE_PASSWORD_KERBEROS OPTIONAL in the server’s FTP.DATA file.

First, the user must authenticate to the Kerberos KDC. This is done by using the kinit application. In this case the user authenticates to Kerberos as USER20.

Next the client connects to the server requesting GSSAPI authentication.

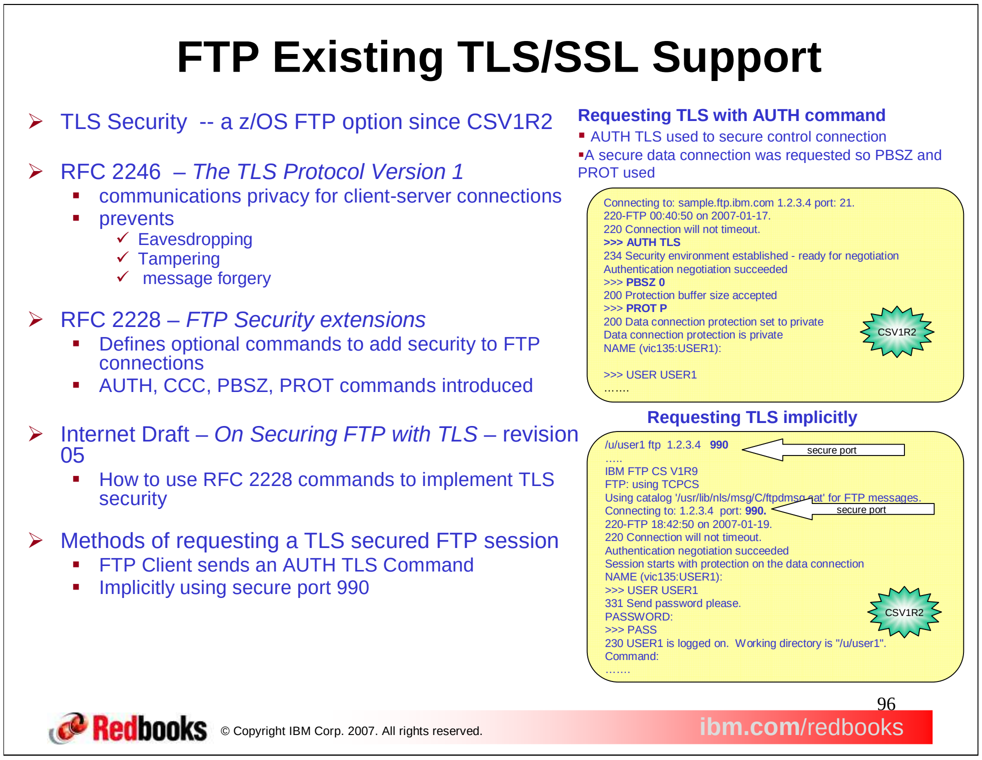
When the GSSAPI authentication is successful, the prompt for the user name is issued. The client supplies the user name, which again in this case is USER20.

Since the user name is the same as the user name which was previously authenticated by Kerberos on the kinit, the FTP server does not require the password, and the user is successfully logged in.



FTP TLS/SSL Compliance

This section describes enhancements to z/OS FTP to comply with RFCs regarding TLS security.



FTP Existing TLS/SSL Support

- TLS Security -- a z/OS FTP option since CSV1R2
- RFC 2246 – *The TLS Protocol Version 1*
  - communications privacy for client-server connections
  - prevents
    - ✓ Eavesdropping
    - ✓ Tampering
    - ✓ message forgery
- RFC 2228 – *FTP Security extensions*
  - Defines optional commands to add security to FTP connections
  - AUTH, CCC, PBSZ, PROT commands introduced
- Internet Draft – *On Securing FTP with TLS* – revision 05
  - How to use RFC 2228 commands to implement TLS security
- Methods of requesting a TLS secured FTP session
  - FTP Client sends an AUTH TLS Command
  - Implicitly using secure port 990

Requesting TLS with AUTH command

- AUTH TLS used to secure control connection
- A secure data connection was requested so PBSZ and PROT used

```
Connecting to: sample.ftp.ibm.com 1.2.3.4 port: 21.
220-FTP 00:40:50 on 2007-01-17.
220 Connection will not timeout.
>>> AUTH TLS
234 Security environment established - ready for negotiation
Authentication negotiation succeeded
>>> PBSZ 0
200 Protection buffer size accepted
>>> PROT P
200 Data connection protection set to private
Data connection protection is private
NAME (vic135:USER1):

>>> USER USER1
…….
```

Requesting TLS implicitly

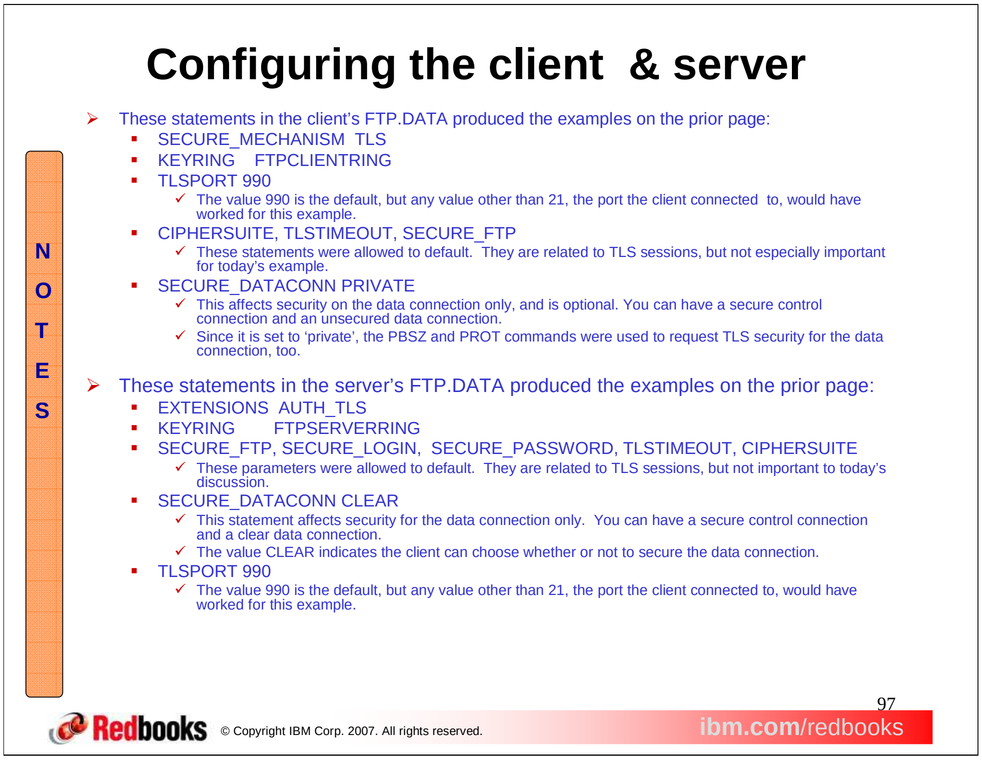
```
/u/user1 ftp  1.2.3.4   990
…..
IBM FTP CS V1R9
FTP: using TCPCS
Using catalog '/usr/lib/nls/msg/C/ftpdmsg.cat' for FTP messages.
Connecting to: 1.2.3.4  port: 990.
220-FTP 18:42:50 on 2007-01-19.
220 Connection will not timeout.
Authentication negotiation succeeded
Session starts with protection on the data connection
NAME (vic135:USER1):
>>> USER USER1
331 Send password please.
PASSWORD:
>>> PASS
230 USER1 is logged on.  Working directory is "/u/user1".
Command:
…….
```

In CSV1R2, z/OS FTP implemented TLS security based on the RFCs and Internet Draft listed. RFC 2246 defines TLS – Transport Layer Security. This is from the abstract: The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

File Transfer Protocols as defined in RFC 959 do not provide a protocol for requesting a secure session. RFC 2228 defines FTP commands to request a secure session, albeit not necessarily a TLS secure session. In this presentation, we will be concerned with the AUTH, CCC, PBSZ and PROT commands which are defined in that RFC. The Internet Draft, *On Securing FTP with TLS*, defines how the commands introduced in RFC 2228 should be used to request and maintain a TLS secured FTP session. The z/OS FTP TLS support is based on draft 05 of *On Securing FTP with TLS.*

One method of requesting a TLS secured FTP session is for the FTP client to send an AUTH TLS command to the server as illustrated in the first example. This establishes TLS security for the control connection. Securing the data connection is optional. In this case, the client was configured for a secure data connection, so the optional commands PBSZ and PROT were used to set up security for the data connection. The notes which follow this page describe the statements in the client's FTP.DATA that caused the client to request an FTP session, and also describe the related statements in the server's FTP.DATA.

The second example demonstrates how FTP establishes an **implicitly secure** connection. The Internet Draft, *On Securing FTP with TLS,* specified that connections to port 990 are assumed to be secure – no AUTH command is needed to secure the connection. In this example, we asked the ftp client to connect to port 990, the secure port. We used the same FTP.DATA sets for client and server as we used in the first example. In those FTP.DATA sets, TLSPORT was set to the default value, 990. The first line shows the ftp client being started with the hostname parameter of 1.2.3.4, and the optional port parameter of 990. Port 990 appears in **boldface** font. Contrasting this example with the prior example, notice the session is secure without the use of the AUTH, PBSZ, and PROT commands. The session is implicitly secured as opposed to explicitly securing the session with commands.



Configuring the client & server

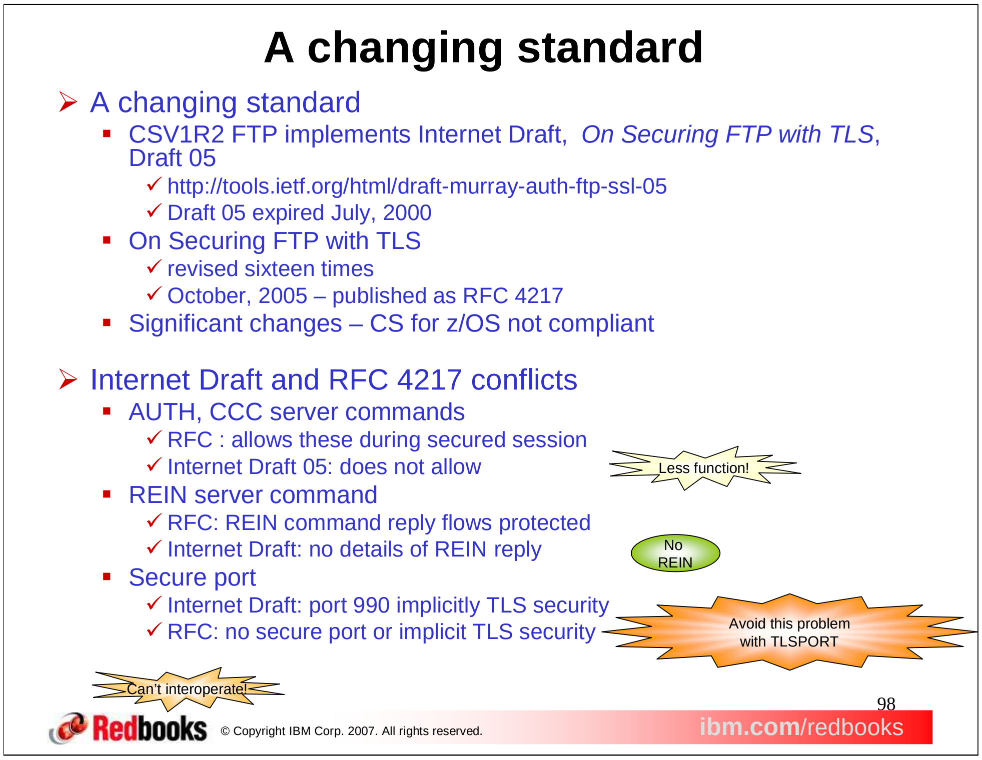
N O T E S

- These statements in the client's FTP.DATA produced the examples on the prior page:
  - SECURE_MECHANISM TLS
  - KEYRING   FTPCLIENTRING
  - TLSPORT 990
    - ✓ The value 990 is the default, but any value other than 21, the port the client connected to, would have worked for this example.
  - CIPHERSUITE, TLSTIMEOUT, SECURE_FTP
    - ✓ These statements were allowed to default. They are related to TLS sessions, but not especially important for today's example.
  - SECURE_DATACONN PRIVATE
    - ✓ This affects security on the data connection only, and is optional. You can have a secure control connection and an unsecured data connection.
    - ✓ Since it is set to 'private', the PBSZ and PROT commands were used to request TLS security for the data connection, too.
- These statements in the server's FTP.DATA produced the examples on the prior page:
  - EXTENSIONS AUTH_TLS
  - KEYRING   FTPSERVERRING
  - SECURE_FTP, SECURE_LOGIN, SECURE_PASSWORD, TLSTIMEOUT, CIPHERSUITE
    - ✓ These parameters were allowed to default. They are related to TLS sessions, but not important to today's discussion.
  - SECURE_DATACONN CLEAR
    - ✓ This statement affects security for the data connection only. You can have a secure control connection and a clear data connection.
    - ✓ The value CLEAR indicates the client can choose whether or not to secure the data connection.
  - TLSPORT 990
    - ✓ The value 990 is the default, but any value other than 21, the port the client connected to, would have worked for this example.

97

Redbooks ibm.com/redbooks

These notes describe how the FTP client and server were configured when executing the examples we have just seen of the client requesting a secure session using the AUTH command and an implicit TLS secured session.



A changing standard

- A changing standard
  - CSV1R2 FTP implements Internet Draft, *On Securing FTP with TLS*, Draft 05
    - ✓ http://tools.ietf.org/html/draft-murray-auth-ftp-ssl-05
    - ✓ Draft 05 expired July, 2000
  - On Securing FTP with TLS
    - ✓ revised sixteen times
    - ✓ October, 2005 – published as RFC 4217
  - Significant changes – CS for z/OS not compliant

- Internet Draft and RFC 4217 conflicts
  - AUTH, CCC server commands
    - ✓ RFC : allows these during secured session
    - ✓ Internet Draft 05: does not allow
  - REIN server command
    - ✓ RFC: REIN command reply flows protected
    - ✓ Internet Draft: no details of REIN reply
  - Secure port
    - ✓ Internet Draft: port 990 implicitly TLS security
    - ✓ RFC: no secure port or implicit TLS security

Less function!

No REIN

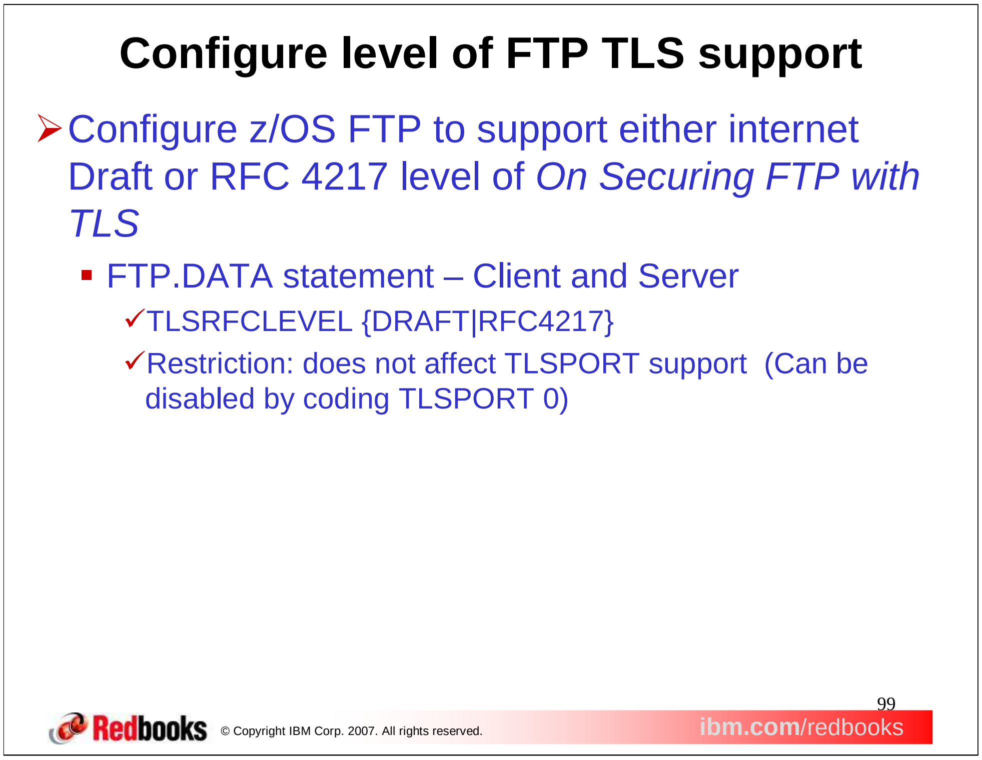
Avoid this problem with TLSPORT

Can't interoperate!

Since CSV1R2, Communications Server for z/OS FTP has supported TLS secured sessions. This support is based on revision 05 of the Internet Draft: *On Securing FTP with TLS.* The URL of revision 05 appears on this slide. The problem is that since the year 2000, the draft was revised eleven more times, and has now been published as an RFC. The CS for z/OS FTP support does not comply with the new standard, RFC 4217.

RFC 4217 has significant differences from the Internet Draft. The RFC is less restrictive than the draft about flowing the AUTH and CCC commands to the server during a secure session. The upshot of this is that the full RFC 4217 functionality of the AUTH and CCC commands is not available to z/OS FTP users. The RFC explicitly states that the REIN server command reply must flow on the protected connection – the server cannot clear the session before sending the reply. The Internet Draft did not specify this level of detail. The z/OS server implementation does not send the reply while the session is still protected; therefore, the z/OS FTP server does not interoperate with an RFC 4217 compliant FTP client when REIN is used during a TLS session. This is not as bad as you might think; REIN is not really recommended during an FTP session regardless of whether you are using TLS or not.

According to draft 05 of Securing FTP with TLS, when FTP clients connect to server port 990, the connection is secured with TLS without flowing an AUTH command – the connection is implicitly secured, as opposed to explicitly securing the connection by sending an AUTH command to the server. The RFC has dropped implicit security and secure port entirely. Thus, a connection between an RFC 4217 compliant FTP and an Internet Draft compliant FTP on the secure port cannot interoperate, because the Internet Draft side believes the connection is secure, and the RFC 4217 compliant side believes the connection is not secure. Again, this is not as bad as you might think. The existing TLSPORT statement for the client and server's FTP.DATA allows you to reassign the TLSPORT, or disable it altogether. Therefore, existing z/OS provides a bypass for this problem.



Configure level of FTP TLS support

- Configure z/OS FTP to support either internet Draft or RFC 4217 level of *On Securing FTP with TLS*
  - FTP.DATA statement – Client and Server
    - ✓TLSRFCLEVEL {DRAFT|RFC4217}
    - ✓Restriction: does not affect TLSPORT support (Can be disabled by coding TLSPORT 0)

The solution is to implement RFC 4217 in z/OS in such a way as to support either RFC compliant sessions, or Internet Draft level sessions from earlier releases of FTP for CS for z/OS.   FTP will support both levels, and provide a configuration option to select which level of TLS support you want.

The FTP.DATA statement TLSRFCLEVEL allows you to select which level of Securing FTP with TLS to use.   This statement is supported for both the client and the server.   TLSRFCLEVEL DRAFT is the default value, and is the value you would select if you wanted the same TLS support that z/OS FTP has offered since CSV1R2.

Even though RFC 4217 has dropped the secure port requirement, configuring TLSRFCLEVEL RFC4217 does not affect the z/OS FTP TLSPORT support. Connections to the port specified by the TLSPORT FTP.DATA statement (port 990 by default) are still implicitly secured with TLS. As in prior releases, you can disable implicit TLS security by coding TLSPORT 0 (zero) in FTP.DATA.

Since RFC 4217 allows you to reset a TLS session with a subsequent AUTH command to the server, the FTP client offers a subcommand to allow you to drive the AUTH command to the server.   This subcommand is valid only when the FTP client has configured TLSRFCLEVEL RFC4217, and when the connection has not been implicitly secured with a connection to the TLSPORT. You can use the auth subcommand to reinstate TLS security after using the ccc subcommand.   You could use the auth subcommand to request TLS security after logging into the server without security. In practice, however, it would be difficult to set up your security server to allow this. The FTP login userid would need access to the private cryptographic keys, an undesirable configuration (because it exposes the private keys).

The ccc and cprotect clear subcommands are part of base FTP support. These two subcommands are equivalent – they do the same thing. They each send a CCC command to the server.   On an RFC 4217 compliant server, the CCC command clears security on the control connection; the data connection is left in its current state. When TLSRFCLEVEL RFC4217 is configured, the FTP client will allow you to execute these subcommands while logged in on a TLS secured session. This is because RFC 4217 allows the CCC command to flow during the FTP session. When TLSRFCLEVEL DRAFT is configured, you get the current behavior which is that these subcommands are not allowed during a TLS session.

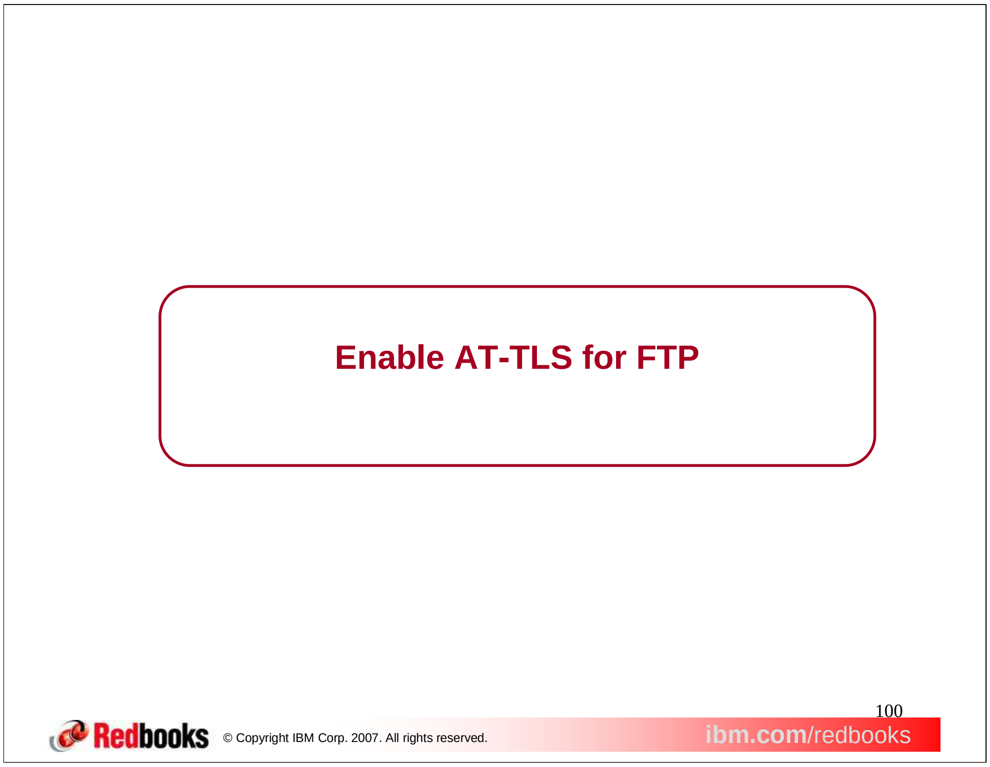
The locsite subcommand is enhanced to let you change the client’s tlsrfclevel.   You have the same options as for the FTP.DATA statement. Use the locstat subcommand to display the client’s current TLSRFCLEVEL value. The locstat subcommand now supports the tlsrfclevel parameter to display only the TLSRFCLEVEL setting.

The server XSTA command has been enhanced to support the TLSRFCLEVEL parameter.   The z/OS FTP client status subcommand with the tlsrfclevel parameter results in an XSTA command being sent to the server. From an OEM client, you could enter QUOTE XSTA (TLSRFCLEVEL to do the same thing.

The comprehensive information returned in the server STAT command reply now includes the TLSRFCLEVEL setting.

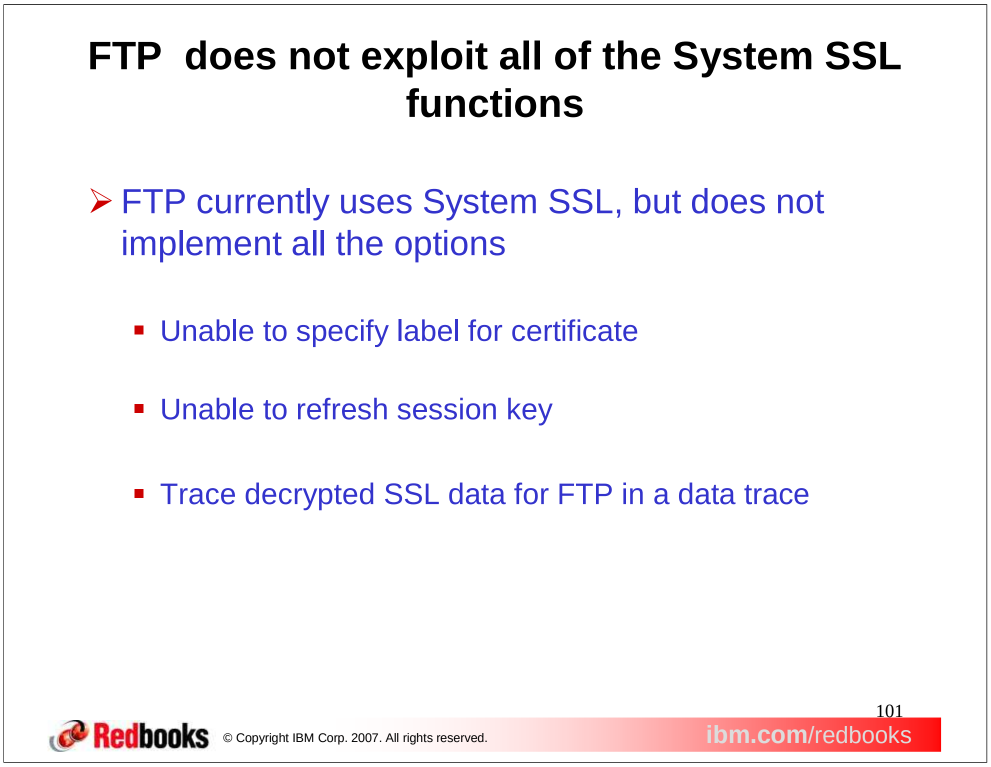
When you configure TLSRFCLEVEL RFC4217, the server will now accept the AUTH and CCC commands during a TLS session. If you set TLSRFCLEVEL to draft, you get the earlier behavior which is to reject these commands during a TLS session. The AUTH command is used to request security for the current session. If your session is already secured, an AUTH command will reset security on the session. The CCC command is used to reset the control connection only.   The data connection security is left in its current state, whatever that may be.   You cannot use server commands to reset the data connection once you have used the CCC command. However, you can reinstate security by using the AUTH command.

RFC 4217 explicitly states that when the session is re-initialized with a REIN command, the control connection is cleared and the data connection reverts to unprotected. Therefore, the server’s action changes from prior release behavior when TLSRFCLEVEL is set to RFC4217. The z/OS FTP client does not send the REIN command to the server unless you use the quote subcommand to send a REIN command to the server (the quote subcommand simply sends its argument to the server and waits for a server reply). Use of QUOTE REIN is not recommended from any FTP implementation regardless of whether you are using TLS security or not because it causes the client and server to lose synchronization.



Enable AT-TLS for FTP

This section discusses the enablement of FTP to use AT-TLS.



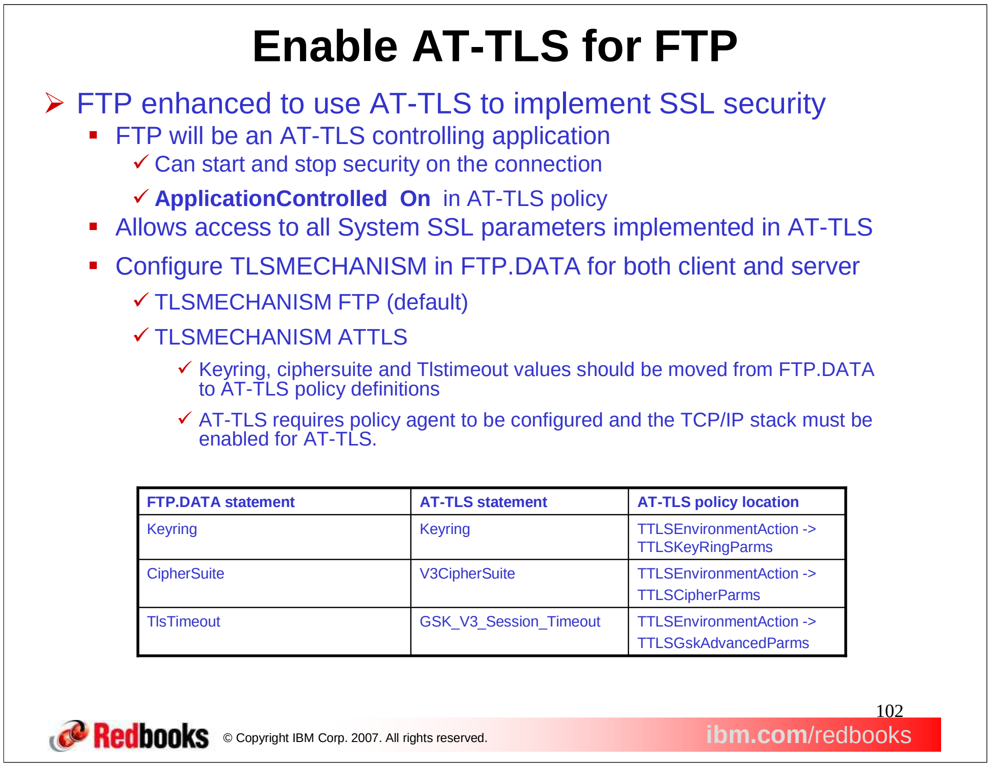
FTP does not exploit all of the System SSL functions

- FTP currently uses System SSL, but does not implement all the options
  - Unable to specify label for certificate
  - Unable to refresh session key
  - Trace decrypted SSL data for FTP in a data trace

101

ibm.com/redbooks

When FTP implemented System SSL in z/OS 1.2, all the functions of System SSL were not exploited. System SSL supports specifying a certificate label to allow certificates other than the default certificate to be used. System SSL also allows session keys to be refreshed during the lifetime of a session. Finally, the TCPIP data trace can be used to trace the decrypted data read by FTP.



Enable AT-TLS for FTP

- ➢ FTP enhanced to use AT-TLS to implement SSL security
  - FTP will be an AT-TLS controlling application
    - ✓ Can start and stop security on the connection
    - ✓ **ApplicationControlled On** in AT-TLS policy
  - Allows access to all System SSL parameters implemented in AT-TLS
  - Configure TLSMECHANISM in FTP.DATA for both client and server
    - ✓ TLSMECHANISM FTP (default)
    - ✓ TLSMECHANISM ATTLS
      - ✓ Keyring, ciphersuite and Tlstimeout values should be moved from FTP.DATA to AT-TLS policy definitions
      - ✓ AT-TLS requires policy agent to be configured and the TCP/IP stack must be enabled for AT-TLS.

FTP.DATA statement	AT-TLS statement	AT-TLS policy location
Keyring	Keyring	TTLSEnvironmentAction -> TTLSKeyRingParms
CipherSuite	V3CipherSuite	TTLSEnvironmentAction -> TTLSCipherParms
TlsTimeout	GSK_V3_Session_Timeout	TTLSEnvironmentAction -> TTLSGskAdvancedParms

The FTP client and server can now be configured to use AT-TLS to support SSL/TLS connections.

There are 3 types of AT-TLS applications. Those that are completely unaware they are using AT-TLS (they use AT-TLS with no code changes at all), those that have AT-TLS awareness but do not control AT-TLS (they can query the stack but not affect the choices it makes), and those that are AT-TLS controlling, meaning the application starts and stops security on the connection. FTP is a controlling AT-TLS application which requires the ApplicationControlled On statement in the AT-TLS policy.

A TTLSEnvironmentAdvancedParms statement is required for FTP to use AT-TLS. The ApplicationControlled statement must be set to On. This allows FTP to start and stop security on a connection. The SecondaryMap statement should also be set to On so that data connections will have the same policy as the control connection. SecondaryMap eliminates the need to code additional TTLSRule statements for the data connection. The FTP protocol will negotiate the security level to be used on the data connection.

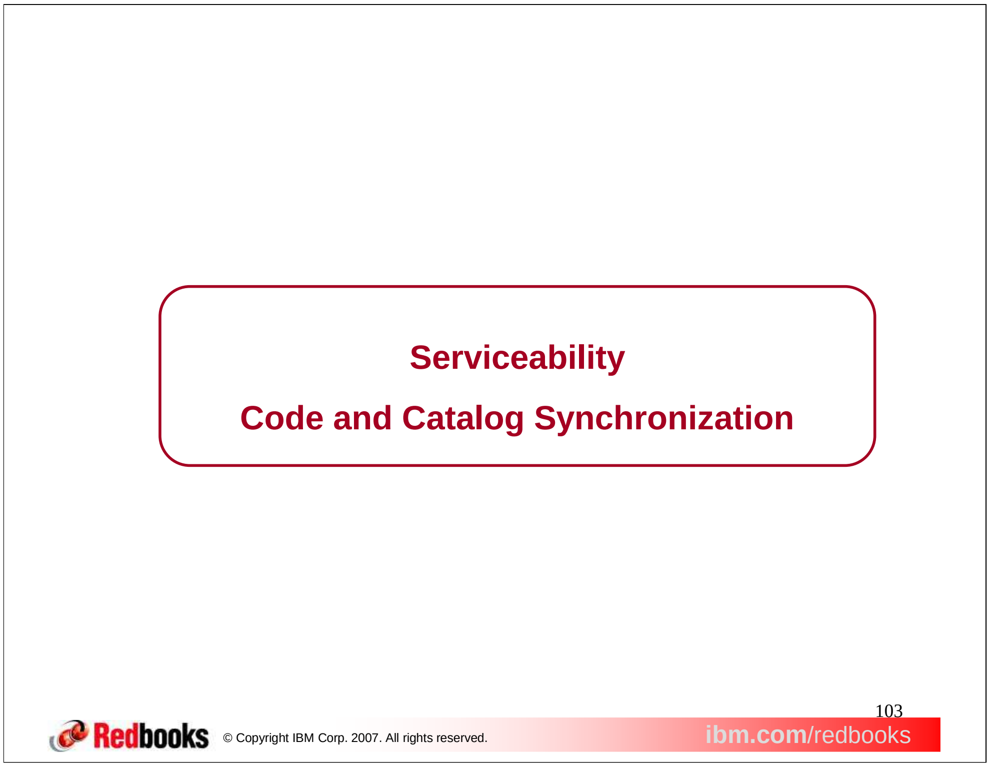
Using AT-TLS allows all of the System SSL parameters supported by AT-TLS to be configured for FTP. For example, a certificate label can be configured instead of the default certificate. AT-TLS can also be configured to refresh the session key on a connection.

A new configuration statement for the FTP client and server, TLSMECHANISM has been created. The default value is FTP, which causes FTP to use the existing System SSL support.

Configuring ATTLS causes FTP to use AT-TLS. This requires AT-TLS policy updates to support AT-TLS on FTP connections. The existing FTP.DATA parameters Keyring, TlsTimeout and Ciphersuite will need to be moved to the AT-TLS policy. The TCP/IP stack must be enabled for AT-TLS and the policy agent must be configured to support AT-TLS.

The TTLSEnvironmentAction statement contains the equivalent AT-TLS parameters. The FTP.DATA Keyring statement will need to be moved to the AT-TLS policy Keyring statement, which is in the TTLSKeyRingParms statement. The CipherSuite statements should be moved to a V3CipherSuite statement in the AT-TLS policy TTLSCipherParms statement. The FTP.DATA TlsTimeout statement should be moved to the GSK_V3_Session_Timeout statement in the AT-TLS policy TTLSGskAdvancedParms statement.

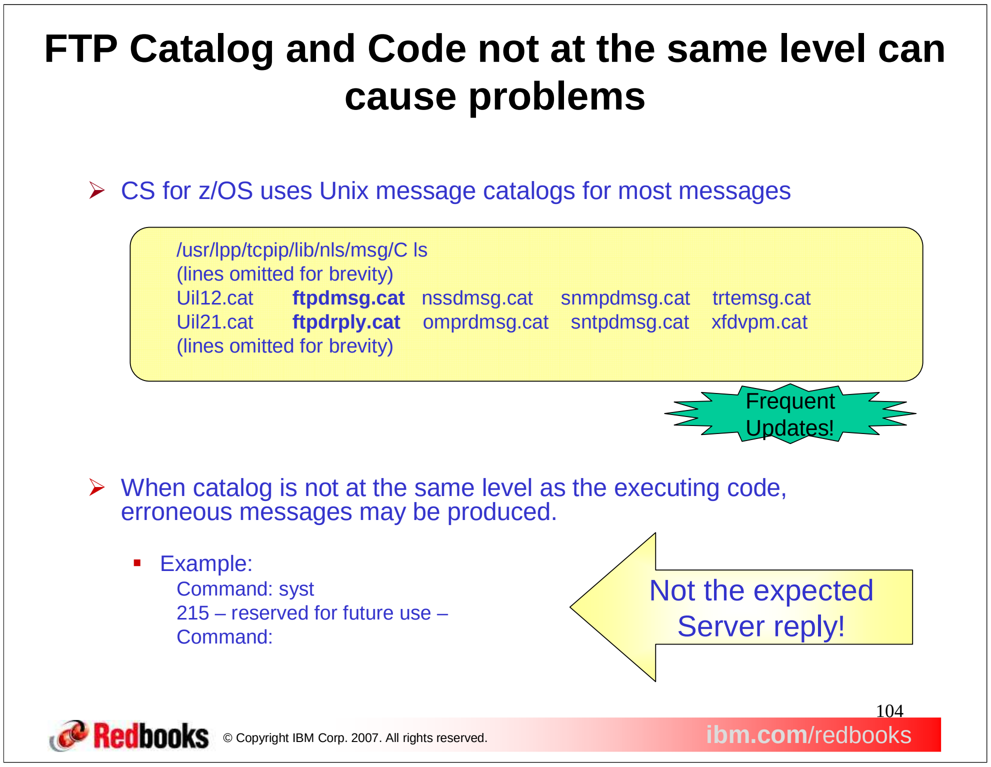
FTP does not support SSLv2 when configured with TLSMECHANISM FTP. AT-TLS does allow SSLv2 connections to be configured, but by default SSLv2 is disabled. It is not recommended that SSLv2 be enabled in the AT-TLS policy for FTP.



Serviceability

Code and Catalog Synchronization

This section describes a serviceability enhancement to FTP to verify the FTP code and FTP catalogs are synchronized.



FTP Catalog and Code not at the same level can cause problems

- CS for z/OS uses Unix message catalogs for most messages

```
/usr/lpp/tcpip/lib/nls/msg/C ls
(lines omitted for brevity)
Uil12.cat      ftpdmsg.cat   nssdmsg.cat    snmpdmsg.cat   trtemsg.cat
Uil21.cat      ftpdrply.cat  omprdmsg.cat   sntpdmsg.cat   xfdvpm.cat
(lines omitted for brevity)
```

Frequent Updates!

- When catalog is not at the same level as the executing code, erroneous messages may be produced.
  - Example:

```
Command: syst
215 – reserved for future use –
Command:
```

The directory shown above is where CS for z/OS provides the Unix message catalogues it uses, along with a partial listing of the Unix message catalogues that CS for z/OS uses.   Files with extension .cat are formatted message files, the executable form of the message catalogue.   For this line item, we are interested in the message catalogues used by FTP, highlighted in **bold** font.

The .cat files are not message source you can translate, but we know that some users are reverse engineering our catalogue source from the .cat files.   Although the messages in a message catalogue can be changed without affecting the source code, you cannot change the order or number of messages in the catalogue. The order and number of messages is strictly bound to the executable code.   Your notes on the next page list restrictions on what you can change in a message catalogue; for this line item we can focus on the problem of adding messages.

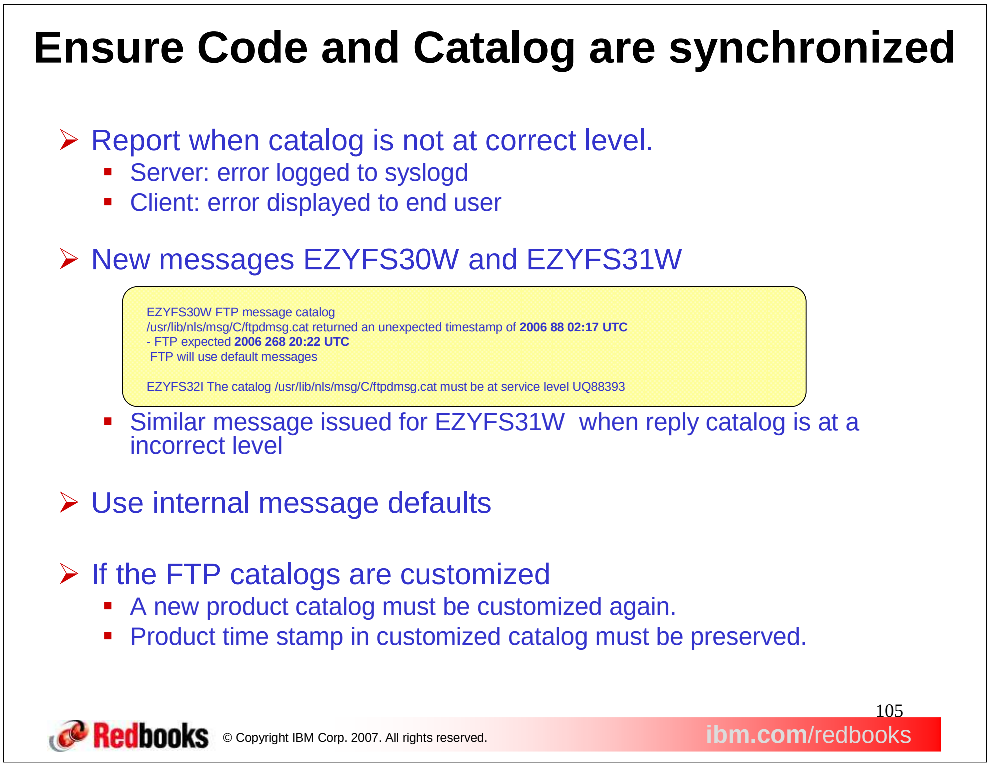
FTP is constantly adding messages to its catalogues. For every release, and often for maintenance fixes as well, CS for z/OS ships FTP catalogue updates as well as executable code updates.

FTP uses the Unix message catalogs to supply the text for FTP end user messages and replies, and operator messages. If these catalogs are not at the same level as the executing code, the wrong text could be fetched from the catalog. For example, if the catalog is down level and FTP is retrieving a new message, the new message may not be defined in the catalog. A message such as '-- reserved for future use –' might be displayed, as in this example. More seriously, had the reply code been 215- instead of 215 – (a space separates 5 and -), the client would, as specified by RFC 959, wait for another 215 reply. Thus, the client would hang because the server won't be sending another reply 215! Finally, the reply code could be completely different.

A catalog synchronization error can arise from several sources:

- The Unix File System containing the catalogs may be incorrectly mounted
- The user has applied code updates but omitted to apply a catalog update
- The user has changed the catalog and it is not kept current when new maintenance is applied.

This support is intended to notify the user when either of the FTP catalogs are not synchronized with the executing code.



Ensure Code and Catalog are synchronized

- Report when catalog is not at correct level.
  - Server: error logged to syslogd
  - Client: error displayed to end user
- New messages EZYFS30W and EZYFS31W

```
EZYFS30W FTP message catalog
/usr/lib/nls/msg/C/ftpdmsg.cat returned an unexpected timestamp of 2006 88 02:17 UTC
- FTP expected 2006 268 20:22 UTC
 FTP will use default messages

EZYFS32I The catalog /usr/lib/nls/msg/C/ftpdmsg.cat must be at service level UQ88393
```

  - Similar message issued for EZYFS31W when reply catalog is at a incorrect level
- Use internal message defaults
- If the FTP catalogs are customized
  - A new product catalog must be customized again.
  - Product time stamp in customized catalog must be preserved.

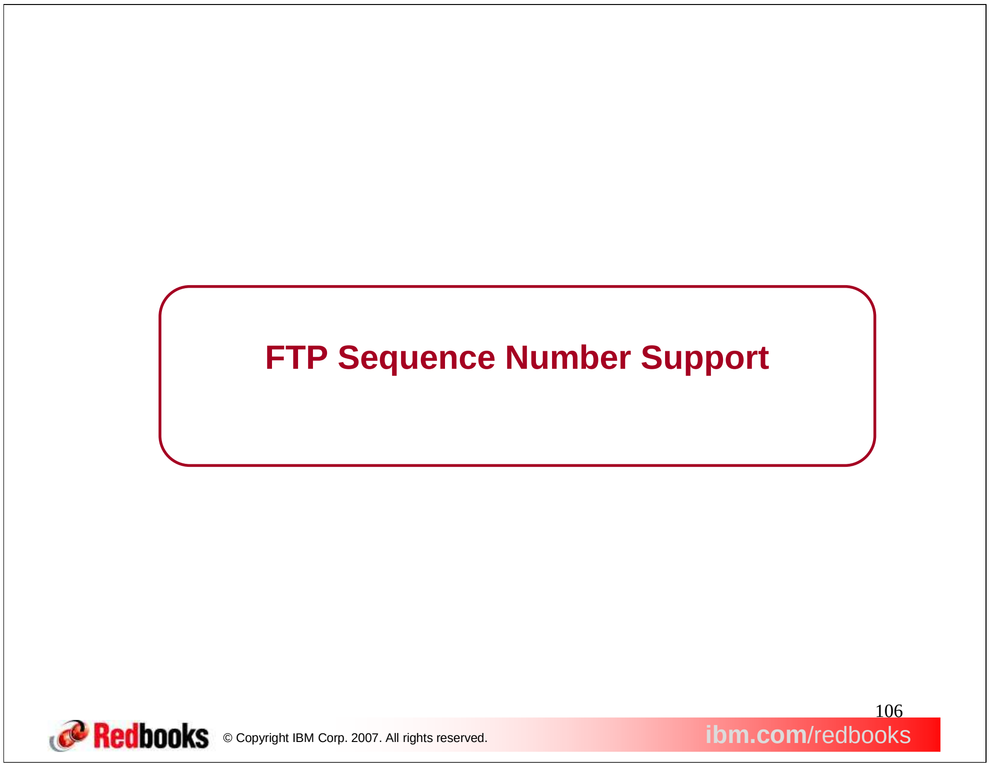
To solve this, FTP takes two actions when the catalog is detected at an incorrect level. A message describing the code and catalog mismatch is generated with information enabling the correct identification of the required message catalog level.

The catalog is closed and default messages are used.

The message (EZYFS30W for the message catalog and EZYFS31W for the reply catalog) identifies the name of the catalog that FTP is connected to and the unexpected time stamp extracted from the message catalog. In this example the time stamp is ‘**2006 88 02:17**’ (i.e. March 29, 2006 at 2:17 am) and the code was expecting **2006 268 20:22** (i.e. Sept 25, 2006 at 8:22 pm). A date and time stamp are contained within the catalog and indicates when the catalog was built for distribution. The expected time stamp is built into the executing code when it is packaged. For a catalog to be considered synchronized, these time stamps must be identical.

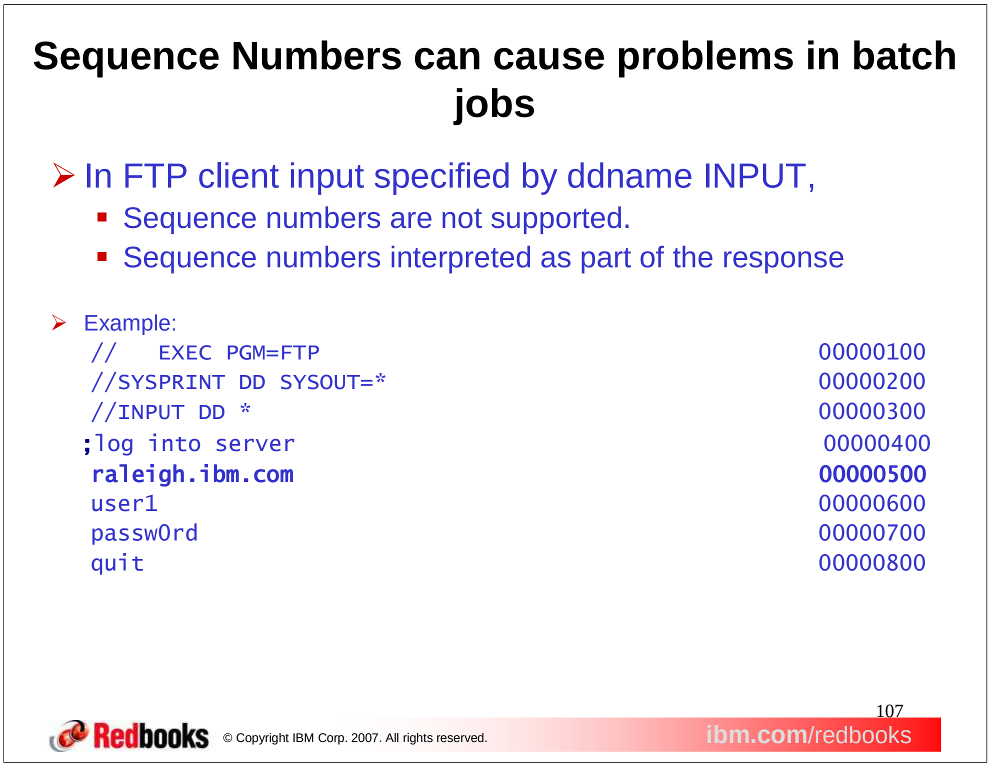
Also, a reminder is issued that default messages will be used. Message EZYFS32I identifies the dataset name of the catalog, and the expected service level. In this case, UQ88393 is a PTF that needs to be installed to bring the catalog to the same level that the code expects. This service level is built into the FTP code when it ships.

When IBM updates an FTP catalog that you have customized, the new product catalog will need to be updated with your local customizations. Along with these updates, the time stamp in the product catalog must be preserved. The Notes in this pitch demonstrate how to do this. Catalogs and code can lose synchronization when migrating to a new release or when new product updates are applied. This is typically caused by mounting an incorrect Unix File System or not updating the Unix File System to contain the correct level of the message catalog. If a customized catalog is being used, it must be kept current with any maintenance that is applied. If the catalog is not kept current with the proper time stamp, the time stamp in the local catalog will not match the time stamp in the product catalog. FTP will detect this and use internal message defaults and any customized messages will not be used.



FTP Sequence Number Support

This section describes the new support to detect and optionally remove sequence numbers in a batch job.



Sequence Numbers can cause problems in batch jobs

- In FTP client input specified by ddname INPUT,
  - Sequence numbers are not supported.
  - Sequence numbers interpreted as part of the response

- Example:

```
 //   EXEC PGM=FTP                                            00000100
 //SYSPRINT DD SYSOUT=*                                      00000200
 //INPUT DD *                                                00000300
;log into server                                              00000400
 raleigh.ibm.com                                             00000500
 user1                                                       00000600
 passw0rd                                                    00000700
 quit                                                        00000800
```

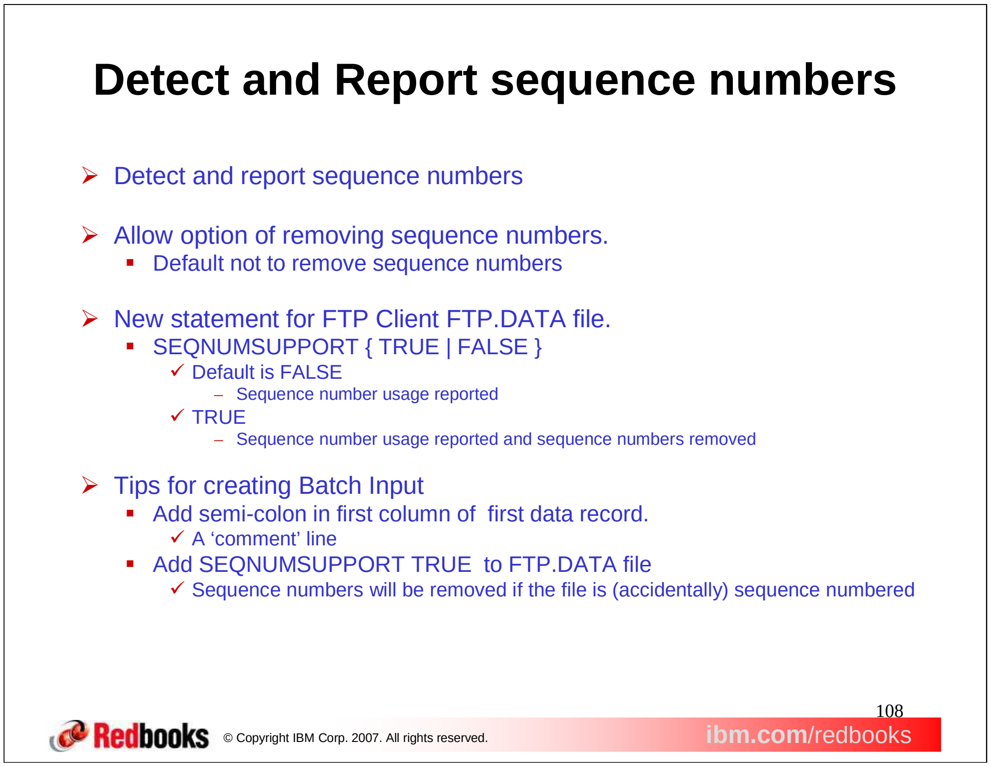
107

Redbooks ibm.com/redbooks

When the FTP client is invoked from JCL, the ddname INPUT describes where the responses to the FTP client's prompts are. The ddname INPUT could point to a data set, a file, or a set of concatenated files. The contents could be defined in the JCL itself, as in this example.

The FTP client supports the use of comments in the input stream when the client is invoked this way. In this example, the line "; log into server" is a comment, because the first column is a semi-colon ';'. The FTP client detects the semi-colon and discards the rest of the line.

In batch, the FTP client reads input via the //INPUT DD statement which contains a list of the responses to each prompt by the FTP client. When updating this file, the user may accidentally activate sequence number support and each line of the input file becomes sequence numbered. As most editors display line numbers, it is not obvious that the file actually may contain sequence numbers. As FTP does not support sequence numbers, when the command input is read, FTP will interpret the sequence numbers as part of the FTP command. This usually results in a command failure. As most of these errors occur in batch jobs, the problem may not immediately be noticed, causing users delay in accomplishing their task. At times, this has even resulted in calls to IBM because FTP is attempting to connect to an incorrect port. This is usually caused when the remote host name is part of the command input and the sequence number becomes interpreted as the remote port to which FTP is connecting. In the example, when FTP is executed it will attempt to connect to PORT 500 because of the sequence number in the input stream.



Detect and Report sequence numbers

- Detect and report sequence numbers
- Allow option of removing sequence numbers.
  - Default not to remove sequence numbers
- New statement for FTP Client FTP.DATA file.
  - SEQNUMSUPPORT { TRUE | FALSE }
    - ✓ Default is FALSE
      - Sequence number usage reported
    - ✓ TRUE
      - Sequence number usage reported and sequence numbers removed
- Tips for creating Batch Input
  - Add semi-colon in first column of first data record.
    - ✓ A ‘comment’ line
  - Add SEQNUMSUPPORT TRUE to FTP.DATA file
    - ✓ Sequence numbers will be removed if the file is (accidentally) sequence numbered

The solution is to report when sequence numbers are detected in the input stream. This may still result in FTP subcommands failing, but messages will inform the user as to the probable cause. This enables FTP to function as it does with prior releases.

As an option, allow the user to remove sequence numbers when detected. This will allow a job that would otherwise fail, to successfully complete.

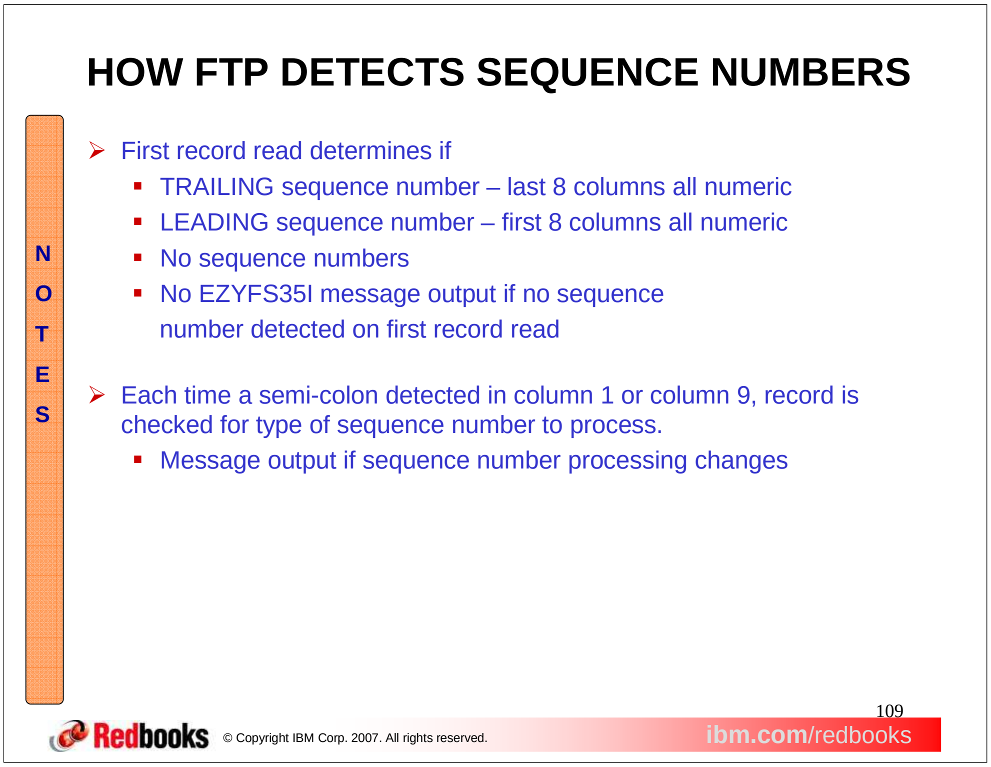
A new statement is added to the FTP Client’s FTP.DATA file. The keyword is SEQNUMSUPPORT and when coded as FALSE, sequence number usage will be reported but not removed. When SEQNUMSUPPORT TRUE is coded, sequence numbers will not only be reported, they will be removed.

With SEQNUMSUPPORT FALSE coded in the FTP.DATA file, the FTP client checks if sequence numbers are present. The type of sequence number is determined by the first record read or whenever a semi-colon is detected in the first data column. If FTP sequence numbers are present, an EZYFS34W message is issued to inform the user that sequence numbers are present and the type of sequence number detected. No sequence numbers will be removed. LEADING sequence numbers usually occur when input is read from a variable length file. TRAILING sequence numbers occur when input is read from fixed length file. Message EZYFS35I is issued when FTP detects a transition from processing LEADING or TRAILING sequence numbers to processing no sequence numbers.

With SEQNUMSUPPORT TRUE coded in the FTP.DATA, FTP will detect and remove sequence numbers. When FTP transitions from one type of sequence number processing to another, it will output EZYFS33I or EZYFS35I. EZYFS33I is issued when FTP will be removing LEADING or TRAILING sequence numbers. When EZYFS35I is output, it means that a previous EZYFS33I message was output indicating the type of sequence numbers FTP was removing. However, the EZYFS35I message indicates that FTP will no longer remove sequence numbers from the input.

When the FTP client reads its subcommands from the INPUT DD statement, it is not aware the subcommands are being input from more than one file. To insure that FTP uses the proper sequence numbering scheme, add a semi-colon as the first data column of the first record of each file. This semi-colon protects the file from any accidental sequence numbering if SEQNUMSUPPORT TRUE is coded in the FTP.DATA file. If a conflicting sequence number is detected, EZYFS34W will be issued the first time a mismatched sequence number is detected.

For example, suppose that the original file starts off with no sequence numbers and then a concatenated file is read in that has TRAILING sequence numbers. If this second file has the semi-colon coded in the first record, FTP would issue an “EZYFS33I FTP will remove TRAILING sequence numbers from input commands”



HOW FTP DETECTS SEQUENCE NUMBERS

N
O
T
E
S

- First record read determines if
  - TRAILING sequence number – last 8 columns all numeric
  - LEADING sequence number – first 8 columns all numeric
  - No sequence numbers
  - No EZYFS35I message output if no sequence number detected on first record read
- Each time a semi-colon detected in column 1 or column 9, record is checked for type of sequence number to process.
  - Message output if sequence number processing changes

Redbooks ibm.com/redbooks

When FTP reads the first record, it determines the type of sequence number by examining the last 8 columns and then the first eight columns of data to determine if it is numeric.

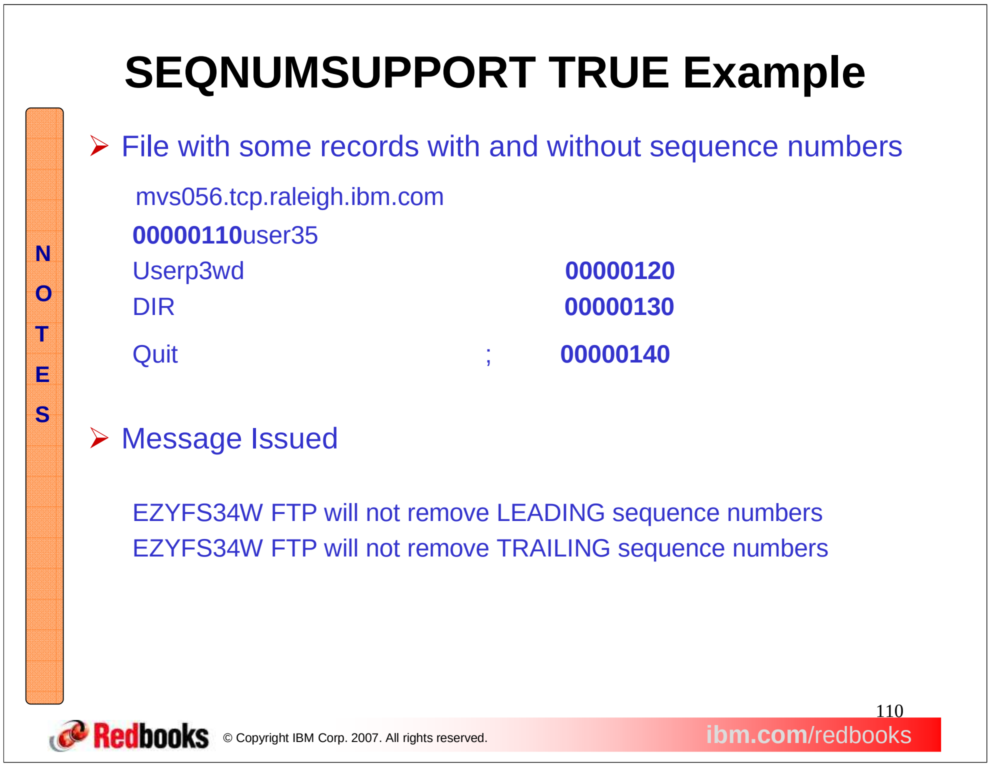
TRAILING sequence numbers have numerals in the last 8 columns which will be replaced with blanks

LEADING sequence numbers have numerals in the first 8 columns and will have the data shifted left 8 columns, with the last 8 columns replaced with blanks.

With fixed length records the first data column is 1. With variable length records, the first data column is 9, because the first 8 columns are occupied by the sequence number.

If a semi-colon is detected in these columns, FTP re-evaluates the type of sequence numbers it expects to process.

When processing TRAILING or LEADING sequence numbers and the expected sequence number does not appear in the columns expected, data will not be removed and command will be processed as entered.



SEQNUMSUPPORT TRUE Example

NOTES

- File with some records with and without sequence numbers

```
mvs056.tcp.raleigh.ibm.com
00000110user35
Userp3wd                                    00000120
DIR                                         00000130
Quit                                ;       00000140
```

- Message Issued

```
EZYFS34W FTP will not remove LEADING sequence numbers
EZYFS34W FTP will not remove TRAILING sequence numbers
```

110

Redbooks ibm.com/redbooks

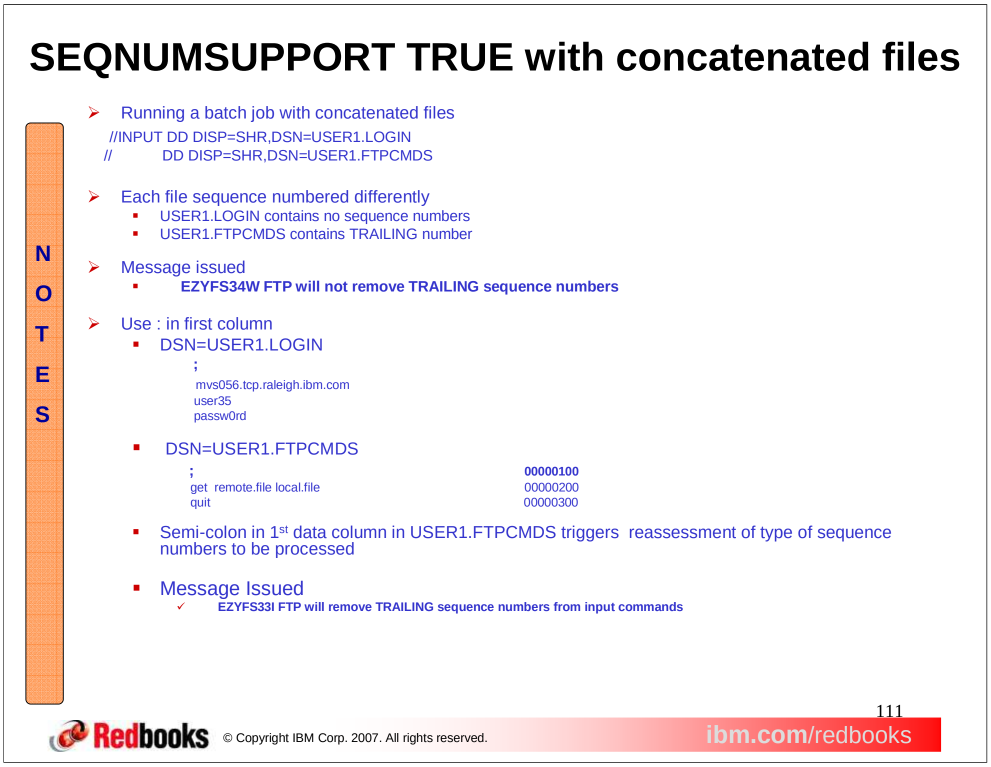
This is an example of a file which contains some records with and without sequence numbers. This is likely to happen when the initial file is created without sequence numbers and then updated with an editor which adds sequence numbers.

The first record of the file contains no sequence numbers. Any sequence numbers in the file will not be stripped off. FTP does not issue any message in this case because there is no action that FTP will be taking. This makes this support transparent with previous releases.

When the second line is read, FTP will detect the sequence number because columns 1 thru 8 are numeric. This is in conflict with the original sequence number detected on the first record which indicated the file contains no sequence numbers. Message ‘EZAFS34W message is issued to indicate LEADING sequence numbers will not be removed.

When the third line is read, the last 8 columns contain numerals. Message EZAFS34W is issued again indicating that TRAILING sequence numbers will not be removed.

Each of these messages will only be issued one time to give an indication of why the subcommand may fail. Thus, no additional message will be issued when the fourth line is read.



SEQNUMSUPPORT TRUE with concatenated files

NOTES

- Running a batch job with concatenated files

```
  //INPUT DD DISP=SHR,DSN=USER1.LOGIN
 //          DD DISP=SHR,DSN=USER1.FTPCMDS
```

- Each file sequence numbered differently
  - USER1.LOGIN contains no sequence numbers
  - USER1.FTPCMDS contains TRAILING number
- Message issued
  - **EZYFS34W FTP will not remove TRAILING sequence numbers**
- Use : in first column
  - DSN=USER1.LOGIN

```
;
 mvs056.tcp.raleigh.ibm.com
user35
passw0rd
```

  - DSN=USER1.FTPCMDS

```
;                                        00000100
get  remote.file local.file              00000200
quit                                     00000300
```

  - Semi-colon in 1st data column in USER1.FTPCMDS triggers reassessment of type of sequence numbers to be processed
  - Message Issued
    - ✓ **EZYFS33I FTP will remove TRAILING sequence numbers from input commands**

111

ibm.com/redbooks

Sequence number support becomes more complicated when multiple files or members are concatenated as input. This example depicts two files being used as input to FTP. One file has no sequence numbers and the other file contains trailing sequence numbers.

When the USER1.LOGIN file is read, FTP detects no sequence numbers and will process all remaining commands without removing sequence numbers.

When USER1.FTPCMDS is read, FTP is not aware that data is being read from the second file. FTP reads the input as one continuous file. When the first record of the 2nd file is read, EZYFS34W is issued to indicate at least one record has been detected with an unexpected sequence number. The sequence number is not removed and can result In a command failure.

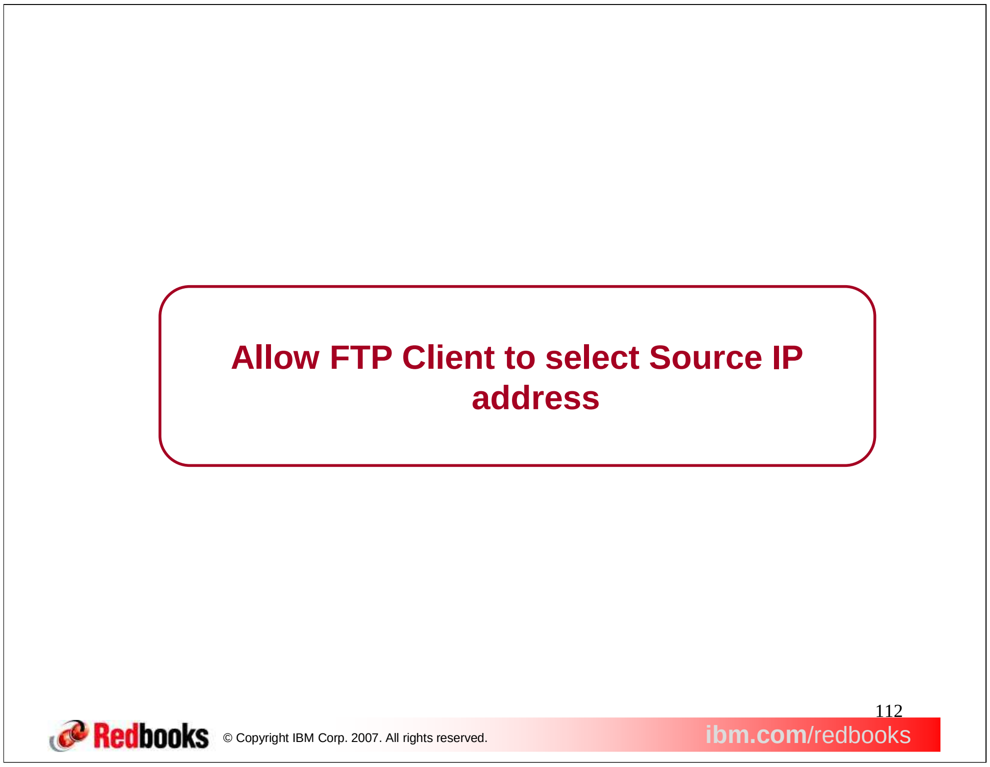
The EZYFS34W message provides a warning as to why the command may have failed.

To enable the full benefit when SEQNUMSUPPORT TRUE is coded in the FTP.DATA file, add a semi-colon as the first record of any file that contains FTP commands. In the above slide, data under **DSN=USER1.LOGIN** and **DSN=USER1.FTPCMDS** show the content of the files. In this sample, the concatenation sequence results in USER1.LOGIN being read first, followed by USER1.FTPCMDS.

When the semi-colon is detected in the first data column from **DSN=USER1.LOGIN**, FTP will interrogate the line to determine the type of sequence number it contains. In this case, no sequence number is detected and it will process the file as containing no sequence numbers.

When the first line of USER1.FTPCMDS is read, it contains a semicolon which causes FTP to reassess the type of sequence number to be processed. In this case a TRAILING sequence number is detected which is different than the NO SEQUENCE numbers that FTP is currently processing. FTP will output message EZYFS33I to indicate that it will begin removing TRAILING sequence numbers.

This allows the commands in the file USER1.FTPCMDS to be processed successfully.



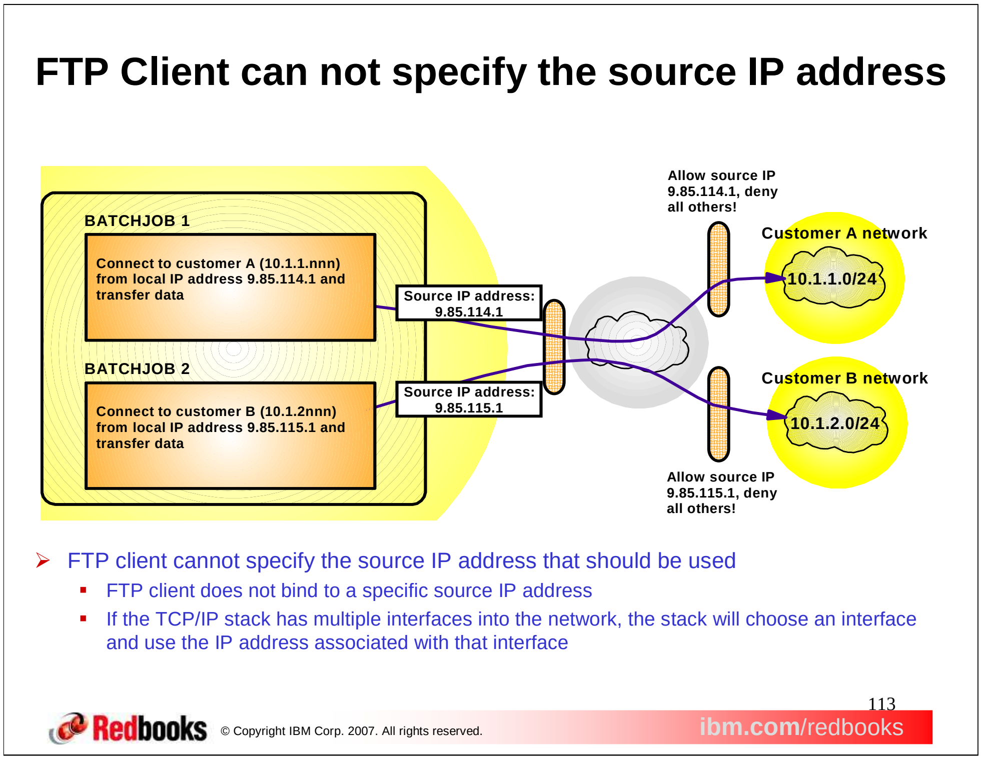
Allow FTP Client to select Source IP address

112

Redbooks

ibm.com/redbooks

This section covers the new function to enables specifying the source IP address that will be used for connections from the FTP client to the FTP server.



FTP Client can not specify the source IP address

BATCHJOB 1
Connect to customer A (10.1.1.nnn) from local IP address 9.85.114.1 and transfer data
BATCHJOB 2
Connect to customer B (10.1.2nnn) from local IP address 9.85.115.1 and transfer data
Source IP address: 9.85.114.1
Source IP address: 9.85.115.1
Allow source IP 9.85.114.1, deny all others!
Customer A network
10.1.1.0/24
Customer B network
10.1.2.0/24
Allow source IP 9.85.115.1, deny all others!

- FTP client cannot specify the source IP address that should be used
  - FTP client does not bind to a specific source IP address
  - If the TCP/IP stack has multiple interfaces into the network, the stack will choose an interface and use the IP address associated with that interface

113

Redbooks ibm.com/redbooks

Currently, there is no way for the FTP client to specify which source IP address should be used when connecting to the FTP server.

The TCP/IP stack determines the source IP address. This can be based on TCP/IP configuration options such as Job-Specific Source IP or it may be determined when the route to the FTP server is found.

In some situations the FTP client may want to use a different source IP address when connecting to different FTP servers. In firewall configurations, it may be necessary to use a specific source IP address for the firewall to allow the connection. But, there is no way for the FTP client, itself, to specify the source IP address that should be used.

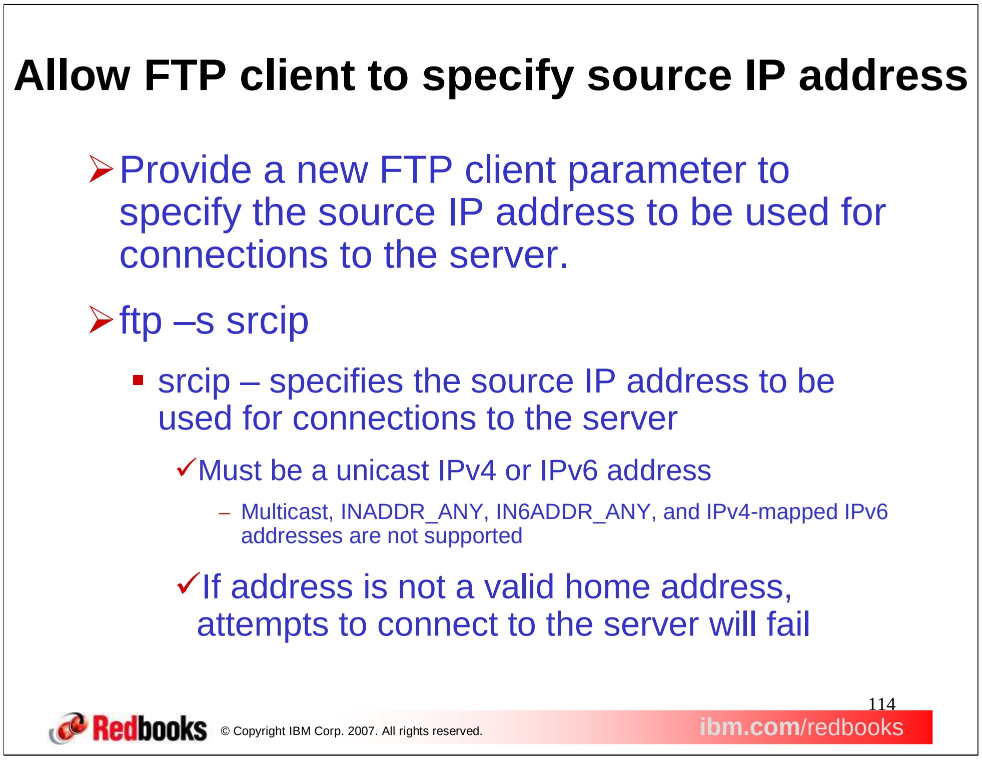
This diagram shows an example of when the FTP client may want to specify the source IP address.

In the diagram, the user has a network setup where the z/OS system running the FTP client has two interfaces into the network.

The user needs to be able to FTP into two other networks which are protected by firewalls. The firewalls are configured to only allow connections from specific IP addresses.

So the only way to successfully FTP into “user A network”, is to use a source IP address of 9.85.114.1

Since there is no way for the FTP client to specify a source IP address, there is no guarantee that the TCP/IP stack would choose the correct interface. Since there are two interfaces into the network the TCP/IP stack may choose either interface.



Allow FTP client to specify source IP address

- Provide a new FTP client parameter to specify the source IP address to be used for connections to the server.
- ftp –s srcip
  - srcip – specifies the source IP address to be used for connections to the server
    - ✓Must be a unicast IPv4 or IPv6 address
      - Multicast, INADDR_ANY, IN6ADDR_ANY, and IPv4-mapped IPv6 addresses are not supported
    - ✓If address is not a valid home address, attempts to connect to the server will fail

Redbooks

ibm.com/redbooks

A new FTP client command line parameter will be added to allow the specification of the source IP address that will be used for connections to the FTP server.

The new command line parameter is:

```
-s srcip
```

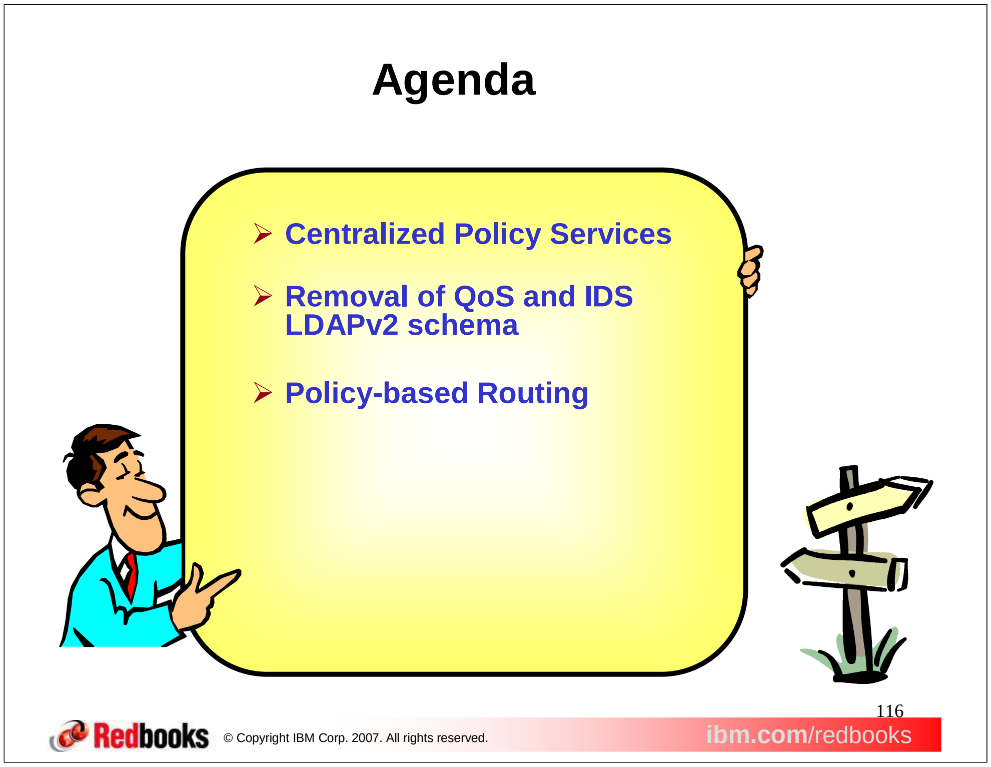
The srcip must be a unicast IPv4 or IPv6 address. Multicast , INADDR_ANY, IN6ADDR_ANY, and IPv4-mapped IPv6 addresses are not supported. If an invalid address is specified then the FTP command will be rejected. If a valid address is specified, but the address is not an active home address on the TCP/IP stack, connections to the server will fail.

ibm.com

Policy Enhancements

International Technical Support Organization

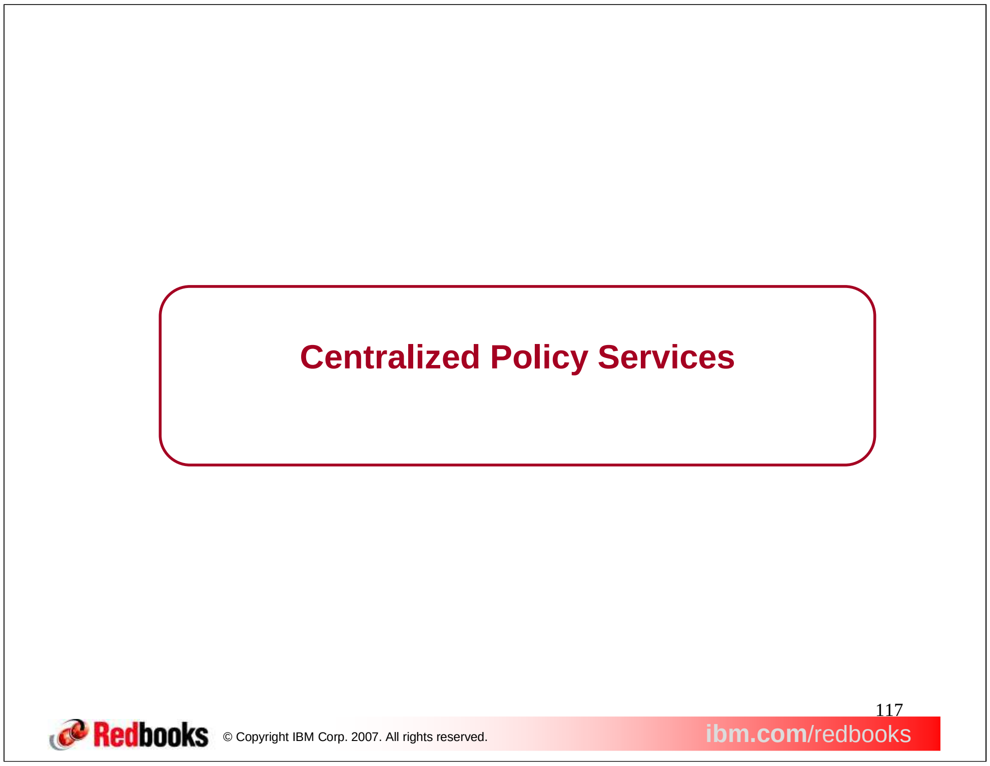
This presentation covers enhancements to the Policy agent and defines a new discipline, routing, used for Policy-based routing.



Agenda

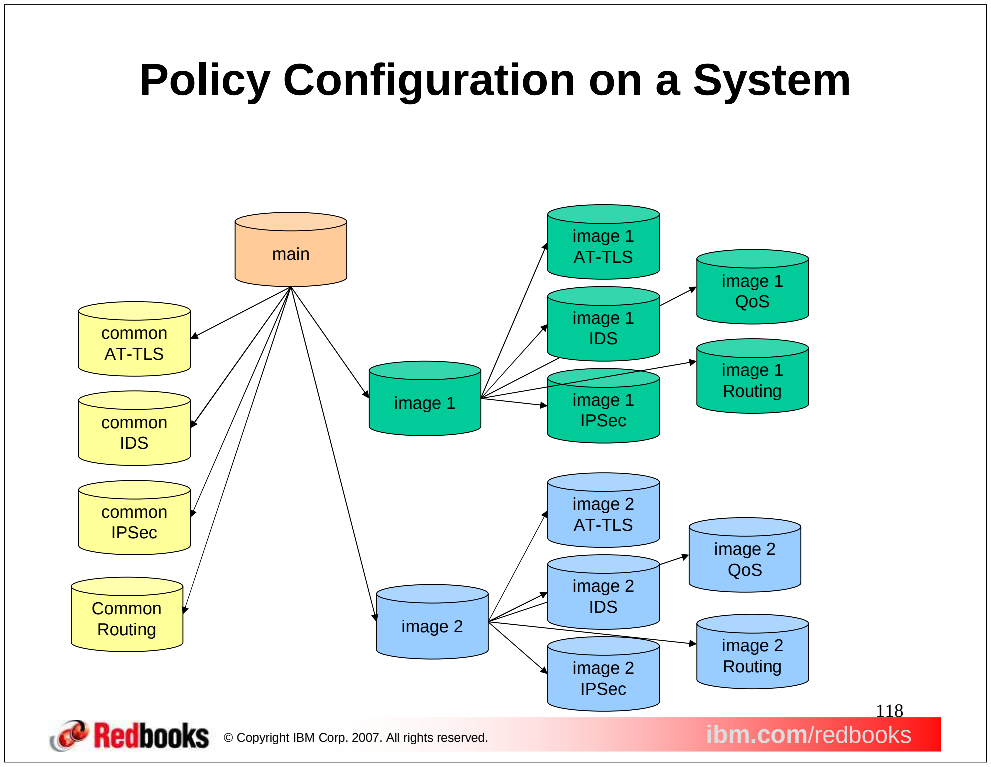
- Centralized Policy Services
- Removal of QoS and IDS LDAPv2 schema
- Policy-based Routing

In this presentation we will discuss managing and administering policies from a central location, the removal of Qos and IDS LDAPv2 schema and a new policy discipline call Policy based routing.



Centralized Policy Services

This section describes the new Centralized Policy Services function.



Policy Configuration on a System

118

Redbooks

ibm.com/redbooks

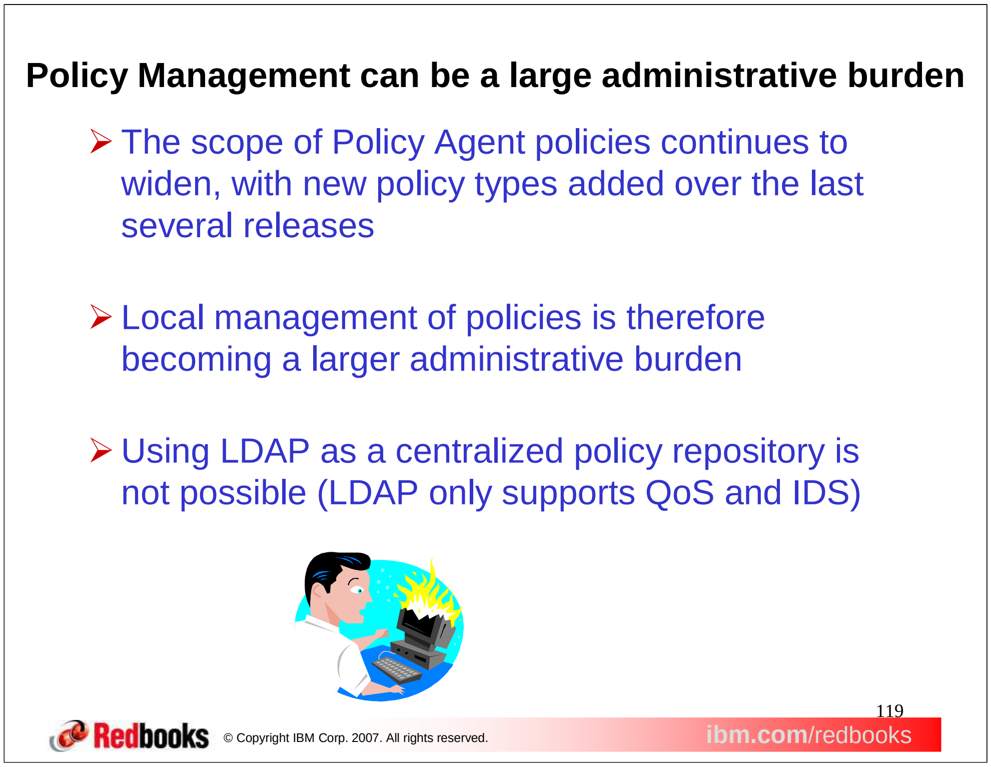
First, let's have a quick refresher on the major attributes of the Policy Agent. Primarily, the Policy Agent is a manager for sets of policy definitions. The policy definitions are categorized into different policy types, as shown on this slide. Each policy type can be used to accomplish various goals.

The Policy Agent is one of several components that provide a more general function known as *policy based networking*. Policy based networking is a way of accomplishing a set of network goals through the use of policy definitions. For example, one network goal may be to provide better quality of service (QoS) for one set of traffic as compared to another set. Policies can be defined to set the IPv4 type of service (TOS) or IPv6 traffic class for the two sets of traffic, to assist in obtaining the required QoS from the network.

This slide graphically depicts the entire set of configuration files that can be used to define the different policy types, as well as general Policy Agent configuration. It's important to understand that various subsets of the configuration files shown might be used, depending on the different policy types in use and the number of TCP/IP stacks supported by an instance of the Policy Agent. Also note that Lightweight Directory Access Protocol (LDAP) configuration is not shown in this diagram.

When the Policy Agent is started the main configuration file is identified using a standard search order. This file in turn can point to one or more image configuration files using the TcpImage statement. Each image configuration file is used to configure policies for one TCP/IP stack. The image files can in turn point to image-specific files for the different policy types. The main configuration file can point to common files for all policy types except QoS. A given common configuration file applies to all TCP/IP stacks. This allows policy definitions that are not unique for each TCP/IP stack to be placed in the common file, and those that are unique to be placed in each image-specific file.

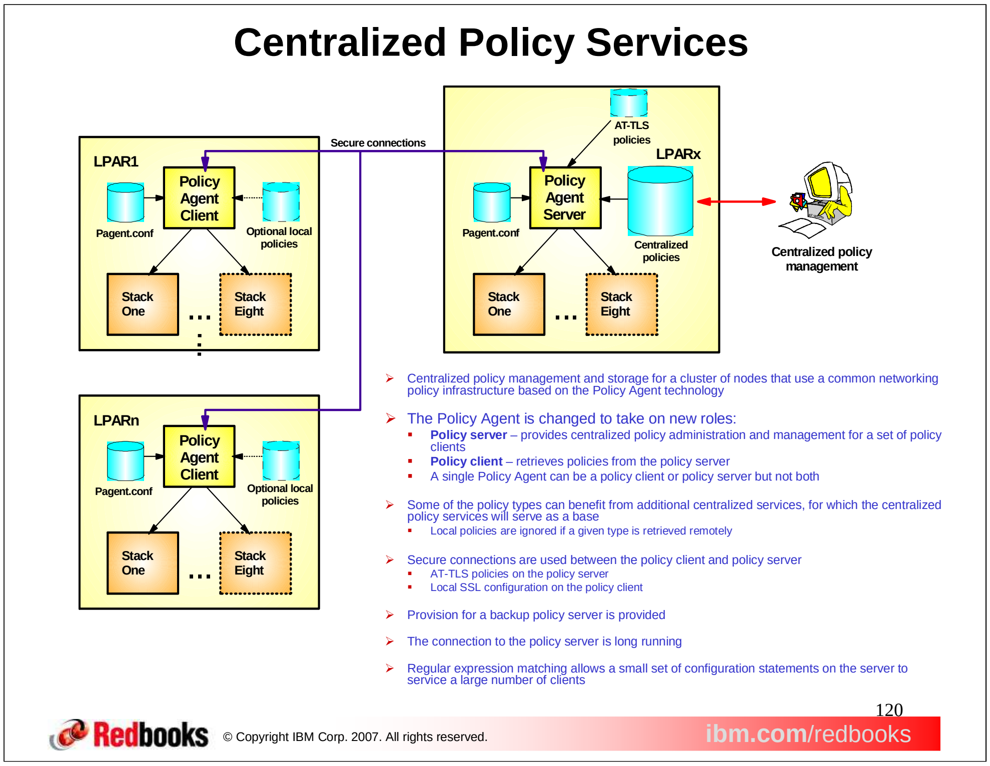
The image QoS file is optional – QoS definitions can be placed directly in the image configuration file instead of a separate file. Also, the statements in the image configuration files can instead be placed directly in the main configuration file, by specifying a TcpImage statement without a separate image file path name. However, such definitions will be shared by all TCP/IP stacks that don't have their own separate image configuration file.



Policy Management can be a large administrative burden

- The scope of Policy Agent policies continues to widen, with new policy types added over the last several releases
- Local management of policies is therefore becoming a larger administrative burden
- Using LDAP as a centralized policy repository is not possible (LDAP only supports QoS and IDS)

The problem being solved by centralized policy services is primarily one of policy management. Each of the last several releases has introduced a new policy type, and the Policy Agent configuration shown on the previous slide needs to be replicated on each system. If the IBM Configuration Assistant for z/OS Communications Server is used to configure policy definitions, it also must be replicated on (or have connectivity to) each system. It isn’t possible to use LDAP as a centralized policy repository, because the LDAP implementation only supports the QoS and IDS policy types.



Centralized Policy Services

- Centralized policy management and storage for a cluster of nodes that use a common networking policy infrastructure based on the Policy Agent technology
- The Policy Agent is changed to take on new roles:
  - **Policy server** – provides centralized policy administration and management for a set of policy clients
  - **Policy client** – retrieves policies from the policy server
  - A single Policy Agent can be a policy client or policy server but not both
- Some of the policy types can benefit from additional centralized services, for which the centralized policy services will serve as a base
  - Local policies are ignored if a given type is retrieved remotely
- Secure connections are used between the policy client and policy server
  - AT-TLS policies on the policy server
  - Local SSL configuration on the policy client
- Provision for a backup policy server is provided
- The connection to the policy server is long running
- Regular expression matching allows a small set of configuration statements on the server to service a large number of clients

This picture shows an overview of the centralized policy services solution. On the left side are a number of policy clients. Each policy client can use local configuration file as usual, if needed. On the right side is the policy server.   Centralized policies are defined, but are not installed in any TCP/IP stacks, on the policy server. These centralized policies are retrieved by the policy clients using the existing Policy Agent API (PAPI).

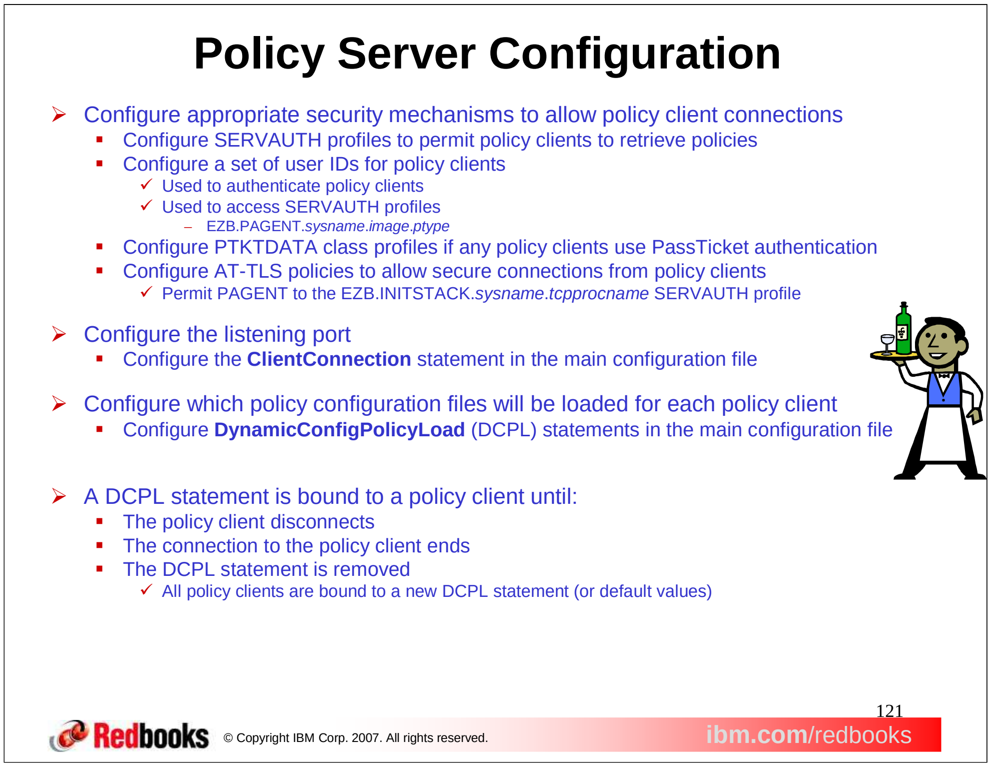
The IBM Configuration Assistant can be used to define the centralized policies, as well as local policies for the policy server and policy client (this is not shown).

To take full advantage of this solution, local policies should not be defined on the policy clients. The policy server is not itself considered a policy client, so local policies on the policy server are normal and expected.

The problem being solved by centralized policy services is primarily one of policy management. Each of the last several releases has introduced a new policy type, and the Policy Agent configuration needs to be replicated on each system. If the IBM Configuration Assistant for z/OS Communications Server is used to configure policy definitions, it also must be replicated on (or have connectivity to) each system. Centralized policy services provides a centralized policy management and storage for a cluster of nodes that use a common networking policy infrastructure based on the Policy Agent technology. Initially a cluster of z/OS nodes is supported. However it can be extended to act as centralized networking policy server for heterogeneous nodes. Centralized management becomes increasingly important as networking policy scope widens (QoS, IDS, IPSec, AT-TLS, PBR, etc.).

The Policy Agent is changed to take on new roles policy server and policy client. The policy server provides centralized policy administration and management for a set of policy clients. The policy client retrieves policies from the policy server. A single Policy Agent can be a policy client or policy server but not both.

Also note that secure long running connections are used between the policy clients and the policy server. The policy server utilizes local AT-TLS policies to accomplish this. But it isn't possible to use AT-TLS policies on the policy client, because of the chicken-and-egg problem: if the AT-TLS policies reside on the policy server, the policy client would need to connect to the policy server in order to obtain the policies that secure that very connection. For this reason, the policy clients are configured as SSL clients, using local definitions in the image configuration files.



Policy Server Configuration

- Configure appropriate security mechanisms to allow policy client connections
  - Configure SERVAUTH profiles to permit policy clients to retrieve policies
  - Configure a set of user IDs for policy clients
    - ✓ Used to authenticate policy clients
    - ✓ Used to access SERVAUTH profiles
      - EZB.PAGENT.*sysname.image.ptype*
  - Configure PTKTDATA class profiles if any policy clients use PassTicket authentication
  - Configure AT-TLS policies to allow secure connections from policy clients
    - ✓ Permit PAGENT to the EZB.INITSTACK.*sysname.tcpprocname* SERVAUTH profile
- Configure the listening port
  - Configure the **ClientConnection** statement in the main configuration file
- Configure which policy configuration files will be loaded for each policy client
  - Configure **DynamicConfigPolicyLoad** (DCPL) statements in the main configuration file
- A DCPL statement is bound to a policy client until:
  - The policy client disconnects
  - The connection to the policy client ends
  - The DCPL statement is removed
    - ✓ All policy clients are bound to a new DCPL statement (or default values)

121

Redbooks

ibm.com/redbooks

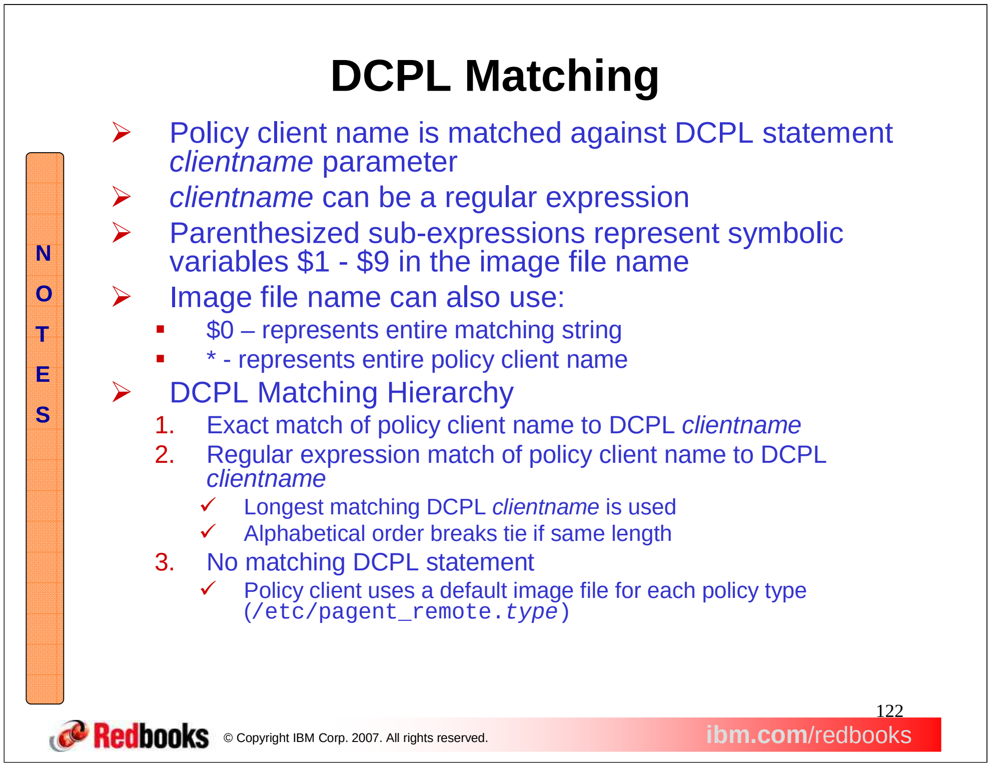
The first step is configuring various security mechanisms to allow policy clients to connect to the policy server. The first item deals with the existing security product EZB.PAGENT.*sysname*.*image*.*ptype* SERVAUTH profile. This profile is currently used to authorize Policy Agent clients (such as the pasearch command and the IKE daemon) to access various policy types for different TCP/IP stacks. For z/OS V1R9, the *image* portion of the profile name is now generic, and can be either a TCP/IP stack or a policy client name. This profile must exist to allow policy clients to retrieve policies. Each policy client presents a user ID when it connects to the Policy Agent. This user ID is used to authenticate with the policy server (using either a password or PassTicket), and to access the EZB.PAGENT.*sysname*.*image*.*ptype* SERVAUTH profiles defined in the previous step. A unique user ID can be created for each policy client, but that is not a requirement. For example, you might decide to use one user ID for a system or other set of policy clients.

You can optionally use program control for the Policy Agent. This provides enhanced control for who is allowed to run the Policy Agent. To do this, permit the Policy Agent user ID to the BPX.DAEMON FACILITY class profile. You may want to use PassTickets instead of passwords to authenticate policy clients with the policy server. This prevents the passwords from being coded in the policy client image configuration files. However, the use of PassTickets requires that you define PTKTDATA class profiles on both the policy client and policy server. These profiles contain a secure key that is used to generate the one-time usage PassTickets. The profile name for these PTKTDATA profiles must be PAGENT on the policy clients, but can be either PAGENT or PAGENT.*userid* on the policy server. The time of day clocks on the policy server and all policy clients using PassTickets must also be reasonably synchronized (within a few minutes). This is because PassTickets are only valid for 10 minutes between generation (on the policy client) and verification (on the policy server). You must configure AT-TLS policies on the policy server to allow secure connections to be established from policy clients. These policies must point to a key ring that contains the appropriate server certificate. Because Policy Agent uses AT-TLS policies, it must be permitted to the EZB.INITSTACK.*sysname*.*tcpprocname* SERVAUTH profile. This allows the Policy Agent to establish sockets prior to the AT-TLS policies being installed.

The next step to configure the policy server is to define a listening socket using the ClientConnection statement. The only parameter on this statement is the port number. The default port number is 16310. The Policy Agent listens for connections using IN6ADDR_ANY and this port number. We recommend that you reserve this port using the PORT statement in the TCP/IP profile.

The last policy server configuration step is to define one or more DynamicConfigPolicyLoad statements. These statements determine what configuration files are used to contain the centralized polices for all policy clients. When a policy client connects, an attempt is made to match the case-sensitive client name to the *clientname* parameter on a DCPL statement. Default values are used if a DCPL statement can’t be matched. The *clientname* parameter can use regular expression characters to match a set of policy clients. Each DCPL statement points to a common configuration file and image-specific configuration files for each policy type. The image-specific configuration file names can contain symbolic replacement or wildcard variables, so that the resulting configuration file is unique for each policy client that matches the DCPL statement. Dynamic update using the -i startup option is not supported for these files.

Once a DCPL statement is matched to a policy client, it is bound to that policy client until one of the events listed on this slide occurs.



DCPL Matching

- Policy client name is matched against DCPL statement *clientname* parameter
- *clientname* can be a regular expression
- Parenthesized sub-expressions represent symbolic variables $1 - $9 in the image file name
- Image file name can also use:
  - $0 – represents entire matching string
  - * - represents entire policy client name
- DCPL Matching Hierarchy
  1. Exact match of policy client name to DCPL *clientname*
  2. Regular expression match of policy client name to DCPL *clientname*
     - ✓ Longest matching DCPL *clientname* is used
     - ✓ Alphabetical order breaks tie if same length
  3. No matching DCPL statement
     - ✓ Policy client uses a default image file for each policy type (`/etc/pagent_remote.`*`type`*)

As noted on the previous slide, policy clients are matched to a DCPL statement using the *clientname* parameter, which can be a regular expression. This regular expression is similar to (but not exactly like) regular expressions used on UNIX commands like grep and awk. Parentheses can be used in the regular expression to create sub-expressions. These sub-expressions can then be represented by symbolic replacement variables in the image configuration file names. Stay tuned for an example to help make this clearer.

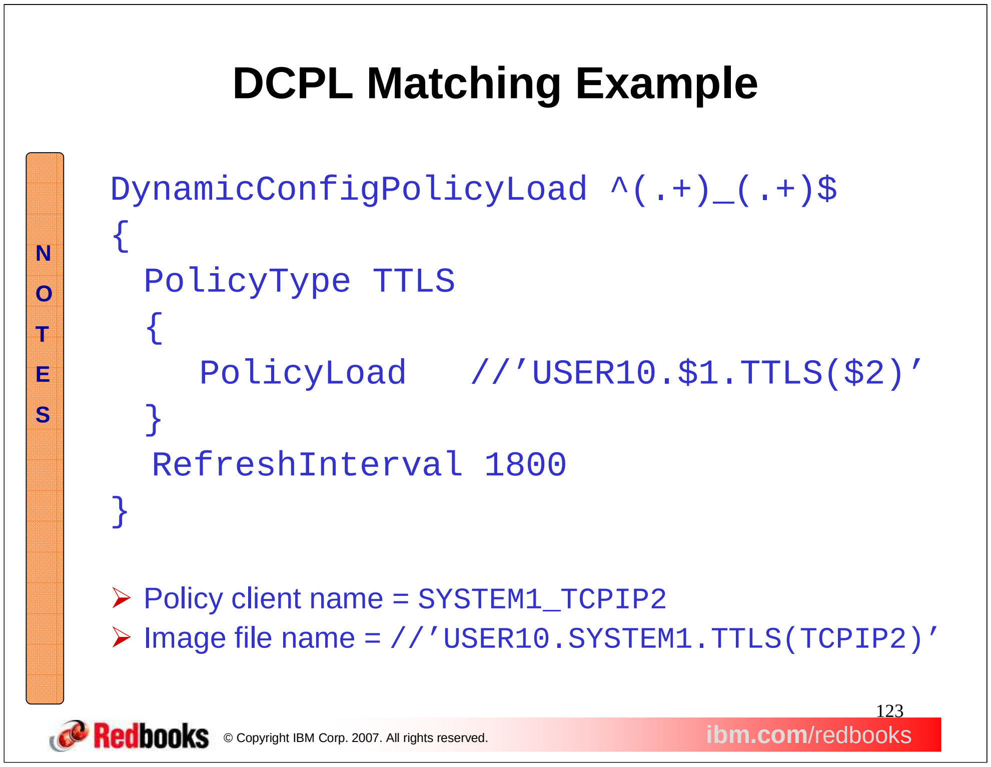
Other symbolic replacement variables can also be used:

- $0 represents the entire portion of the policy client name that matches the regular expression (in other words it isn't limited to a sub-expression).
- * is a wildcard that represents the entire policy client name.

Note that for some regular expressions, $0 and * might resolve to the same value, while for others they won't. If you really want to substitute the entire policy client name, use the * wildcard.

A given policy client is matched to a DCPL statement using a matching hierarchy. The order that the DCPL statements are specified in the configuration file is not important.

1. An exact match between the policy client name and the DCPL statement. In this case the DCPL *clientname* does not contain any regular expression characters (it's just a string, like "client42"). You could use this form to override a more general DCPL statement for a specific policy client, for example.
2. A regular expression match to a DCPL statement. If multiple statements could match, the one with the longest *clientname* parameter is chosen. If multiple matching statements exist with the same length *clientname*, the one chosen is based on alphabetical order of the *clientname*.
3. If no DCPL statement matches, default values are used for the configuration file names and other parameters. The default configuration file names take the form: `/etc/pagent_remote.`*`type`*, where *`type`* is one of the following: `ids, ipsec, qos, routing, ttls.`



DCPL Matching Example

```
DynamicConfigPolicyLoad ^(.+)_(.+)$
{
  PolicyType TTLS
  {
     PolicyLoad   //'USER10.$1.TTLS($2)'
  }
  RefreshInterval 1800
}
```

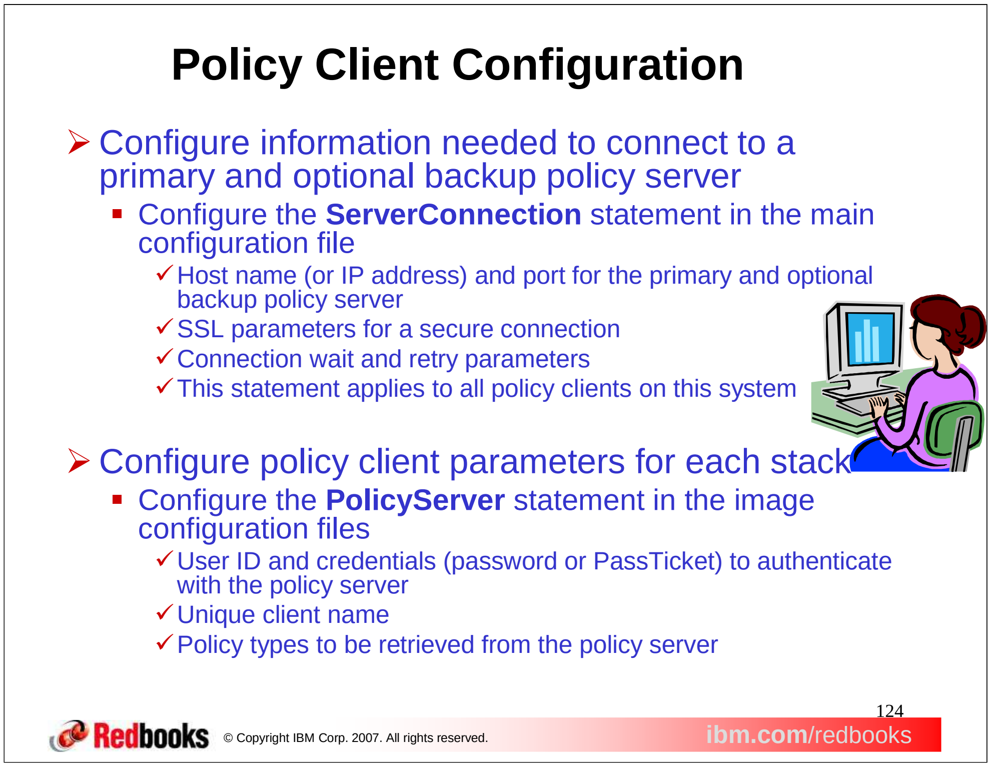
- Policy client name = SYSTEM1_TCPIP2
- Image file name = //'USER10.SYSTEM1.TTLS(TCPIP2)'

So let's put the pieces together. This slide shows a DCPL statement with the *clientname* "`^(.+)_(.+)$`". The ^ (caret) and $ (currency) characters delineate the name. These characters represent the start and end of a string, so using them means the entire policy client name, not just a portion, must match. The regular expression contains 2 sub-expressions contained in parentheses, separated by an underscore. The underscore is not a regular expression character, so it literally matches an underscore in the policy client name. Each sub-expression matches one or more of any character.

The image configuration file name contains 2 symbolic substitution variables, $1 and $2, that correspond to the sub-expressions in the regular expression.

Now, suppose the policy client named "`SYSTEM1_TCPIP2`" connects to the policy server. Since this name consists of 2 strings separated by an underscore, it matches the regular expression on the DCPL statement. The portions of the policy client name that match the 2 sub-expressions replace the symbolic substitution variables, so the resulting image file name is "`//'USER10.SYSTEM1.TTLS(TCPIP2)'`".

A scheme such as this example allows a PDS to be set up for each policy type for each remote system, containing a member for each policy client on those systems. The power of regular expressions and symbolic replacement variables allows many other schemes to also be used.

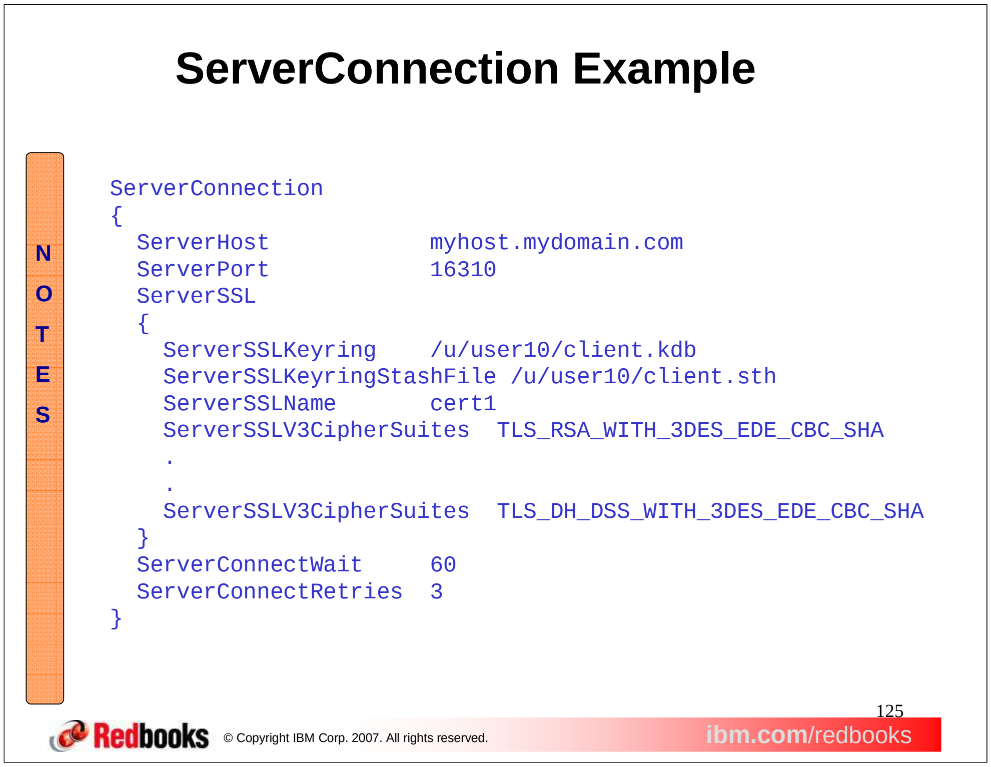


Policy Client Configuration

- Configure information needed to connect to a primary and optional backup policy server
  - Configure the **ServerConnection** statement in the main configuration file
    - ✓ Host name (or IP address) and port for the primary and optional backup policy server
    - ✓ SSL parameters for a secure connection
    - ✓ Connection wait and retry parameters
    - ✓ This statement applies to all policy clients on this system
- Configure policy client parameters for each stack
  - Configure the **PolicyServer** statement in the image configuration files
    - ✓ User ID and credentials (password or PassTicket) to authenticate with the policy server
    - ✓ Unique client name
    - ✓ Policy types to be retrieved from the policy server

The first step in configuring the policy client is to specify the ServerConnection statement in the main configuration file. This statement establishes configuration data that apply to all policy clients on this system. The slide shows the data that can be configured.

For each policy client (TCP/IP stack) on the system, configure a PolicyServer statement in the image configuration file. This statement contains parameters to authenticate the policy client with the policy server, the policy client name, and which policy types to be retrieved. The policy client name must be unique. If a name is not specified then the default of *systemname_stackname* is used. The FLUSH and PURGE parameters can also optionally be configured on the PolicyServer statement.

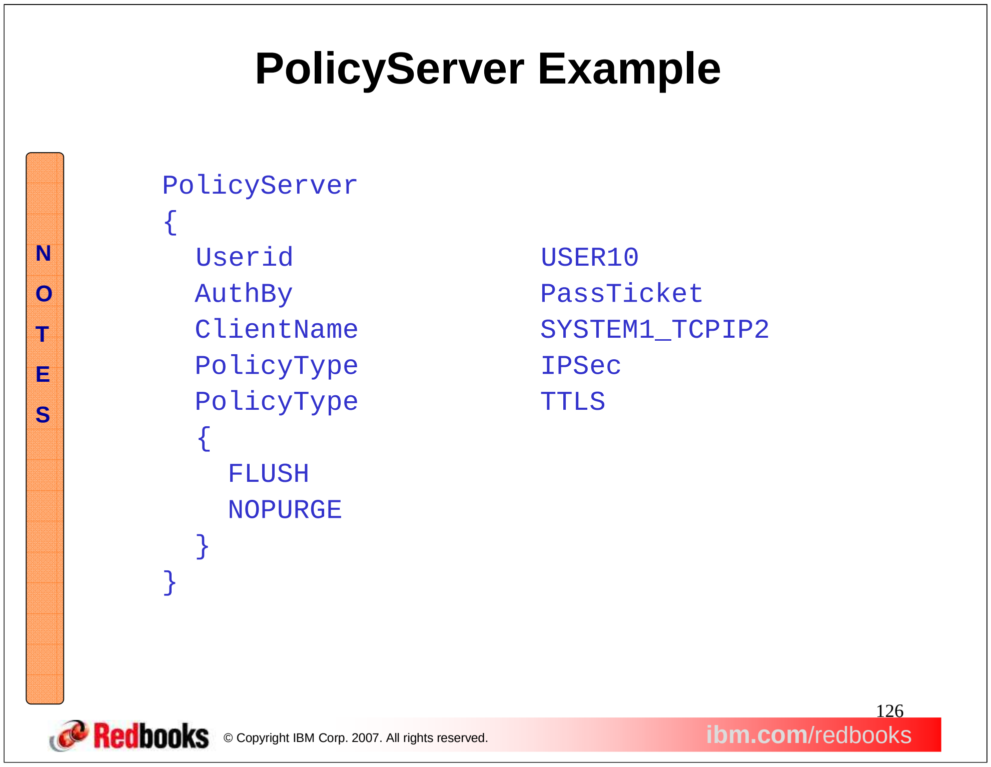


ServerConnection Example

N
O
T
E
S

```
ServerConnection
{
  ServerHost            myhost.mydomain.com
  ServerPort            16310
  ServerSSL
  {
    ServerSSLKeyring    /u/user10/client.kdb
    ServerSSLKeyringStashFile /u/user10/client.sth
    ServerSSLName       cert1
    ServerSSLV3CipherSuites  TLS_RSA_WITH_3DES_EDE_CBC_SHA
    .
    .
    ServerSSLV3CipherSuites  TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
  }
  ServerConnectWait     60
  ServerConnectRetries  3
}
```

Here’s a sample ServerConnection statement.   In this example the primary policy server, located at myhost.mydomain.com and listening on port 16310, is configured. A backup policy server is not configured. The Policy Agent, acting as a policy client will try to connect to the policy server. If the connection attempt fails, then the policy agent will retry 3 times.



PolicyServer Example

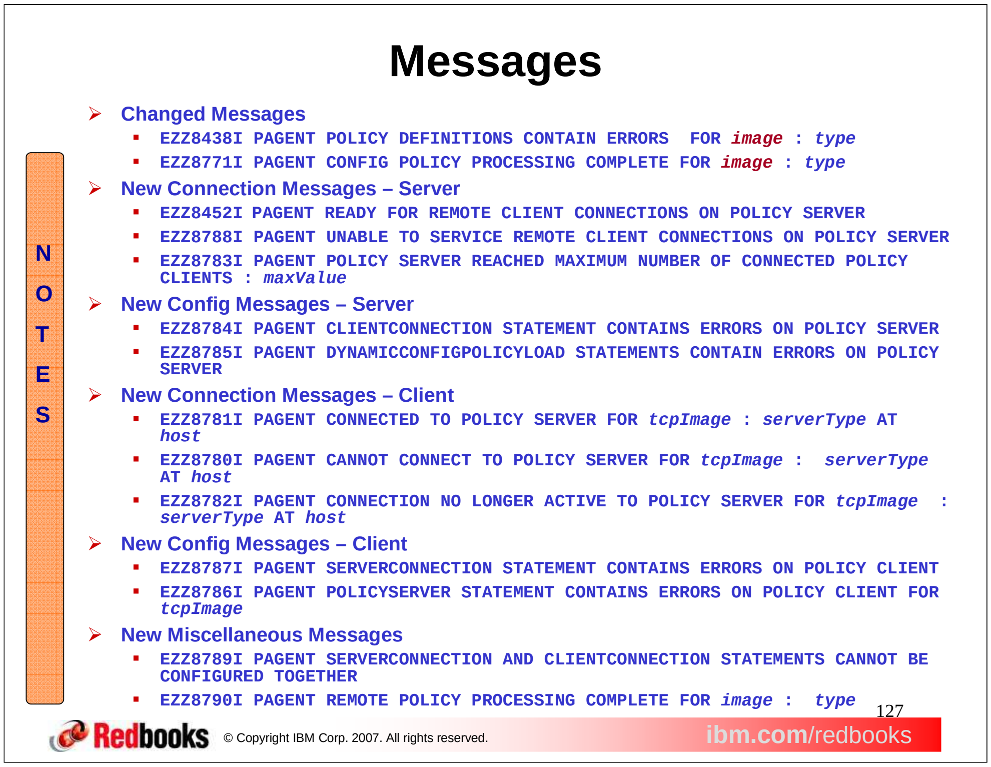
N
O
T
E
S

```
PolicyServer
{
  Userid                 USER10
  AuthBy                 PassTicket
  ClientName             SYSTEM1_TCPIP2
  PolicyType             IPSec
  PolicyType             TTLS
  {
    FLUSH
    NOPURGE
  }
}
```

126

Redbooks ibm.com/redbooks

Here’s a sample PolicyServer statement. Note that the additional set of braces is optional for the PolicyType parameter. If you don’t need to specify the FLUSH/NOFLUSH or PURGE/NOPURGE parameters for a policy type, the braces can be omitted. The FLUSH/NOFLUSH and PURGE/NOPURGE parameters configured on the TcpImage statement are used as defaults.



Messages

- Changed Messages
  - EZZ8438I PAGENT POLICY DEFINITIONS CONTAIN ERRORS FOR *image* : *type*
  - EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR *image* : *type*
- New Connection Messages – Server
  - EZZ8452I PAGENT READY FOR REMOTE CLIENT CONNECTIONS ON POLICY SERVER
  - EZZ8788I PAGENT UNABLE TO SERVICE REMOTE CLIENT CONNECTIONS ON POLICY SERVER
  - EZZ8783I PAGENT POLICY SERVER REACHED MAXIMUM NUMBER OF CONNECTED POLICY CLIENTS : *maxValue*
- New Config Messages – Server
  - EZZ8784I PAGENT CLIENTCONNECTION STATEMENT CONTAINS ERRORS ON POLICY SERVER
  - EZZ8785I PAGENT DYNAMICCONFIGPOLICYLOAD STATEMENTS CONTAIN ERRORS ON POLICY SERVER
- New Connection Messages – Client
  - EZZ8781I PAGENT CONNECTED TO POLICY SERVER FOR *tcpImage* : *serverType* AT *host*
  - EZZ8780I PAGENT CANNOT CONNECT TO POLICY SERVER FOR *tcpImage* : *serverType* AT *host*
  - EZZ8782I PAGENT CONNECTION NO LONGER ACTIVE TO POLICY SERVER FOR *tcpImage* : *serverType* AT *host*
- New Config Messages – Client
  - EZZ8787I PAGENT SERVERCONNECTION STATEMENT CONTAINS ERRORS ON POLICY CLIENT
  - EZZ8786I PAGENT POLICYSERVER STATEMENT CONTAINS ERRORS ON POLICY CLIENT FOR *tcpImage*
- New Miscellaneous Messages
  - EZZ8789I PAGENT SERVERCONNECTION AND CLIENTCONNECTION STATEMENTS CANNOT BE CONFIGURED TOGETHER
  - EZZ8790I PAGENT REMOTE POLICY PROCESSING COMPLETE FOR *image* : *type*

127

Redbooks ibm.com/redbooks

The configuration chores are complete. Now let’s take a look at new and changed messages. Shown here are the only messages that changed. The change is to replace the TCP/IP stack name variable with the more generic *image* variable. The *image* can identify either a TCP/IP stack or a policy client name.

This slide also shows connection-oriented messages that can be issued on the policy server. Note that the EZZ8783I message is issued if more than the maximum number of policy clients try to connect to the policy server, so you are unlikely to see this message. The EZZ8452I message indicates that the ClientConnection statement is properly configured and the Policy Agent is listening for remote connections. If for any reason the Policy Agent is unable to listen for remote connections, but the ClientConnection statement is configured – for example no TCP/IP stacks are started – you’ll see message EZZ8788I

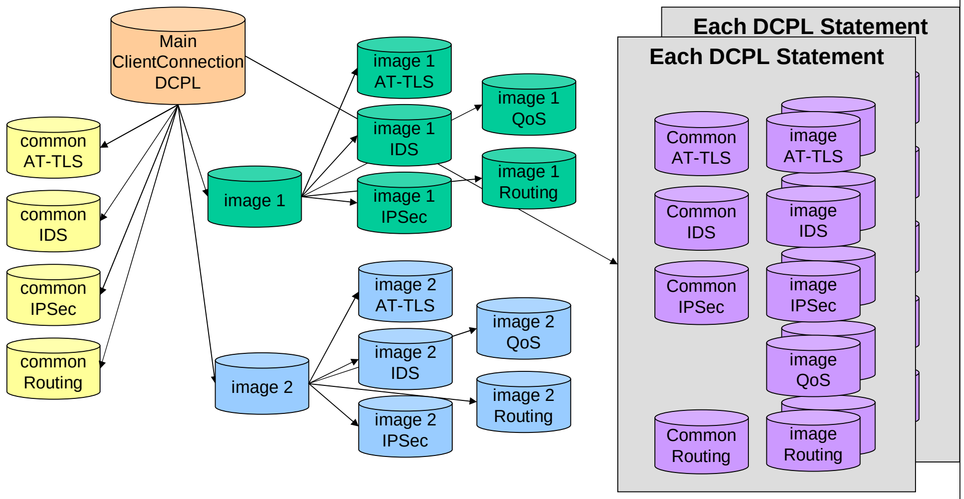
Messages EZZ8784I and EZZ8785I indicate problems with the new configuration statements for the policy server, CLIENTCONNECTION and DYNAMICCONFIGPOLICYLOAD.

As with the policy server, there are some connection-related messages issued on the policy client. The first 2 messages shown indicate success or failure when trying to connect to a policy server. The EZZ8782I message is issued if an active connection to the policy server ends. In most cases, the policy client will retry the connections to the primary and backup policy servers.

Messages EZZ8787I and EZZ8786I indicate problems with the new configuration statements for the policy client, SERVERCONNECTION and POLICYSERVER.

If you try to configure the Policy Agent as both a policy server and a policy client, you’ll see the EZZ8789I message. Message EZZ8790I is a companion to the existing EZZ8771I message. The new message is issued to the client console to distinguish remote policies from local policies (the existing message is issued for local policies). The new message is also issued on the policy server, but only to the log file, to avoid flooding the console if a large number of policy clients exist.

Policy Server Configuration Pictorial View



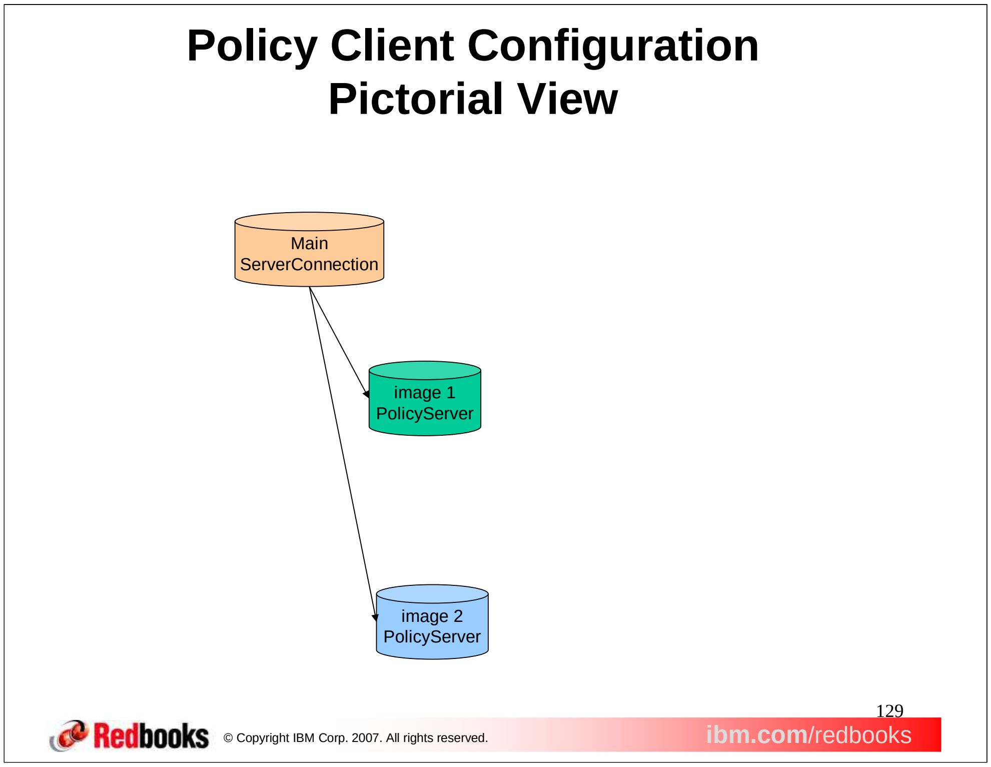
That concludes the changes to the externals for centralized policy services. Now let’s look at some more details of the solution.

This slide graphically shows the changed configuration on the policy server. The left side of the slide is the same as the earlier slide that showed the configuration prior to V1R9. All of that configuration can still be used to configure local policies on the policy server.

The ClientConnection statement and one or more DynamicConfigPolicyLoad statements need to be configured in the main configuration file, in order to provide policy services for policy clients.

The gray boxes show the new configuration files needed to define centralized policies. Each gray box represents one DCPL statement, which in turn provides configuration for a set of policy clients. There is one common configuration file for all policy types except QoS. There are one or more image configuration files for each policy type. If the image file names on the DCPL statement use symbolic replacement or wildcard variables, then each policy client using a given DCPL statement needs a separate image configuration file.

More configuration is required, but is centralized on one system. Only one instance of the IBM Configuration Assistant is needed (if used), and it only needs to connect to the policy server.



Policy Client Configuration Pictorial View

Main ServerConnection

image 1 PolicyServer

image 2 PolicyServer

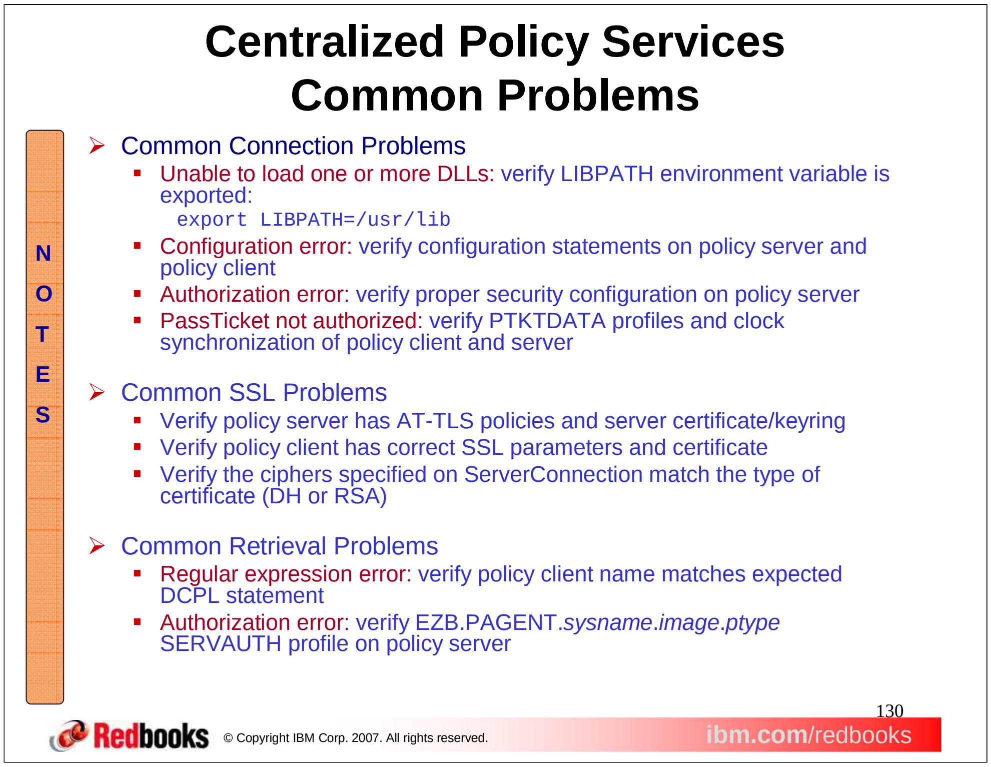
129

ibm.com/redbooks

This slide shows the changed configuration on the policy client, if all policy types are retrieved from the policy server. If some local policies are still needed, some subset of the left side of the previous slide will still be needed.

All that’s needed if all policy types are retrieved remotely is the main configuration file and an image configuration file for each TCP/IP stack.

The ServerConnection statement must be configured in the main configuration file, and the PolicyServer statement must be configured in each image configuration file, in order to retrieve policies from the policy server.



Centralized Policy Services Common Problems

NOTES

- Common Connection Problems
  - Unable to load one or more DLLs: verify LIBPATH environment variable is exported:
    ```
    export LIBPATH=/usr/lib
    ```
  - Configuration error: verify configuration statements on policy server and policy client
  - Authorization error: verify proper security configuration on policy server
  - PassTicket not authorized: verify PTKTDATA profiles and clock synchronization of policy client and server
- Common SSL Problems
  - Verify policy server has AT-TLS policies and server certificate/keyring
  - Verify policy client has correct SSL parameters and certificate
  - Verify the ciphers specified on ServerConnection match the type of certificate (DH or RSA)
- Common Retrieval Problems
  - Regular expression error: verify policy client name matches expected DCPL statement
  - Authorization error: verify EZB.PAGENT.*sysname.image.ptype* SERVAUTH profile on policy server

130

Redbooks ibm.com/redbooks

This slide shows some common connection problems that you might encounter. See IP Diagnosis for more details. For the PassTicket not authorized problem see z/OS Security Server RACF Security Administrator's Guide, Single signon function.

The connections use SSL, and here are some of the the common SSL problems that you might encounter. A common error is incorrect SSL configuration for the policy client. If the server AT-TLS policy uses HandshakeRole Server, ServerConnection ServerSSLName parameter must specify the server's certificate. If server AT-TLS policy uses HandshakeRole ServerWithClientAuth, ServerConnection ServerSSLName parameter must specify the client's certificate. See IP Diagnosis for more details. Also see IP Configuration Guide, Appendix B. TLS/SSL Security for detailed information on certificates and establishing a correct SSL environment.

There are some common policy retrieval problems that you might encounter. The policy server logs a message for a successful match of a client name to a DCPL statement as well as an unsuccessful match. If a regular expression error is encountered then view the messages in the policy server's log file. Following are examples of a log messages:

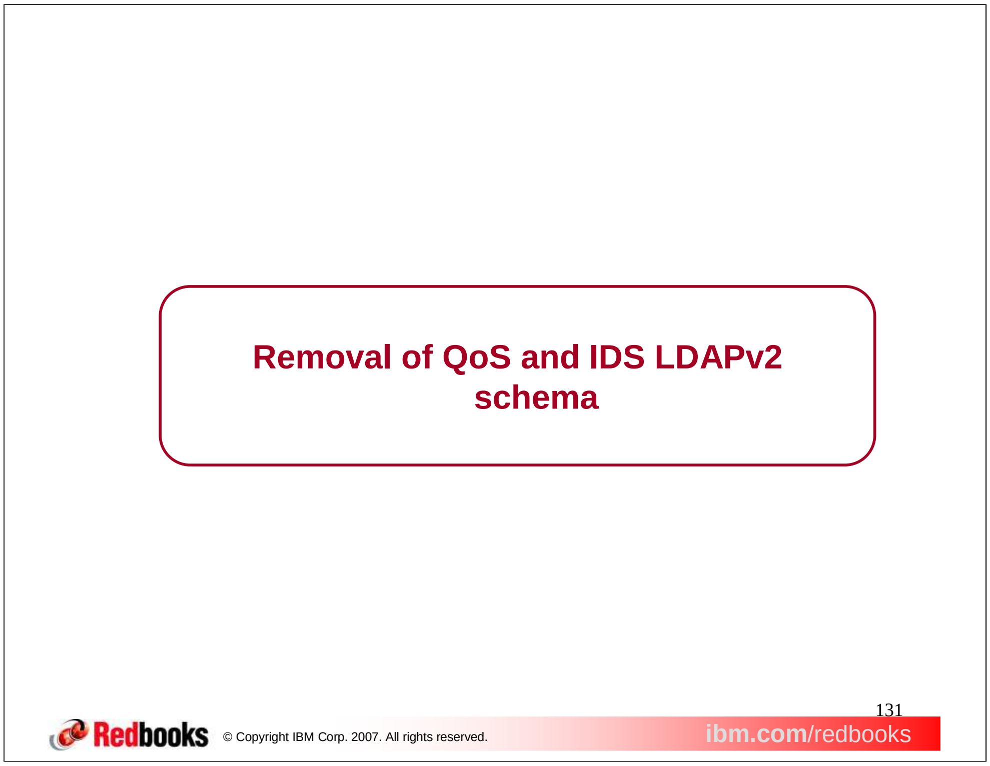
Log message for successful match:

```
client PEP 'clientJF' using DynamicConfigPolicyLoad statement
'client(.*)'
```

Log message for unsuccessful match:

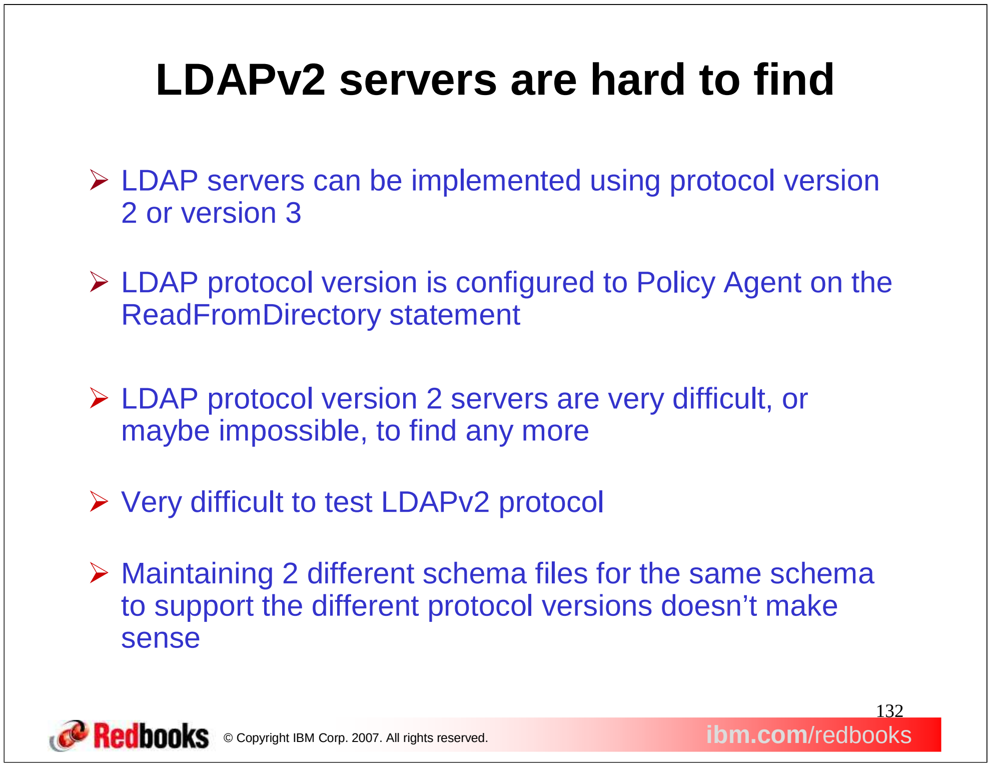
```
client PEP 'ClientJF' doesn't match any DynamicConfigPolicyLoad
statement, using defaults for all disciplines
```

See IP Diagnosis for more details.



Removal of QoS and IDS LDAPv2 schema

This section describes the removal of QoS and IDS LDAPv2 schema

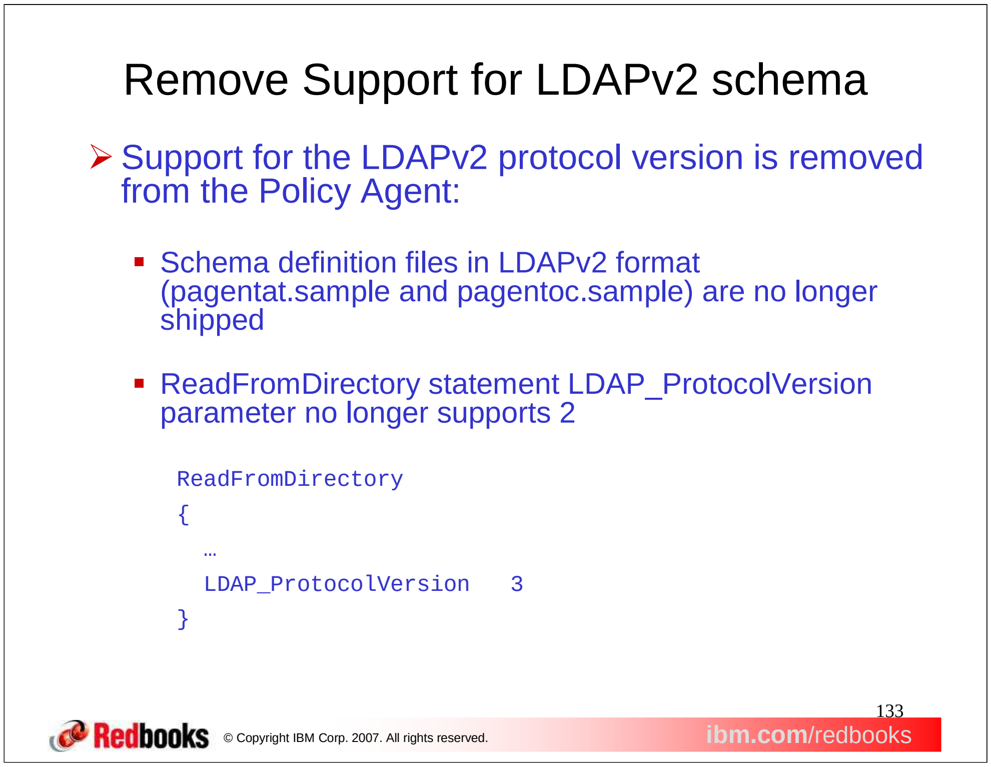


LDAPv2 servers are hard to find

- LDAP servers can be implemented using protocol version 2 or version 3
- LDAP protocol version is configured to Policy Agent on the ReadFromDirectory statement
- LDAP protocol version 2 servers are very difficult, or maybe impossible, to find any more
- Very difficult to test LDAPv2 protocol
- Maintaining 2 different schema files for the same schema to support the different protocol versions doesn't make sense

The Policy Agent can use an LDAP server to contain local QoS and IDS policies. LDAP servers can use either protocol version 2 (LDAPv2) or protocol version 3 (LDAPv3). Protocol version 3 provides many advantages compared to protocol version 2.

The problem with protocol version 2 servers is that most vendors no longer support that protocol version. It is becoming increasingly difficult to even find LDAPv2 servers to test with. In this environment, it no longer makes sense to continue to support LDAPv2 for the Policy Agent.



Remove Support for LDAPv2 schema

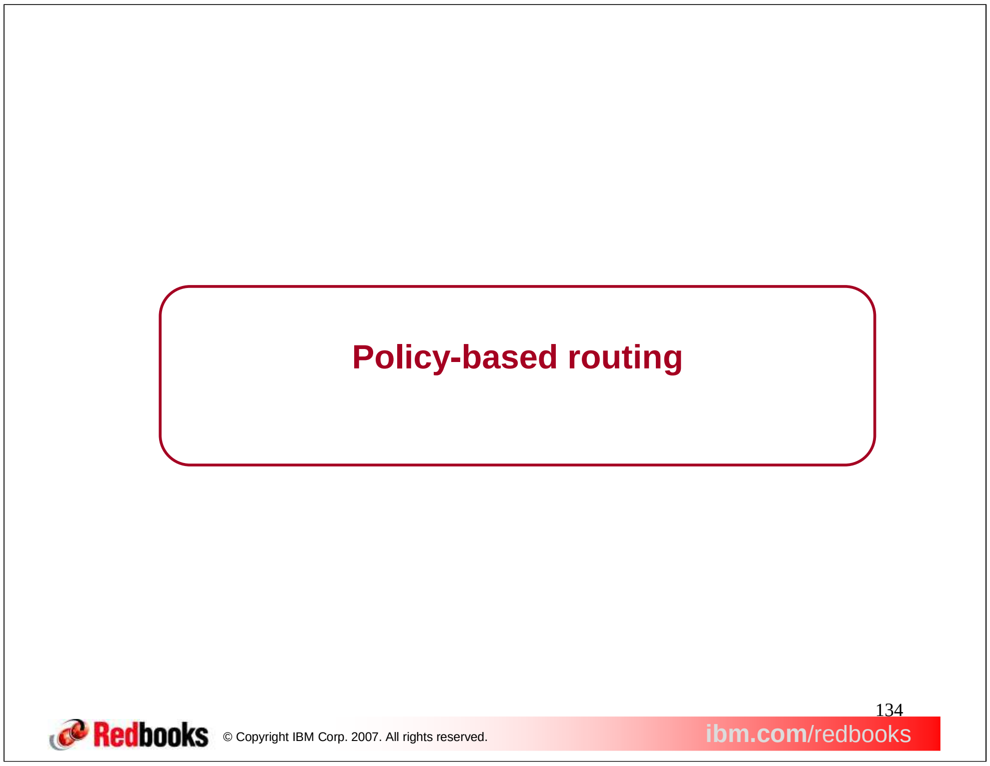
- Support for the LDAPv2 protocol version is removed from the Policy Agent:
  - Schema definition files in LDAPv2 format (pagentat.sample and pagentoc.sample) are no longer shipped
  - ReadFromDirectory statement LDAP_ProtocolVersion parameter no longer supports 2

```
ReadFromDirectory
{
  …
  LDAP_ProtocolVersion   3
}
```

133

ibm.com/redbooks

So, support for LDAPv2 is dropped with V1R9. This was announced in a previous release. The protocol version is configured on the ReadFromDirectory statement, and no longer supports version 2.

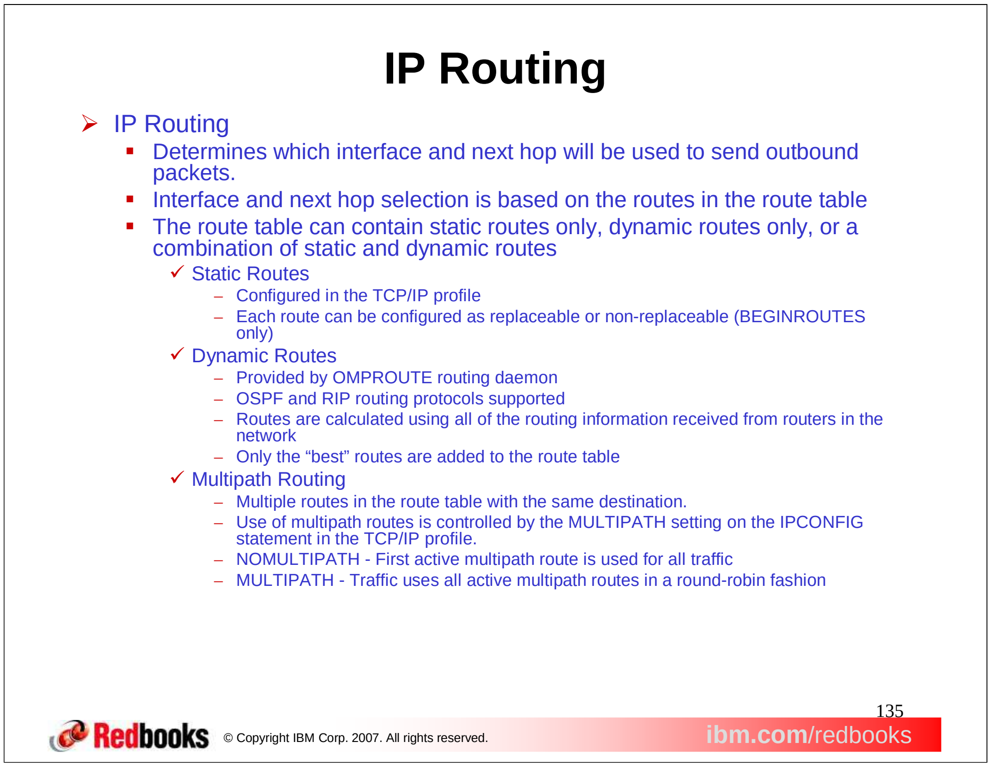


Policy-based routing

134

ibm.com/redbooks

This section covers the policy-based routing function added to z/OS Communications Server V1R9.



IP Routing

- IP Routing
  - Determines which interface and next hop will be used to send outbound packets.
  - Interface and next hop selection is based on the routes in the route table
  - The route table can contain static routes only, dynamic routes only, or a combination of static and dynamic routes
    - ✓ Static Routes
      - – Configured in the TCP/IP profile
      - – Each route can be configured as replaceable or non-replaceable (BEGINROUTES only)
    - ✓ Dynamic Routes
      - – Provided by OMPROUTE routing daemon
      - – OSPF and RIP routing protocols supported
      - – Routes are calculated using all of the routing information received from routers in the network
      - – Only the “best” routes are added to the route table
    - ✓ Multipath Routing
      - – Multiple routes in the route table with the same destination.
      - – Use of multipath routes is controlled by the MULTIPATH setting on the IPCONFIG statement in the TCP/IP profile.
      - – NOMULTIPATH - First active multipath route is used for all traffic
      - – MULTIPATH - Traffic uses all active multipath routes in a round-robin fashion

135

Redbooks

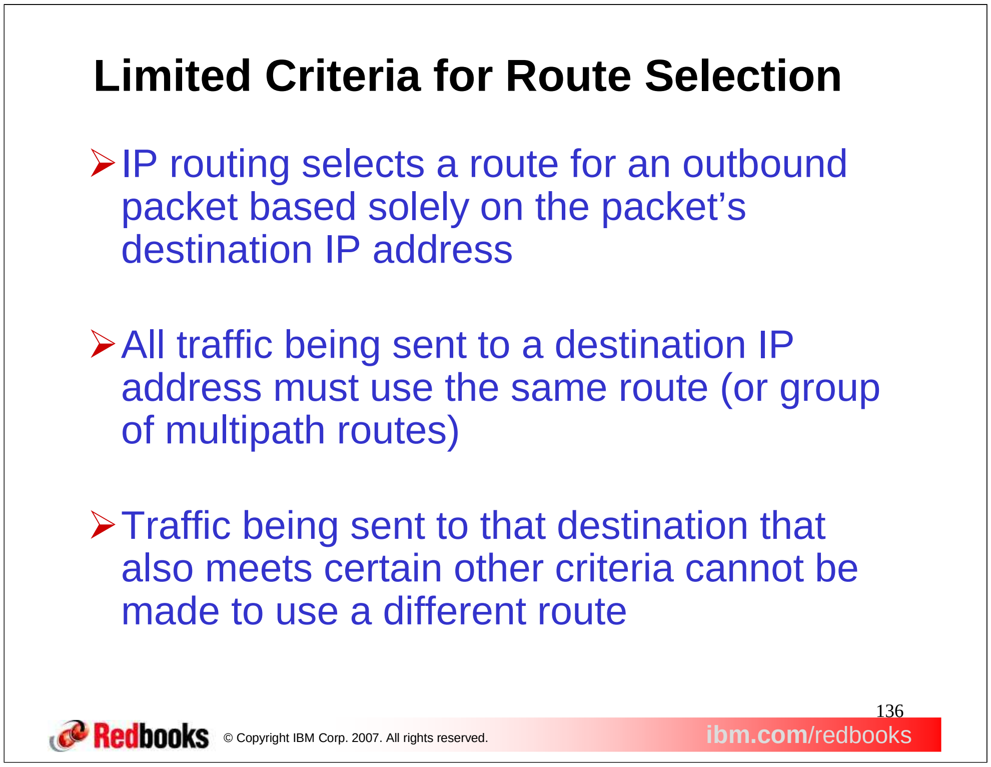
ibm.com/redbooks

IP Routing is the TCP/IP stack function that uses a table of routes to determine which interface and next hop will be used for IP traffic that is leaving the stack. The IP route table may contain only static routes, or it may contain only dynamic routes, or it may contain a combination of the two. It also may contain a combination of routes for single destinations, known as host routes, routes for all destinations in a IP subnet, IP network, or IP supernet, and routes that can be used for any destination, known as default routes. When IP Routing is searching the route table for a route to be used for sending traffic to a destination, it searches for the first active route that includes the destination IP address, in the order host, subnet, network, supernet, default.

Static routes in the z/OS Communications Server route table are configured in the TCP/IP profile using either the BEGINROUTES statement or the GATEWAY statement. The BEGINROUTES statement is an alternative to the GATEWAY statement that allows addresses to be specified using a BSD style syntax and that has some enhancements that are not available with the GATEWAY statement. When the BEGINROUTES statement is used, each static route can be configured as either replaceable or non-replaceable. This setting determines whether the route can be replaced by dynamic routes that are learned by the OMPROUTE routing daemon. Replaceable routes can be replaced by dynamic routes learned by OMPROUTE. Non-replaceable routes cannot be replaced by dynamic routes learned by OMPROUTE.

Dynamic routes in the z/OS Communications Server route table are provided by the routing daemon OMPROUTE. OMPROUTE uses information learned from routers in the network, via either the OSPF or RIP routing protocol, to calculate the dynamic routes. This information may provide many different routes to network destinations, but only the “best” routes to each destination are added to the stack route table. The “best” routes are determined by assigning cost values to each route, based on configuration information within OMPROUTE and the network routers.

Multipath is a function of IP Routing that determines the processing to be performed when there are multiple routes in the route table to the same destination. There can be multiple dynamic routes added to the same destination, as mentioned on the previous slide, or you can configure multiple static routes to the same destination. All static routes to a destination are considered to have the same cost and all are added to the route table. When the routing information learned by OMPROUTE results in multiple dynamic routes to a destination with the same cost, all are added to the route table. The Multipath function is controlled by a setting on the IPCONFIG statement in the TCP/IP profile. When NOMULTIPATH is specified on the IPCONFIG statement and multiple routes to a destination are in the route table, all traffic sent to that destination uses the first active route to the destination. The other routes to the destination provide backup should the first route become inactive, but they are not used as long as the first route is active. When MULTIPATH is specified on the IPCONFIG statement and multiple routes to a destination are in the route table, each of the active routes are used in a round-robin fashion. The method in which they are used is controlled by a qualifier configured on the MULTIPATH parameter. The possible values for this qualifier are PERConnection and PERPacket. When PERConnection is specified then for TCP, the next multipath route is selected for each new connection. The connection uses that route as long as the route is usable. For UDP/RAW, the next multipath route is selected for each new route lookup. When PERPacket is specified then the next multipath route is selected for each packet sent.



Limited Criteria for Route Selection

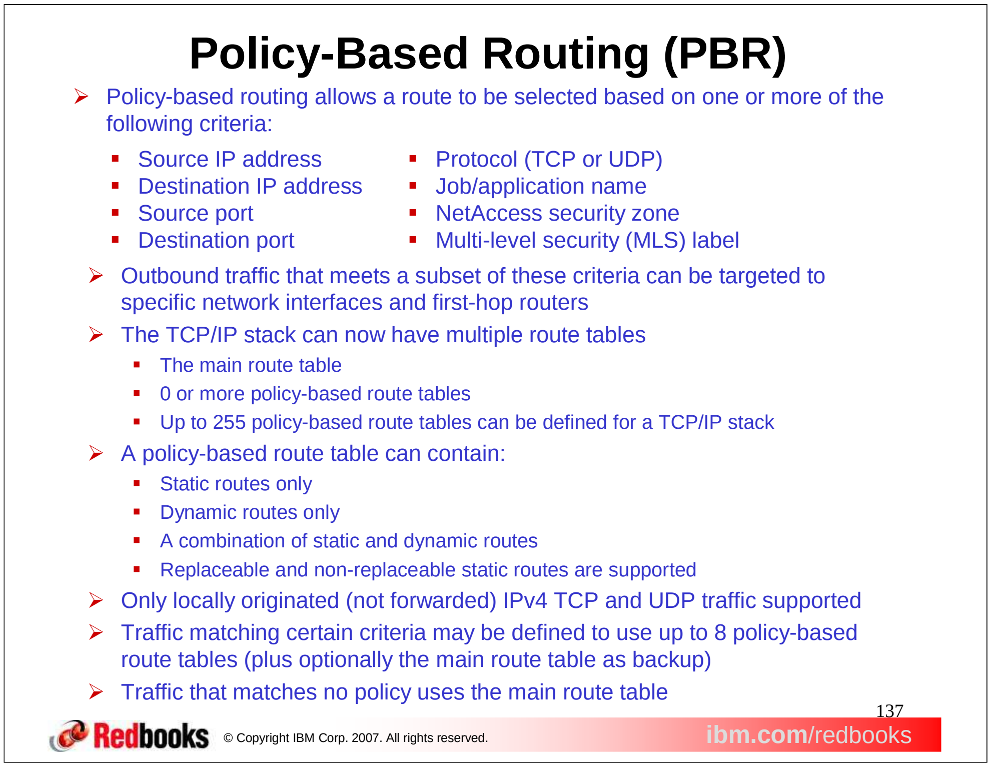
- IP routing selects a route for an outbound packet based solely on the packet’s destination IP address
- All traffic being sent to a destination IP address must use the same route (or group of multipath routes)
- Traffic being sent to that destination that also meets certain other criteria cannot be made to use a different route

136

Redbooks ibm.com/redbooks

As covered in the Background section, when IP Routing needs to send outbound traffic, it searches the route table for a route that matches the destination IP address of the traffic. This may be a host route specific to the destination address, or a route to a subnet, network, or supernet containing the destination address, or it may be the default route which covers all destination addresses.

A limitation that has existed with IP Routing is due to the fact that only the destination IP address could be used when selecting a route for outbound traffic. All traffic destined for a particular IP address had to use the same route or group of multipath routes. There has been no way to use different routes for different purposes such as for FTP traffic, for secure traffic, for Enterprise Extender traffic, etc.



Policy-Based Routing (PBR)

- Policy-based routing allows a route to be selected based on one or more of the following criteria:
  - Source IP address
  - Destination IP address
  - Source port
  - Destination port
  - Protocol (TCP or UDP)
  - Job/application name
  - NetAccess security zone
  - Multi-level security (MLS) label
- Outbound traffic that meets a subset of these criteria can be targeted to specific network interfaces and first-hop routers
- The TCP/IP stack can now have multiple route tables
  - The main route table
  - 0 or more policy-based route tables
  - Up to 255 policy-based route tables can be defined for a TCP/IP stack
- A policy-based route table can contain:
  - Static routes only
  - Dynamic routes only
  - A combination of static and dynamic routes
  - Replaceable and non-replaceable static routes are supported
- Only locally originated (not forwarded) IPv4 TCP and UDP traffic supported
- Traffic matching certain criteria may be defined to use up to 8 policy-based route tables (plus optionally the main route table as backup)
- Traffic that matches no policy uses the main route table

Redbooks

ibm.com/redbooks

Policy-based routing addresses this limitation of IP Routing by allowing a route to be selected based on much more than just the destination IP address. The additional route selectors, which are listed on this slide, can be used to cause traffic that meets more specific criteria to be targeted to specific network interfaces and first-hop routers.

So, how does policy-based routing allow IP Routing to use these additional route selectors?

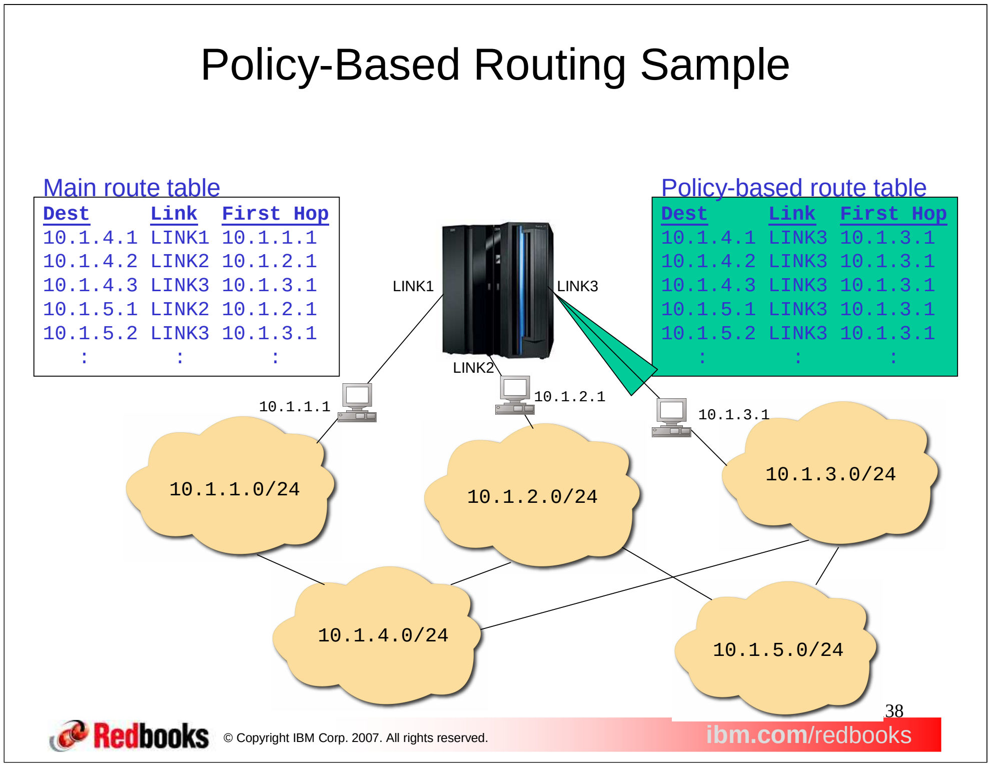
It is made possible through the use of multiple route tables. In addition to the main route table, the TCP/IP stack can now have multiple policy-based route tables. Policy-based route tables have many of the same characteristics as the main route table. They can contain both static and dynamic routes and the static routes can be configured as both replaceable and non-replaceable.

Policy-based routing is not supported for all types of IP traffic. The support is limited to locally originated IPv4 TCP and UDP traffic. All IPv6 traffic, all forwarded traffic, and all traffic using protocols other than TCP and UDP is not processed by policy-based routing and continues to be routed using only the main route table. For example, ICMP Echo request packets sent by the Ping command will continue to be routed using the main route table.

Each policy-based route table can be configured such that all static routes and dynamic routes that it contains are limited to specific links and next hop routers. Static routes are limited to those links and next hops simply by using only those links and next hops on any static routes configured for the policy-based route table. For dynamic routing, where the routes will be added to the route table by OMPROUTE, it is necessary to control the way in which OMPROUTE computes those routes. This is done, for each policy-based route table, through the configuration of dynamic routing parameters which identify the link and next hops that OMPROUTE may use when computing routes for the table.

Once you have policy-based route tables created, how do you cause different types of traffic to use the different route tables?

Using policy agent, policy can be written which indicates that traffic that matches particular combinations of the various route selection criteria will be routed using certain route tables. A particular type of traffic can be defined to use up to 8 policy-based route tables, plus the main route table as backup. Traffic that matches none of the defined policies continues to be routed using the main route table.



Policy-Based Routing Sample

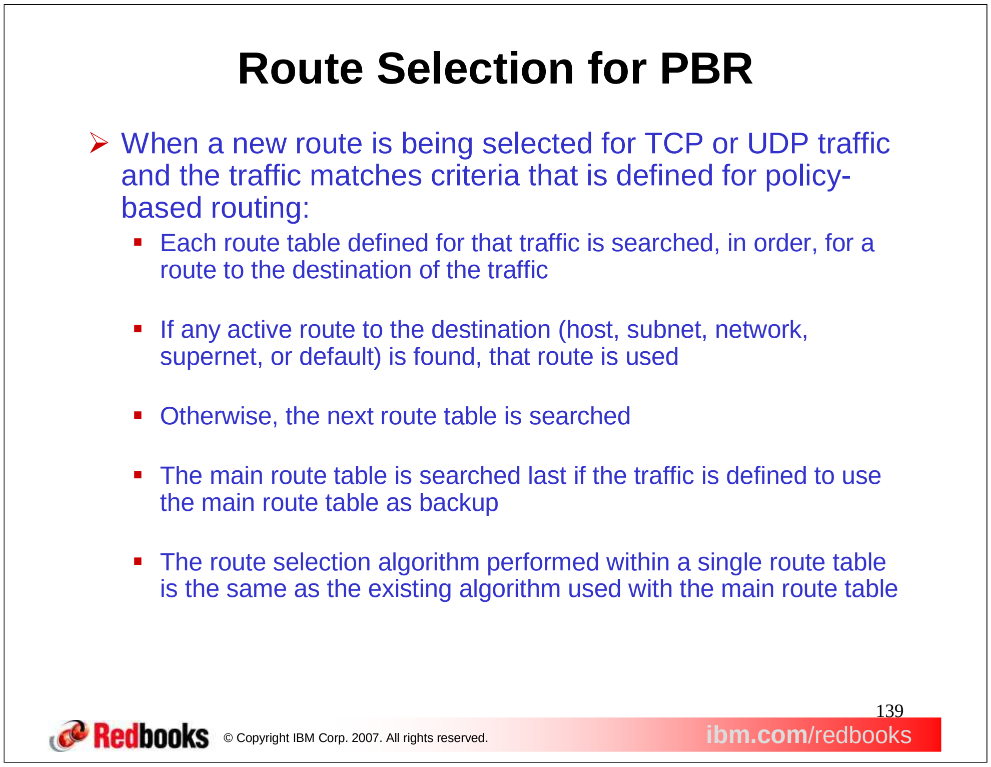
Main route table

Dest	Link	First Hop
10.1.4.1	LINK1	10.1.1.1
10.1.4.2	LINK2	10.1.2.1
10.1.4.3	LINK3	10.1.3.1
10.1.5.1	LINK2	10.1.2.1
10.1.5.2	LINK3	10.1.3.1
:	:	:

Policy-based route table

Dest	Link	First Hop
10.1.4.1	LINK3	10.1.3.1
10.1.4.2	LINK3	10.1.3.1
10.1.4.3	LINK3	10.1.3.1
10.1.5.1	LINK3	10.1.3.1
10.1.5.2	LINK3	10.1.3.1
:	:	:

In this sample, we have a node connected to a set of IP subnets. You can see, in the partial table shown, that the main route table contains routes to destinations throughout the network and that these routes use all of the three available network links. These may be routes that were added to the main route table by OMPROUTE, in which case the location of the destinations and the dynamic routing configuration throughout the network has resulted in these routes being the best routes available. If there is a need for a certain type of IP traffic (for example all traffic sent by a specific job name) to be sent out LINK3, a policy-based route table such as the one shown could be created. In this particular case, the policy-based route table contains routes to all of the same destinations as are in the main route table. However, all of the routes in the policy-based route table use LINK3.



Route Selection for PBR

- When a new route is being selected for TCP or UDP traffic and the traffic matches criteria that is defined for policy-based routing:
  - Each route table defined for that traffic is searched, in order, for a route to the destination of the traffic
  - If any active route to the destination (host, subnet, network, supernet, or default) is found, that route is used
  - Otherwise, the next route table is searched
  - The main route table is searched last if the traffic is defined to use the main route table as backup
  - The route selection algorithm performed within a single route table is the same as the existing algorithm used with the main route table

Once the set of route tables that can be used for some type of outbound traffic has been determined, how does IP Routing search for a route in those tables?

Most often there will be one policy-based route table defined to be used for the traffic, but there may be as many as eight. Each of the policy-based route tables is searched, in the order defined, for a route to the destination. If any active route to the destination is found in a route table, the search is stopped and that route is used for the traffic. This route may be a host route, a subnet, network, or supernet route, or a default route. If no active route to the destination is found in a route table, the search continues with the next route table. If all policy-based route tables are searched without success, the main route table may also be searched if the policy indicates that the main route table can be used as a backup.

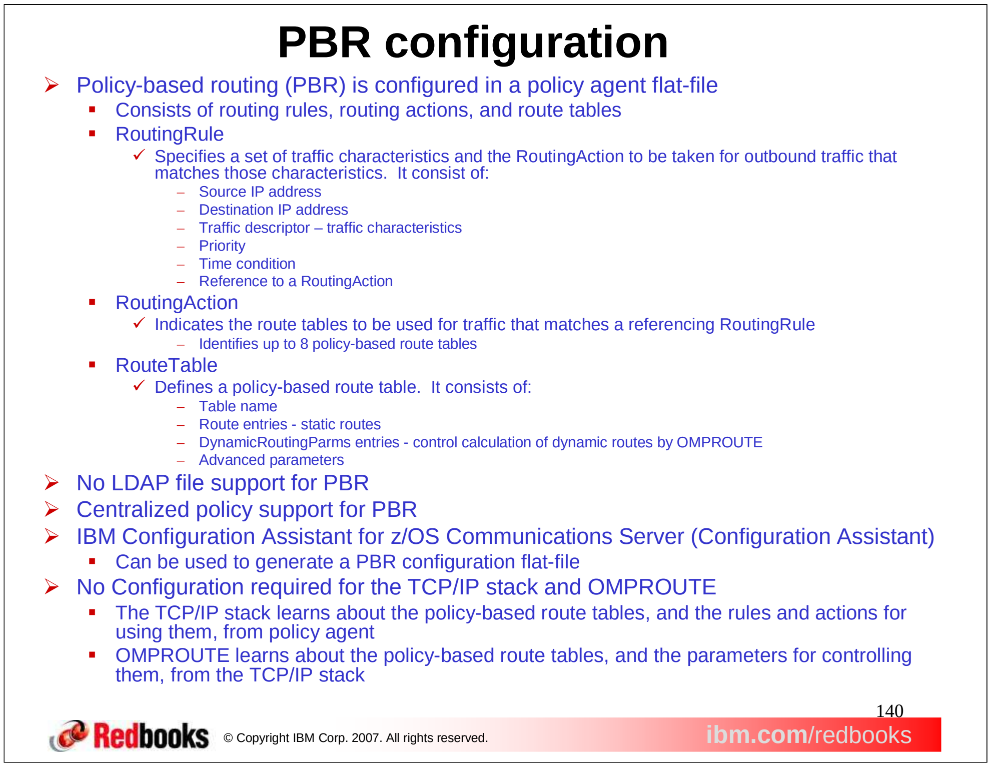
Route selection within a route table occurs in the following order:

•If a route exists to the destination address (a host route), it is chosen.

•If no host route exists to the destination address:

> •If subnet, network, or supernet routes exist to the destination, the route with the most specific network mask (the mask with the most bits on) is chosen.
>
> •If the destination is a multicast destination and a multicast default route exists, that route is chosen.

Default routes are chosen when no other route exists to a destination.



PBR configuration

- Policy-based routing (PBR) is configured in a policy agent flat-file
  - Consists of routing rules, routing actions, and route tables
  - RoutingRule
    - ✓ Specifies a set of traffic characteristics and the RoutingAction to be taken for outbound traffic that matches those characteristics. It consist of:
      - Source IP address
      - Destination IP address
      - Traffic descriptor – traffic characteristics
      - Priority
      - Time condition
      - Reference to a RoutingAction
  - RoutingAction
    - ✓ Indicates the route tables to be used for traffic that matches a referencing RoutingRule
      - Identifies up to 8 policy-based route tables
  - RouteTable
    - ✓ Defines a policy-based route table. It consists of:
      - Table name
      - Route entries - static routes
      - DynamicRoutingParms entries - control calculation of dynamic routes by OMPROUTE
      - Advanced parameters
- No LDAP file support for PBR
- Centralized policy support for PBR
- IBM Configuration Assistant for z/OS Communications Server (Configuration Assistant)
  - Can be used to generate a PBR configuration flat-file
- No Configuration required for the TCP/IP stack and OMPROUTE
  - The TCP/IP stack learns about the policy-based route tables, and the rules and actions for using them, from policy agent
  - OMPROUTE learns about the policy-based route tables, and the parameters for controlling them, from the TCP/IP stack

140

Redbooks

ibm.com/redbooks

Policy-based routing is configured in a policy agent flat file and it is supported by the centralized policy function. You can manually create the policy agent flat-file or you can use the IBM Configuration Assistant for z/OS Communications Server to generate the file. The items configured for policy-based routing consist of routing rules, routing actions, and route tables.

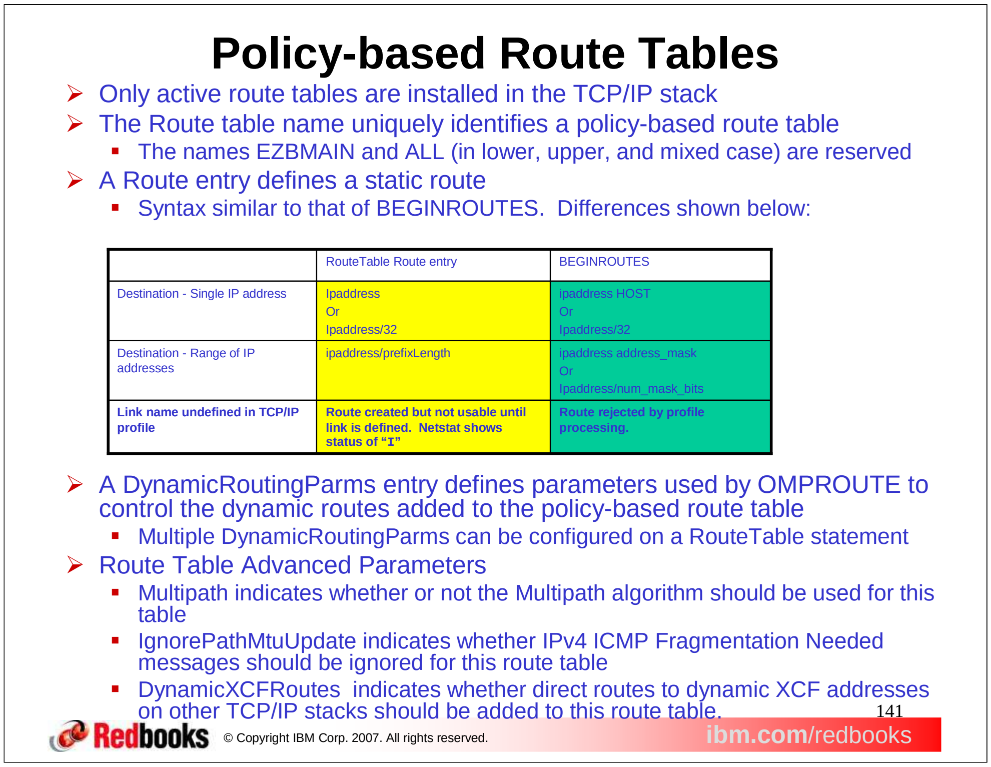
The RoutingRule statement is where you will identify a type of traffic that you want to be routed using policy-based routing. The traffic can be identified by any combination of its source IP address, its destination IP address, and any of a set of other traffic characteristics in the traffic descriptor. Each RoutingRule can be given a priority and a time condition. Lastly, each RoutingRule will reference a RoutingAction which will define the action to be taken for the traffic. The source and destination IP address, if specified in a RoutingRule, indicate the source or destination IP address used in the type of outbound traffic being defined. The source IP address for an outbound TCP connection or an outbound UDP packet can be influenced by a number of configuration and application options. See the source IP address information in *z/OS Communications Server: IP Configuration Guide* for the hierarchy of ways that the source IP address of an outbound packet is determined.

The traffic descriptor defines the remainder of the characteristics that can be used to identify a type of traffic that will use policy-based routing. If the traffic descriptor is used to identify a particular type of traffic, it can be configured inline in the RoutingRule or the RoutingRule can reference one or more previously configured traffic descriptors. A traffic descriptor can specify any combination of source and destination port, traffic protocol, job name of the sending application, NetAccess security zone of the traffic, and MLS security label of the traffic. An outbound packet's destination IP address is used to determine the packet's NetAccess security zone in the NetAccess table defined in the TCP/IP profile. The MLS security label is the label associated with the NetAccess security zone. Each RoutingRule can be configured with a priority value, which is used to select a rule for outbound traffic when the traffic could match the characteristics configured for multiple rules. If these rules are not configured with different priority values, the precedence of the rules is unpredictable. Note that rule priority is not explicitly configured when the IBM Configuration Assistant for z/OS Communications Server is used to configure policy-based routing. In that case, rule priority is determined by the order of the rules as shown on the rules panel. The time condition of a RoutingRule controls when the rule is active and installed in the TCP/IP stack. The reference to a RoutingAction provides a link to the RoutingAction that will define the action to be taken for the type of traffic defined by the RoutingRule.

The RoutingAction statement is where you will identify the set of policy-based route tables that will be used to route traffic that you have defined with a RoutingRule statement. A RoutingAction can specify up to eight policy-based route tables that will be searched, in order, to find a route to the destination of the traffic. In addition, the RoutingAction may indicate that the main route table should also be searched when a usable route is not found in any of the policy-based route tables specified.

The RouteTable statement is where you will define the characteristics of each policy-based route table. To define a policy-based route table, you will need to provide a name for the table. In addition, you may define static routes to be added to the table, dynamic routing parameters to control the dynamic routes that will added to the table by OMPROUTE, as well as a few advanced table parameters.

Good news! There are no changes needed in either the TCP/IP stack configuration or the OMPROUTE configuration for policy-based routing. The TCP/IP stack learns, from policy agent, about the policy rules, policy actions, and policy-based route tables that you have configured. OMPROUTE learns, from the TCP/IP stack, about the policy-based route tables that you have configured to use dynamic routing.



Policy-based Route Tables

- Only active route tables are installed in the TCP/IP stack
- The Route table name uniquely identifies a policy-based route table
  - The names EZBMAIN and ALL (in lower, upper, and mixed case) are reserved
- A Route entry defines a static route
  - Syntax similar to that of BEGINROUTES. Differences shown below:

	RouteTable Route entry	BEGINROUTES
Destination - Single IP address	Ipaddress Or Ipaddress/32	ipaddress HOST Or Ipaddress/32
Destination - Range of IP addresses	ipaddress/prefixLength	ipaddress address_mask Or Ipaddress/num_mask_bits
Link name undefined in TCP/IP profile	**Route created but not usable until link is defined. Netstat shows status of "I"**	**Route rejected by profile processing.**

- A DynamicRoutingParms entry defines parameters used by OMPROUTE to control the dynamic routes added to the policy-based route table
  - Multiple DynamicRoutingParms can be configured on a RouteTable statement
- Route Table Advanced Parameters
  - Multipath indicates whether or not the Multipath algorithm should be used for this table
  - IgnorePathMtuUpdate indicates whether IPv4 ICMP Fragmentation Needed messages should be ignored for this route table
  - DynamicXCFRoutes indicates whether direct routes to dynamic XCF addresses on other TCP/IP stacks should be added to this route table.

Up to 255 policy-based route tables can be defined for a stack, but only the active tables are installed in the stack. A route table is active if it is referenced by an active routing rule and its associated action.   Like the main route table, a policy-based route table can contain both static and dynamic routes.

The table name specified for a policy-based route table may be from one to eight characters in length. With the addition of the policy-based routing function, the main route table has also been given a name so it can be identified in displays and messages. The name of the main route table is EZBMAIN.

Policy-based route table names can be configured in lower case, upper case, or mixed case. Reports presented on the MVS console (for example, Netstat and OMPROUTE Display) are displayed in all upper case. When a name is provided to filter Netstat or OMPROUTE Display output, the case of the name is ignored. If the same name is used for multiple policy-based route tables, but using a different case for each the names will be indistinguishable in MVS console reports and when reports are filtered by table name, all tables with that name will be included, regardless of case. You may want to define all table names using UPPER case.

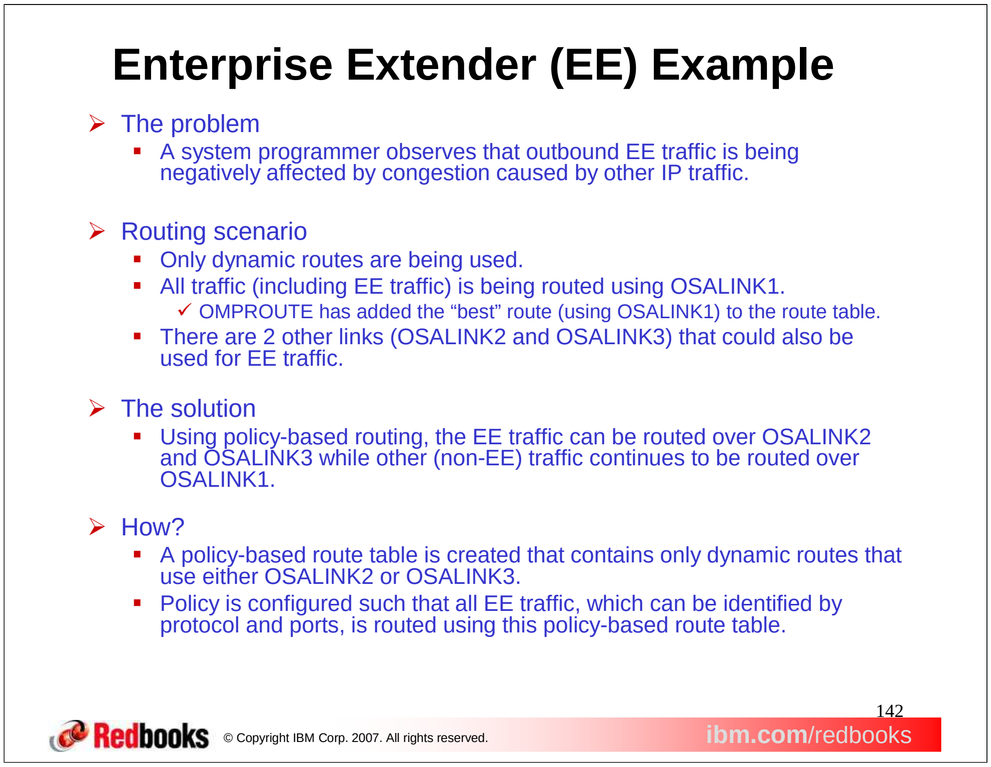
A Route entry on the RouteTable statement is used to define a static route. The syntax of the Route entry is similar to the syntax of the ROUTE entry on the BEGINROUTES statement, used to define static routes for the main route table. The differences between the two are in the way that the route destination is specified and in the way that a route is processed by the stack when the link that the route uses is not defined to the stack.

A DynamicRoutingParms entry on the RouteTable statement is used to define a link and, optionally, a next hop IP address that are to be used by OMPROUTE to control the dynamic routes that will be added to the route table. If the link is not defined in the TCP/IP profile, the DynamicRoutingParms definition is kept, but not used until the link is defined.   Multiple DynamicRoutingParms entries can be specified on a RouteTable statement to allow the route table to use multiple links and next hops.

The Multipath setting on the RouteTable statement allows you to indicate when the multipath algorithm used for a policy-based route table should be different from the algorithm being used for the main route table. The main route table uses the IPCONFIG MULTIPATH / NOMULTIPATH setting from the TCP/IP profile. If a different multipath setting is needed for traffic using a policy-based route table, use the RouteTable Multipath parameter. You can specify UseGlobal which indicates the IPCONFIG multipath setting will be used for this policy-based routing table. You can indicate that either the perpacket or the perconnection multipath algorithm is to be used for the table or you can also indicate that no multipath processing should be used for the table.

The IgnorePathMtuUpdate option allows you to control whether ICMP Fragmentation Needed messages will be applied to the routes in the table. This is an advanced option that should not normally need to be set. When path MTU discovery is enabled for the stack, IPv4 ICMP Fragmentation Needed messages are used to lower the MTU value used to send data to a specific destination. The path MTU is updated for all routes to the destination. By default, all routes to the destination in policy-based route tables are also updated. You may wish to ignore path MTU updates for a policy-based route table containing routes known to use paths that support large MTU values. If there are routes in another route table for the same destinations and those routes may require a smaller path MTU value, IgnorePathMtuUpdate Yes will ensure that a path MTU update that results from sending data on a small MTU route will not cause an update to the path MTU for the routes in the policy-based route table.

The DynamicXCFRoutes option allows you to control whether direct routes to dynamic XCF addresses on other TCP/IP stacks should be added to the route table. This is an advanced option that should not normally need to be set. Consider this option if you have locally originated traffic that will use a policy-based route table whose destination will be the dynamic XCF address of another stack. The routes that will be added to the policy-based route table as a result of this option being set are the same routes that are automatically generated in the main route table when dynamic XCF links are active.

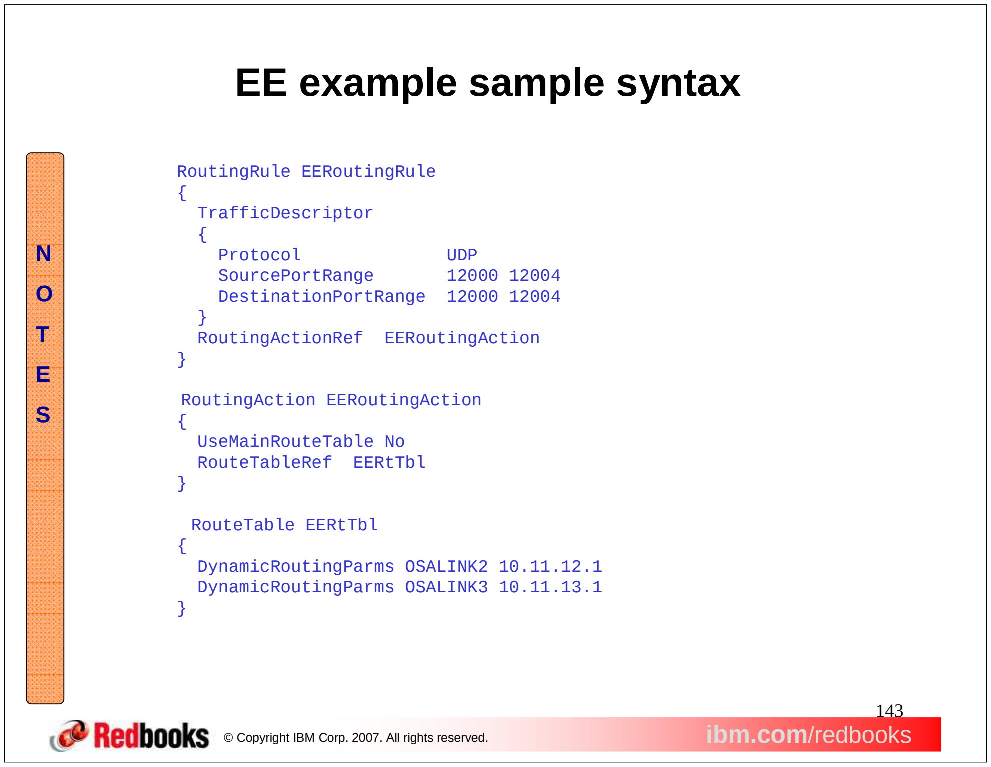


Enterprise Extender (EE) Example

- The problem
  - A system programmer observes that outbound EE traffic is being negatively affected by congestion caused by other IP traffic.
- Routing scenario
  - Only dynamic routes are being used.
  - All traffic (including EE traffic) is being routed using OSALINK1.
    - ✓ OMPROUTE has added the “best” route (using OSALINK1) to the route table.
  - There are 2 other links (OSALINK2 and OSALINK3) that could also be used for EE traffic.
- The solution
  - Using policy-based routing, the EE traffic can be routed over OSALINK2 and OSALINK3 while other (non-EE) traffic continues to be routed over OSALINK1.
- How?
  - A policy-based route table is created that contains only dynamic routes that use either OSALINK2 or OSALINK3.
  - Policy is configured such that all EE traffic, which can be identified by protocol and ports, is routed using this policy-based route table.

As an example of a situation where policy-based routing may be useful, consider the scenario where a system programmer has determined that his outbound Enterprise Extender traffic is being negatively affected by congestion on the link being used by the main route table. His stack is configured to use dynamic routing and the dynamic routes are sending all traffic over the same link. He knows that if he could somehow move only the Enterprise Extender traffic to another available link, he could get that traffic flowing better.

Using policy-based routing he can move the EE traffic. He can create a policy-based route table that will only contain routes that use the other available links. He can then define policy such that all of the EE traffic will use that route table. All of the EE traffic will now be routed using the other available links while all other traffic, which does not match the policy, will continue to be routed using the main route table.

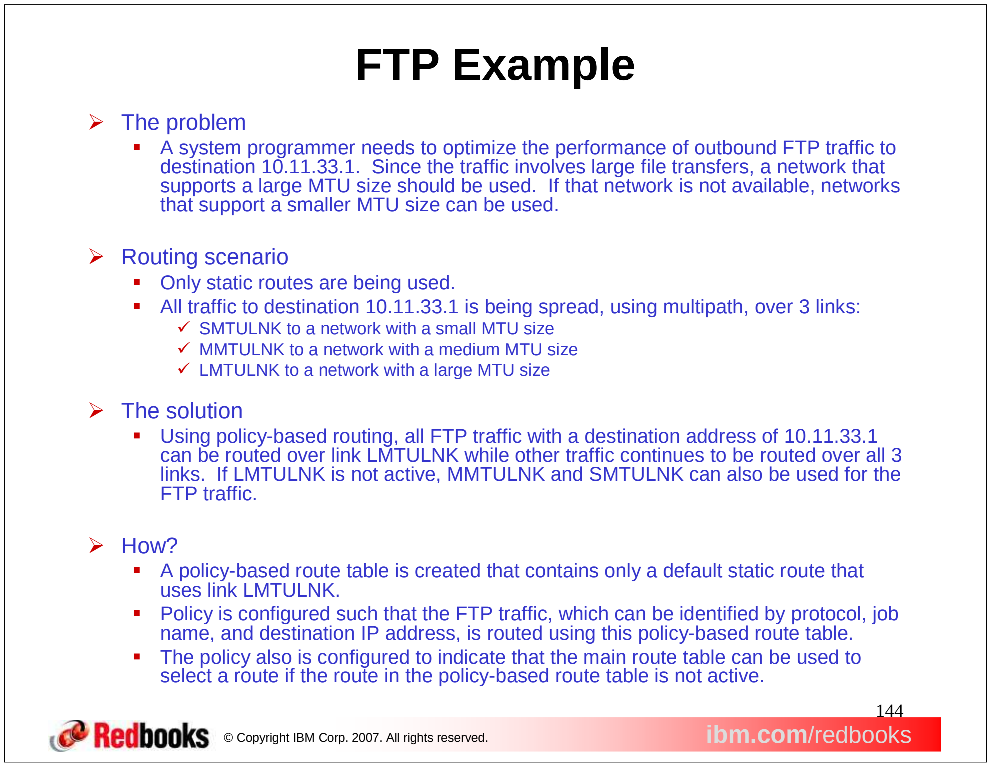


EE example sample syntax

N
O
T
E
S

```
RoutingRule EERoutingRule
{
  TrafficDescriptor
  {
    Protocol              UDP
    SourcePortRange       12000 12004
    DestinationPortRange  12000 12004
  }
  RoutingActionRef  EERoutingAction
}

RoutingAction EERoutingAction
{
  UseMainRouteTable No
  RouteTableRef  EERtTbl
}

 RouteTable EERtTbl
{
  DynamicRoutingParms OSALINK2 10.11.12.1
  DynamicRoutingParms OSALINK3 10.11.13.1
}
```

This notes page shows an example of policy that could be written to solve our example problem, where the system programmer needed to move his EE traffic from a congested link to other available links.

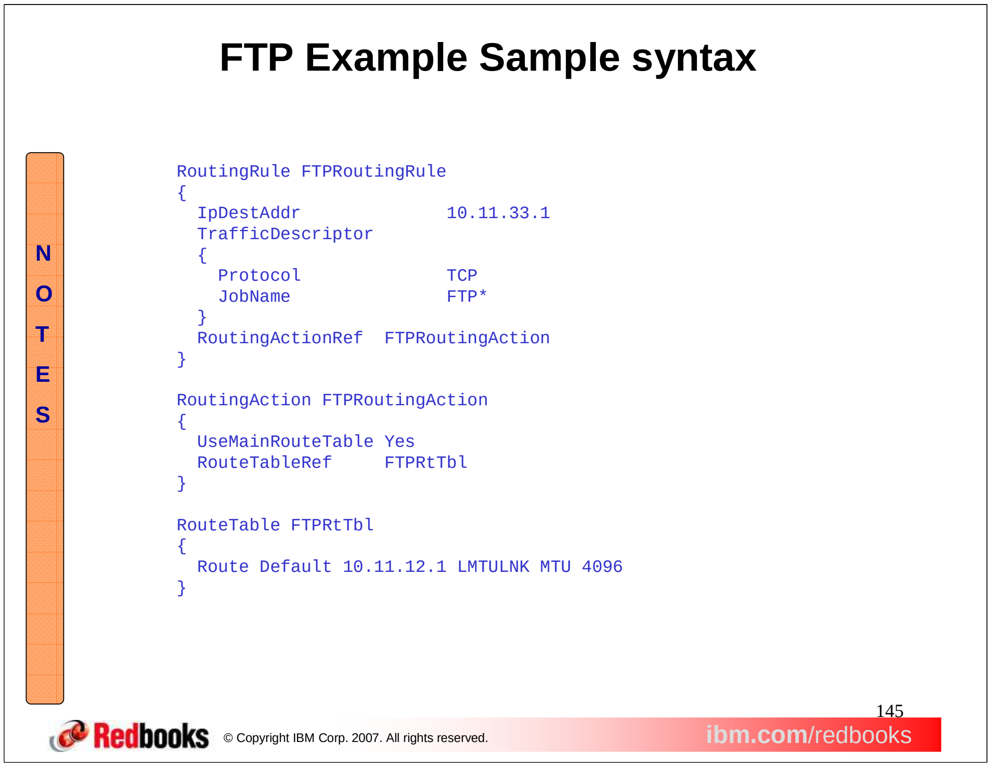


FTP Example

- The problem
  - A system programmer needs to optimize the performance of outbound FTP traffic to destination 10.11.33.1. Since the traffic involves large file transfers, a network that supports a large MTU size should be used. If that network is not available, networks that support a smaller MTU size can be used.
- Routing scenario
  - Only static routes are being used.
  - All traffic to destination 10.11.33.1 is being spread, using multipath, over 3 links:
    - ✓ SMTULNK to a network with a small MTU size
    - ✓ MMTULNK to a network with a medium MTU size
    - ✓ LMTULNK to a network with a large MTU size
- The solution
  - Using policy-based routing, all FTP traffic with a destination address of 10.11.33.1 can be routed over link LMTULNK while other traffic continues to be routed over all 3 links. If LMTULNK is not active, MMTULNK and SMTULNK can also be used for the FTP traffic.
- How?
  - A policy-based route table is created that contains only a default static route that uses link LMTULNK.
  - Policy is configured such that the FTP traffic, which can be identified by protocol, job name, and destination IP address, is routed using this policy-based route table.
  - The policy also is configured to indicate that the main route table can be used to select a route if the route in the policy-based route table is not active.

As another example of a situation where policy-based routing may be useful, consider the scenario where a system programmer needs to optimize the performance of her outbound FTP traffic to a particular destination. Her stack is configured to use static routing and the static routes are configured such that all traffic is being spread across three different links. These links access networks with a variety of MTU sizes. She knows that she could improve the performance of this FTP traffic if she could make all of that traffic go out over the link to the network with the largest MTU, whenever possible.

Using policy-based routing she can move this specific FTP traffic. She can create a policy-based route table that contains only routes that use the link to the network with the largest MTU. She can then define policy such that all of the FTP traffic to the particular destination will use that route table. All of that FTP traffic will now be routed using the large MTU network while all other traffic, which does not match the policy, will continue to be routed using the main route table. Since she wants this FTP traffic to continue to be routed, even if the link to the large MTU network becomes unavailable, she can indicate that the main route table be used as a backup to the policy-based route table.

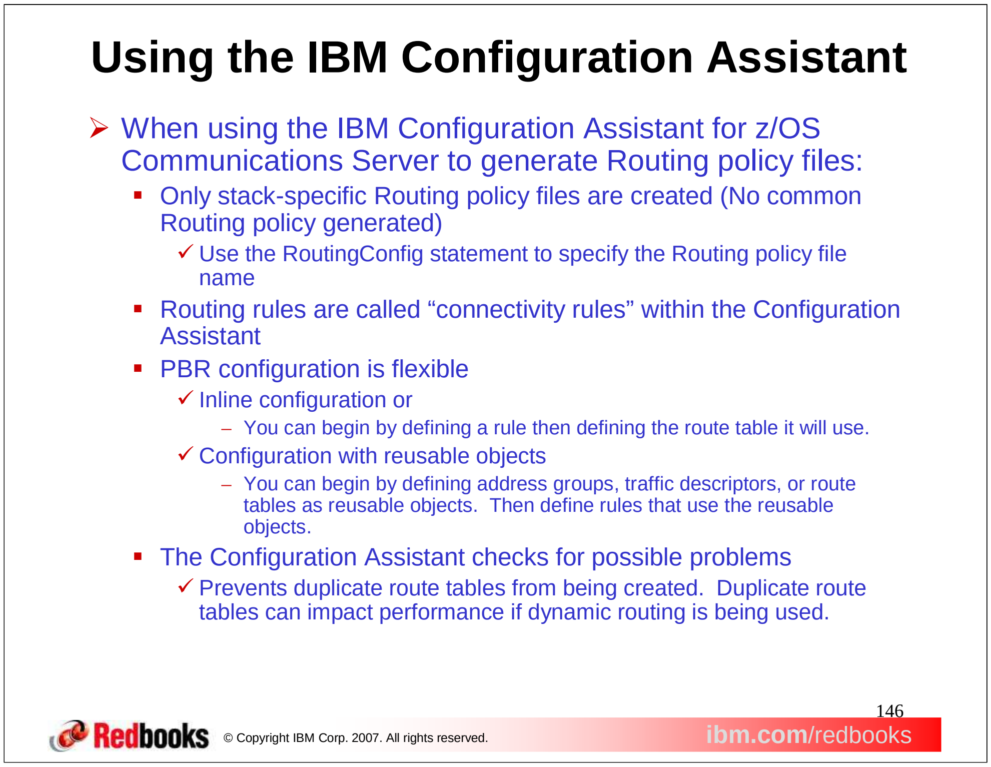


FTP Example Sample syntax

NOTES

```
RoutingRule FTPRoutingRule
{
  IpDestAddr              10.11.33.1
  TrafficDescriptor
  {
    Protocol              TCP
    JobName               FTP*
  }
  RoutingActionRef  FTPRoutingAction
}

RoutingAction FTPRoutingAction
{
  UseMainRouteTable Yes
  RouteTableRef     FTPRtTbl
}

RouteTable FTPRtTbl
{
  Route Default 10.11.12.1 LMTULNK MTU 4096
}
```

This notes page shows an example of policy that could be written to solve our example problem, where the system programmer needed to move her outbound FTP traffic being sent to a particular destination such that it would use a link to a network with a large MTU. This example assumes that all outbound FTP traffic sent to destination 10.11.33.1 is sent by jobs with names beginning with “FTP”.



Using the IBM Configuration Assistant

- When using the IBM Configuration Assistant for z/OS Communications Server to generate Routing policy files:
  - Only stack-specific Routing policy files are created (No common Routing policy generated)
    - ✓ Use the RoutingConfig statement to specify the Routing policy file name
  - Routing rules are called “connectivity rules” within the Configuration Assistant
  - PBR configuration is flexible
    - ✓ Inline configuration or
      - You can begin by defining a rule then defining the route table it will use.
    - ✓ Configuration with reusable objects
      - You can begin by defining address groups, traffic descriptors, or route tables as reusable objects. Then define rules that use the reusable objects.
  - The Configuration Assistant checks for possible problems
    - ✓ Prevents duplicate route tables from being created. Duplicate route tables can impact performance if dynamic routing is being used.

As mentioned previously, the IBM Configuration Assistant for z/OS Communications Server can be used as an alternative to manually creating your policy agent configuration flat-file for policy-based routing. When this method is used, there are no common configuration files created. All configuration is placed in a stack-specific configuration file. Use the RoutingConfig statement in your image configuration file to specify the name and location of the file.

Within the Configuration Assistant, routing rules are called connectivity rules. This is the name that the Configuration Assistant uses for rules across the different types of policy.

Using the Configuration Assistant will simplify the job of configuring policy-based routing in a few ways.

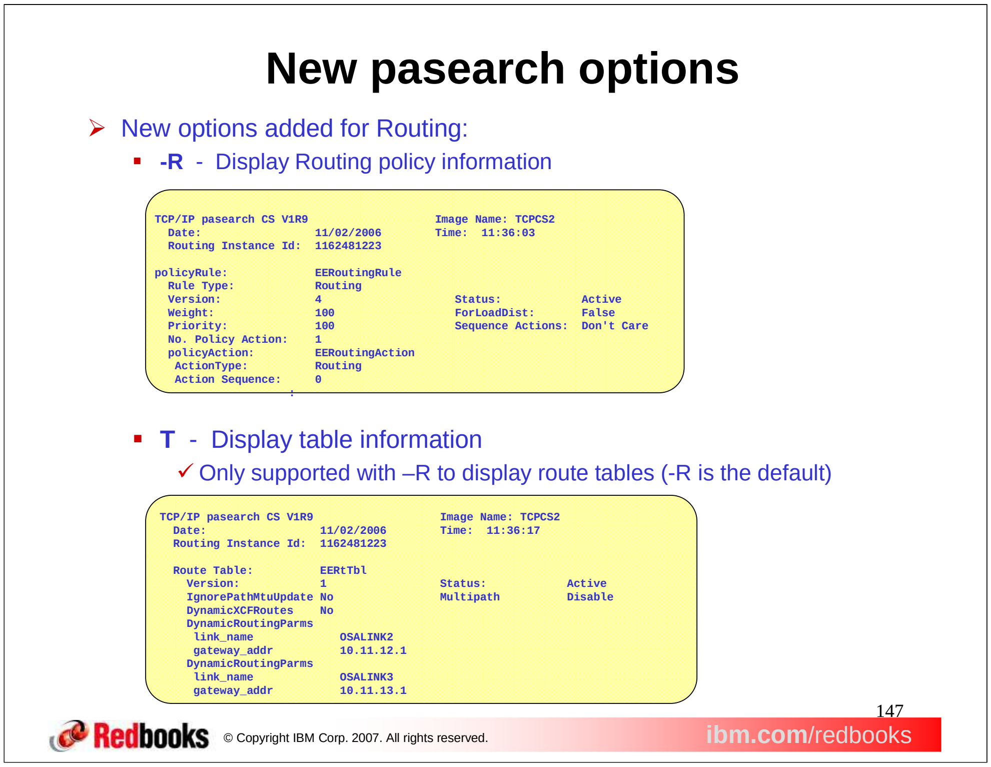
First, there is no need to configure the policy rule and action as two separate objects. In the Configuration Assistant, these are configured as a single object that identifies both the characteristics of the traffic and the route tables that are to be used for that traffic.

Next, the rule priority does not need to be manually configured. Instead, the priority is managed by the Configuration Assistant based on the order that the rules are displayed on the rules panel.

Lastly, the Configuration Assistant will verify the information you enter and will generate a flat-file containing policy statements that are free of syntax errors.

The Configuration Assistant will allow you flexibility in how you configure your Routing policy. You can configure everything inline for each rule and associated route table. Alternatively, you can create address groups, traffic descriptors, and route tables as reusable objects and then define rules that use these reusable objects.

If you use the Configuration Assistant to create your configuration files, it will ensure that you do not create duplicate route tables. Duplicate route tables should be avoided as they increase complexity and, in the case of route tables with dynamic routing support, they impact performance.



New pasearch options

- New options added for Routing:
  - **-R** - Display Routing policy information

```
TCP/IP pasearch CS V1R9                 Image Name: TCPCS2
  Date:                   11/02/2006    Time:  11:36:03
  Routing Instance Id:    1162481223

policyRule:               EERoutingRule
  Rule Type:              Routing
  Version:                4                Status:            Active
  Weight:                 100              ForLoadDist:       False
  Priority:               100              Sequence Actions:  Don't Care
  No. Policy Action:      1
  policyAction:           EERoutingAction
   ActionType:            Routing
   Action Sequence:       0
                     :
```

  - **T** - Display table information
    - ✓ Only supported with –R to display route tables (-R is the default)

```
TCP/IP pasearch CS V1R9                 Image Name: TCPCS2
  Date:                   11/02/2006    Time:  11:36:17
  Routing Instance Id:    1162481223

  Route Table:            EERtTbl
    Version:              1             Status:            Active
    IgnorePathMtuUpdate No              Multipath          Disable
    DynamicXCFRoutes    No
    DynamicRoutingParms
     link_name               OSALINK2
     gateway_addr            10.11.12.1
    DynamicRoutingParms
     link_name               OSALINK3
     gateway_addr            10.11.13.1
```

In order to allow for querying of Routing policies, the set of options available for use with the pasearch command is expanded to include the -R and -T options. Use the -R option to indicate that the pasearch command is requesting Routing policy information. **Pasearch -R** will display active Routing rules and associated Routing actions. The active rule name displayed in this example is EERoutingRule. This active rule references the Routing action named EERoutingAction. The priority for this rule is 100. This active rule is installed in stack TCPCS2.

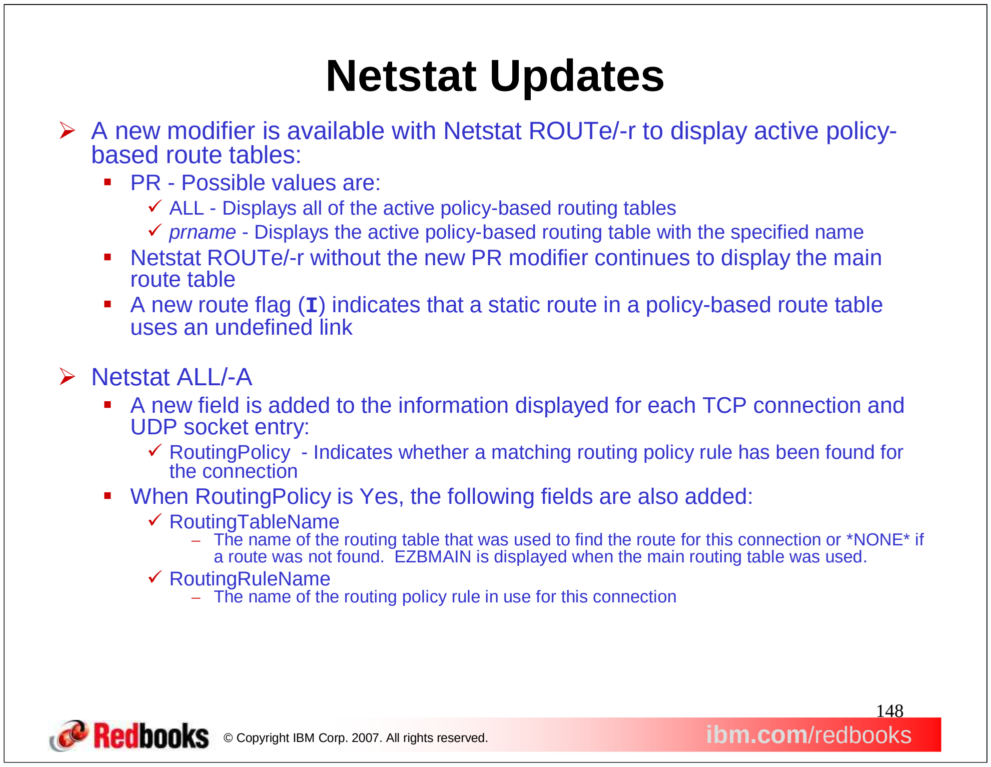
Use the -T option to indicate that the pasearch command is requesting table information. Currently, the -T option is only used to display policy-based route table information. This slide shows the display of an active Route table named EERtTbl. It is the result of a **pasearch -T -f EERtTbl** command**.** By using the pasearch -f option you can display a single Route table. This display indicates that two DynamicRoutingParms are defined for the table. Path MTU Update messages will be processed and the multipath routing algorithm is disabled for this routing table. In addition, direct routes to dynamic XCF addresses on other TCP/IP stacks will not be added to the route table.

These 2 new options can be used in combination with a variety of other pasearch options to control the information that will be displayed in response to the pasearch command. Following are some of the options that can be used in combination with the -R and -T options:

- Use -R with -e to display Routing policy rules and actions - this is the default.
- Use -R with -r to display Routing policy rules.
- Use -R with -a to display Routing policy actions.
- Use -T with -R to display Routing route tables - this is the default.

Use -A with any of the combinations above to display active policy information - this is the default. Use -**I** with any of the combinations above to display inactive policy information. Use -f with any of the combinations above to filter the information displayed by policy name.

Note that a routing rule is active based on the IpTimeCondition. A routing action is always active. A route table is active if it is referenced by an active routing rule and its associated action.



Netstat Updates

- A new modifier is available with Netstat ROUTe/-r to display active policy-based route tables:
  - PR - Possible values are:
    - ✓ ALL - Displays all of the active policy-based routing tables
    - ✓ *prname* - Displays the active policy-based routing table with the specified name
  - Netstat ROUTe/-r without the new PR modifier continues to display the main route table
  - A new route flag (**`I`**) indicates that a static route in a policy-based route table uses an undefined link
- Netstat ALL/-A
  - A new field is added to the information displayed for each TCP connection and UDP socket entry:
    - ✓ RoutingPolicy - Indicates whether a matching routing policy rule has been found for the connection
  - When RoutingPolicy is Yes, the following fields are also added:
    - ✓ RoutingTableName
      - The name of the routing table that was used to find the route for this connection or *NONE* if a route was not found. EZBMAIN is displayed when the main routing table was used.
    - ✓ RoutingRuleName
      - The name of the routing policy rule in use for this connection

148

Redbooks

ibm.com/redbooks

A new modifier has been added to the Netstat ROUTe/-r command. The modifier is PR and it is used to indicate that active policy-based route tables are to be displayed. Since only active route tables are installed in the TCP/IP stack by policy agent, only active tables can be displayed by Netstat. The values that can be specified on the PR modifier are ALL or the name of a policy-based route table. Use ALL to request the display of all active policy-based route tables. Use the name of a policy-based route table to display only that active table.

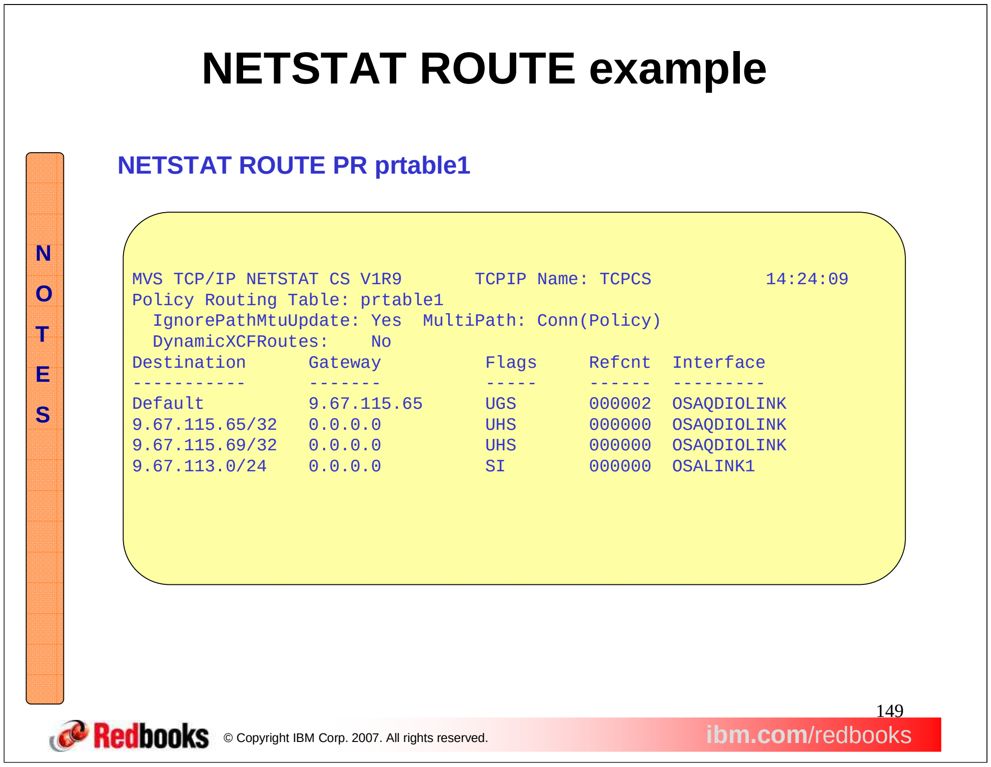
The IQDIO modifier displays the HiperSockets Accelerator routing table, which is separate from the main route table and any policy-based route tables. Therefore, an error message will be issued if the PR modifier is used in combination with the IQDIO modifier.

Policy-based routing does not apply to IPv6 route tables. Therefore, no information will be displayed if the PR modifier is used in combination with the ADDRTYPE=IPV6 modifier.

A new flag has been added to the set of flags that can be displayed for each route included in the report. The I flag indicates a static route that is configured to use a link that is not defined to the stack.

The report generated by the Netstat ALL/-A command has been modified to include policy-based routing information for each TCP connection and UDP socket. The field RoutingPolicy indicates whether a matching routing policy rule has been found for the connection or socket entry. If so, the fields RoutingTableName and RoutingRuleName provide the names of the routing table and routing policy rule being used.

For an Enterprise Extender (EE) UDP socket entry, the RoutingPolicy value is always No. The routing policy information for an EE UDP socket entry is displayed using the DISPLAY NET,EEDIAG,TEST=YES command. For details on using this command, refer to z/OS Communications Server: SNA Operation.



NETSTAT ROUTE example

NOTES

NETSTAT ROUTE PR prtable1

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: Yes  MultiPath: Conn(Policy)
  DynamicXCFRoutes:    No
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.113.0/24    0.0.0.0          SI        000000  OSALINK1
```

149

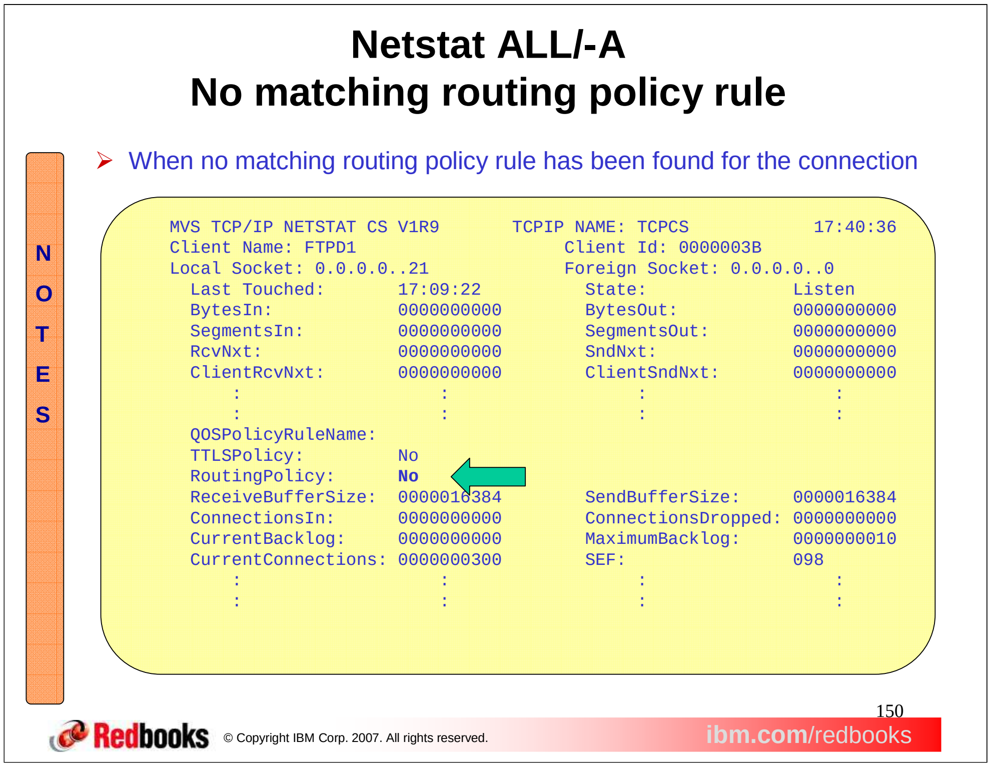
Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

This example uses the PR modifier with the name of a policy-based route table. The resulting report includes the name of the route table, the three table-specific configuration values (IgnorePathMtuUpdate, MultiPath, and DynamicXCFRoutes), and the routes contained in the route table.

The MultiPath value shown in this example indicates that the perconnection multipath algorithm is to be used for the table. The value in the parentheses, Policy, indicates that this setting was configured on the policy RouteTable statement that defined this table. When the multipath setting for a policy-based route table is being inherited from the IPCONFIG MULTIPATH setting, the value in parentheses is Profile.

The last line of the report shows a static route that is defined using a link that is not currently defined to the stack. The I flag is used to indicate this.

When PR ALL is specified, similar information is repeated for all of the policy-based route tables.



Netstat ALL/-A
No matching routing policy rule

- When no matching routing policy rule has been found for the connection

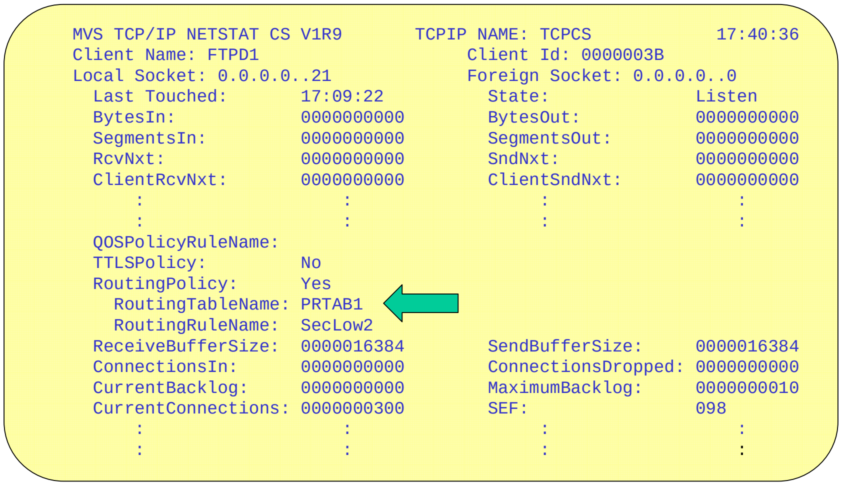
```
MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS            17:40:36
Client Name: FTPD1                      Client Id: 0000003B
Local Socket: 0.0.0.0..21               Foreign Socket: 0.0.0.0..0
  Last Touched:        17:09:22           State:             Listen
  BytesIn:             0000000000         BytesOut:          0000000000
  SegmentsIn:          0000000000         SegmentsOut:       0000000000
  RcvNxt:              0000000000         SndNxt:            0000000000
  ClientRcvNxt:        0000000000         ClientSndNxt:      0000000000
      :                    :                  :                  :
      :                    :                  :                  :
  QOSPolicyRuleName:
  TTLSPolicy:          No
  RoutingPolicy:       No
  ReceiveBufferSize:   0000016384         SendBufferSize:    0000016384
  ConnectionsIn:       0000000000         ConnectionsDropped: 0000000000
  CurrentBacklog:      0000000000         MaximumBacklog:    0000000010
  CurrentConnections:  0000000300         SEF:               098
      :                    :                  :                  :
      :                    :                  :                  :
```

This example shows a connection for which a matching routing policy rule has not been found. Note that the RoutingTableName and RoutingRuleName fields are not displayed.

Netstat ALL/-A
A matching routing policy rule exist

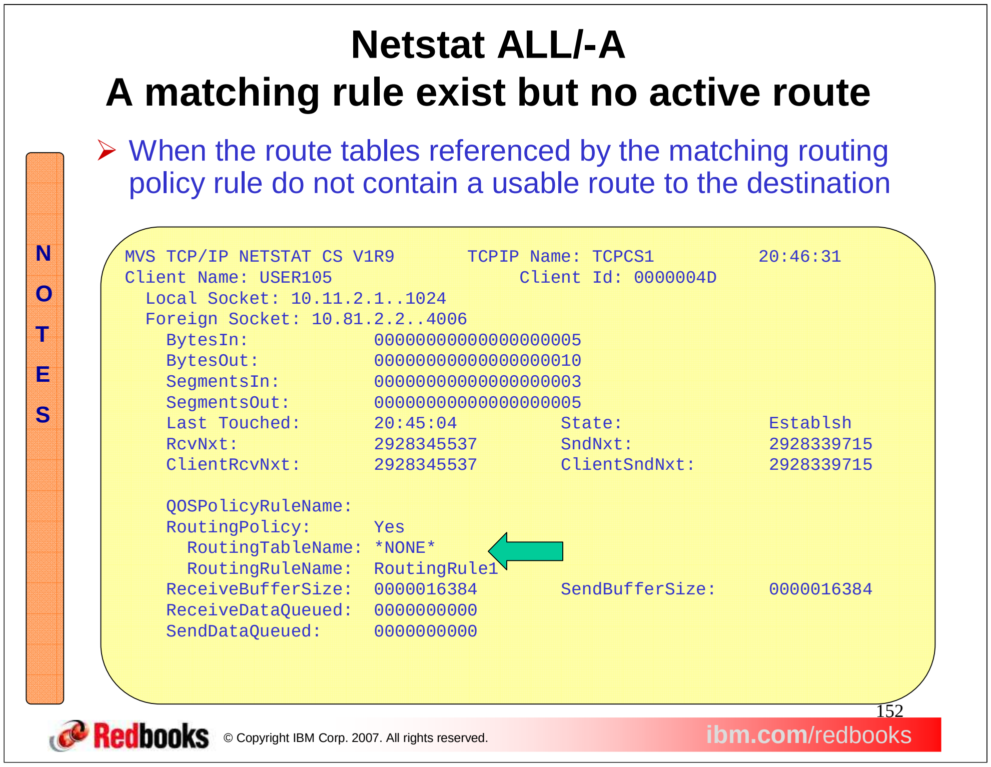
N O T E S

- When a matching routing policy rule has been found for the connection



```
MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS            17:40:36
Client Name: FTPD1                    Client Id: 0000003B
Local Socket: 0.0.0.0..21             Foreign Socket: 0.0.0.0..0
  Last Touched:       17:09:22          State:              Listen
  BytesIn:            0000000000        BytesOut:           0000000000
  SegmentsIn:         0000000000        SegmentsOut:        0000000000
  RcvNxt:             0000000000        SndNxt:             0000000000
  ClientRcvNxt:       0000000000        ClientSndNxt:       0000000000
      :                   :                 :                   :
      :                   :                 :                   :
  QOSPolicyRuleName:
  TTLSPolicy:         No
  RoutingPolicy:      Yes
    RoutingTableName: PRTAB1
    RoutingRuleName:  SecLow2
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ConnectionsIn:      0000000000        ConnectionsDropped: 0000000000
  CurrentBacklog:     0000000000        MaximumBacklog:     0000000010
  CurrentConnections: 0000000300        SEF:                098
      :                   :                 :                   :
      :                   :                 :                   :
```

This example shows a connection for which a matching routing policy rule has been found. The RoutingTableName and RoutingRuleName fields are now displayed, showing the route table and routing policy rule being used.

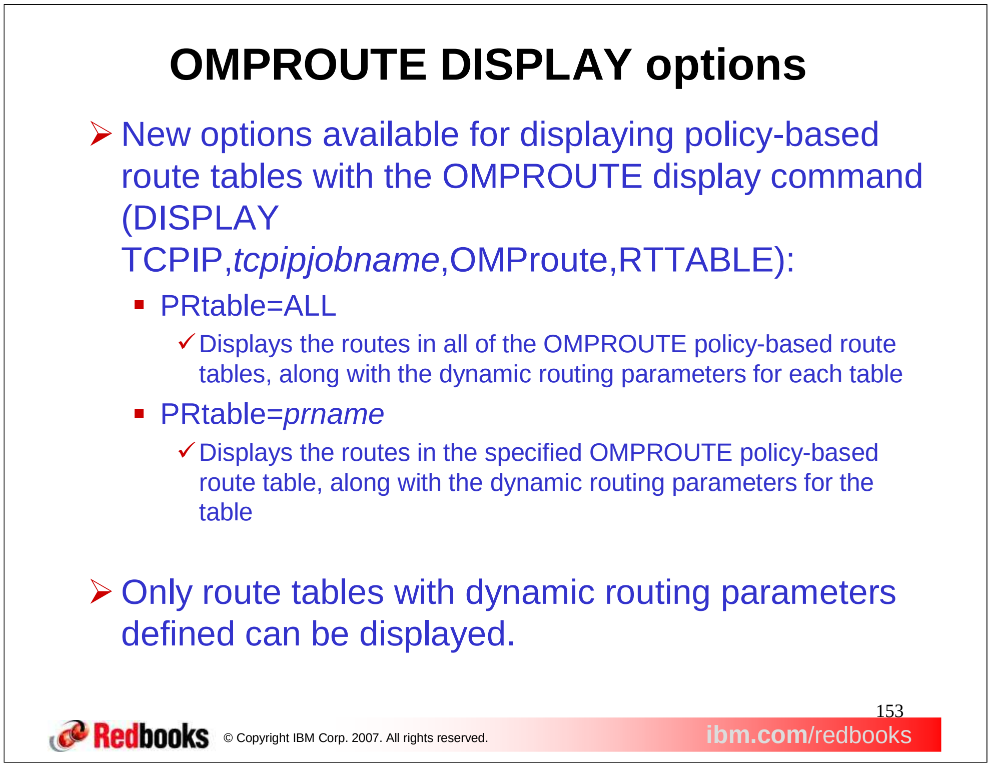


Netstat ALL/-A
A matching rule exist but no active route

- When the route tables referenced by the matching routing policy rule do not contain a usable route to the destination

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS1          20:46:31
Client Name: USER105                  Client Id: 0000004D
  Local Socket: 10.11.2.1..1024
  Foreign Socket: 10.81.2.2..4006
    BytesIn:            00000000000000000005
    BytesOut:           00000000000000000010
    SegmentsIn:         00000000000000000003
    SegmentsOut:        00000000000000000005
    Last Touched:       20:45:04          State:             Establsh
    RcvNxt:             2928345537        SndNxt:            2928339715
    ClientRcvNxt:       2928345537        ClientSndNxt:      2928339715

    QOSPolicyRuleName:
    RoutingPolicy:      Yes
      RoutingTableName: *NONE*
      RoutingRuleName:  RoutingRule1
    ReceiveBufferSize:  0000016384        SendBufferSize:    0000016384
    ReceiveDataQueued:  0000000000
    SendDataQueued:     0000000000
```

This example shows a connection for which a matching routing policy rule has been found. However, a search of all of the associated route tables has failed to find a usable route to the destination. When this is the case, the RoutingTableName is displayed as *NONE*.



OMPROUTE DISPLAY options

- New options available for displaying policy-based route tables with the OMPROUTE display command (DISPLAY TCPIP,*tcpipjobname*,OMProute,RTTABLE):
  - PRtable=ALL
    - ✓ Displays the routes in all of the OMPROUTE policy-based route tables, along with the dynamic routing parameters for each table
  - PRtable=*prname*
    - ✓ Displays the routes in the specified OMPROUTE policy-based route table, along with the dynamic routing parameters for the table
- Only route tables with dynamic routing parameters defined can be displayed.

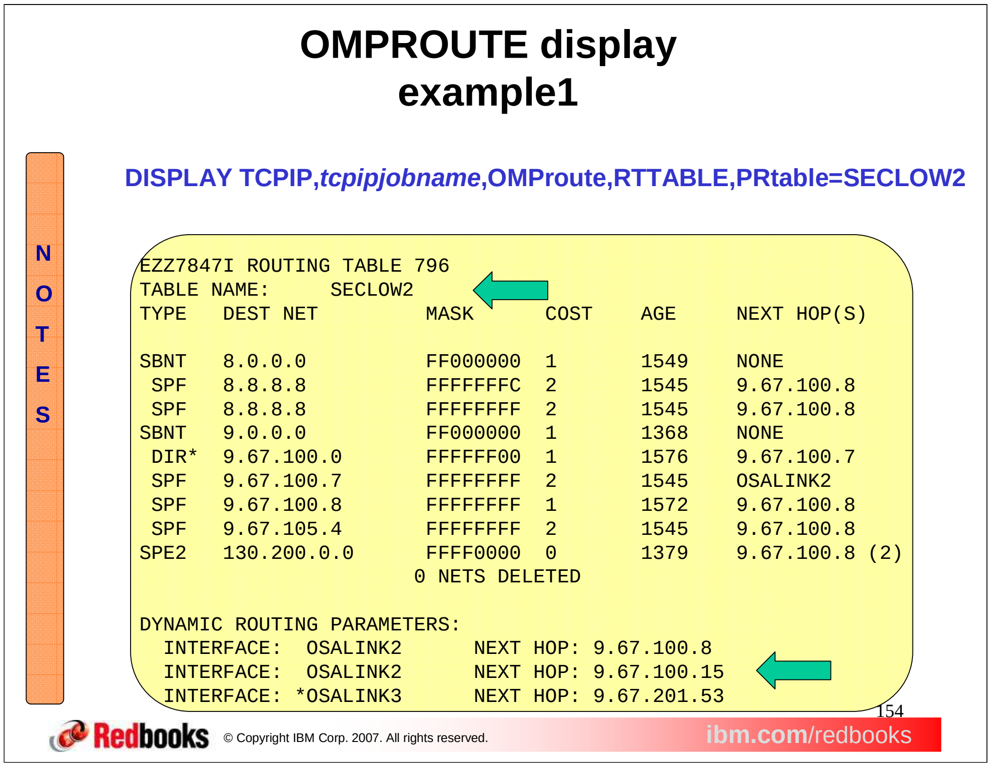
153

Redbooks ibm.com/redbooks

A new option has been added to the OMPROUTE DISPLAY RTTABLE command. The option is PRtable and it is used to indicate that OMPROUTE policy-based route tables are to be displayed. The values that can be specified on the PRtable option are ALL or the name of a policy-based route table. Use ALL to request the display of all OMPROUTE policy-based route tables. Use the name of a policy-based route table to display only that table.

The PRtable option can be used in combination with the DEST= option to display details of the routes in policy-based route tables to a particular destination.

OMPROUTE has no knowledge of policy-based route tables that are defined without dynamic routing parameters. Those route tables are using static routing only. Since OMPROUTE has no knowledge of those tables, they cannot be displayed with the OMPROUTE DISPLAY command.



OMPROUTE display example1

DISPLAY TCPIP,*tcpipjobname*,OMProute,RTTABLE,PRtable=SECLOW2

```
EZZ7847I ROUTING TABLE 796
TABLE NAME:     SECLOW2
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    OSALINK2
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8 (2)
                       0 NETS DELETED

DYNAMIC ROUTING PARAMETERS:
  INTERFACE:  OSALINK2      NEXT HOP: 9.67.100.8
  INTERFACE:  OSALINK2      NEXT HOP: 9.67.100.15
  INTERFACE: *OSALINK3      NEXT HOP: 9.67.201.53
```

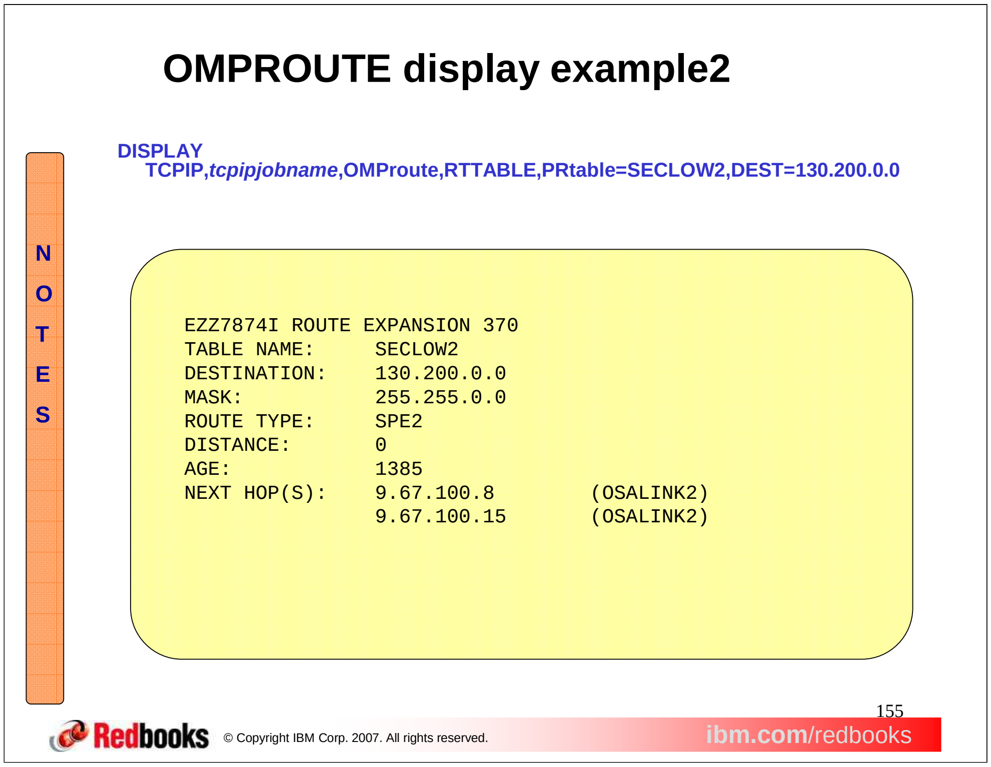
In this example, the PRtable=*prname* option has been used to display a particular policy-based route table. The *prname* value of SECLOW2 results in only that route table being displayed.

Table SECLOW2 is defined with three dynamic routing parameters that each specify a link and next hop. All dynamic routes added to this table should be either direct routes over one of these links or indirect routes over one of the links that have the associated IP address as next hop.

Most of the information in the display of a policy-based route table is the same as what is included in the display of the main route table. What is added for policy-based route tables is the name of the table at the top and the dynamic routing parameters being used for the table at the bottom.

The asterisk beside the link name in the last dynamic routing parameter shown in this example indicates that OSALINK3 is either not currently defined to the TCP/IP stack or not currently active. In either case, there would be no dynamic routes in the route table over that link.

When PRtable=ALL is specified, similar information is repeated for all of the OMPROUTE policy-based route tables.



OMPROUTE display example2

DISPLAY
TCPIP,*tcpipjobname*,OMProute,RTTABLE,PRtable=SECLOW2,DEST=130.200.0.0

NOTES

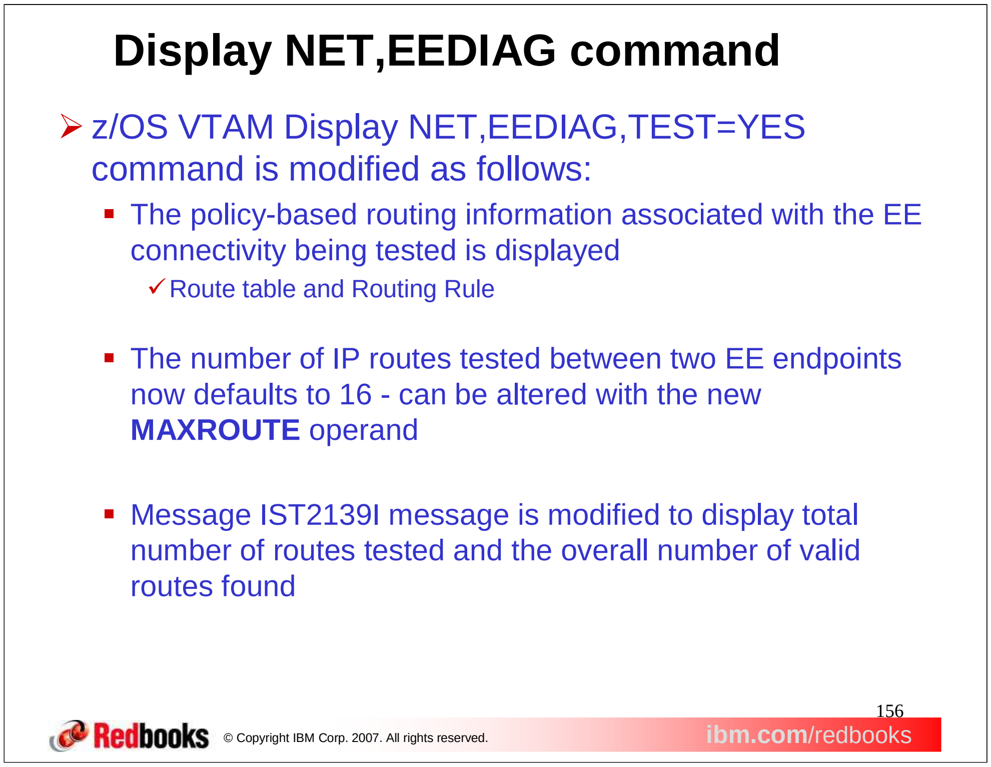
```
EZZ7874I ROUTE EXPANSION 370
TABLE NAME:      SECLOW2
DESTINATION:     130.200.0.0
MASK:            255.255.0.0
ROUTE TYPE:      SPE2
DISTANCE:        0
AGE:             1385
NEXT HOP(S):     9.67.100.8         (OSALINK2)
                 9.67.100.15        (OSALINK2)
```

155

Redbooks ibm.com/redbooks

In this example, the DEST=*ip_addr* option has been used in order to display the multiple next hops to the 130.200.0.0 network that appear in policy-based route table SECLOW2.

If PRtable=ALL was specified instead of PRtable=SECLOW2 and other policy-based route tables also contained routes to 130.200.0.0, the information for the routes in each route table would be included in the report.



Display NET,EEDIAG command

- z/OS VTAM Display NET,EEDIAG,TEST=YES command is modified as follows:
  - The policy-based routing information associated with the EE connectivity being tested is displayed
    - ✓ Route table and Routing Rule
  - The number of IP routes tested between two EE endpoints now defaults to 16 - can be altered with the new **MAXROUTE** operand
  - Message IST2139I message is modified to display total number of routes tested and the overall number of valid routes found

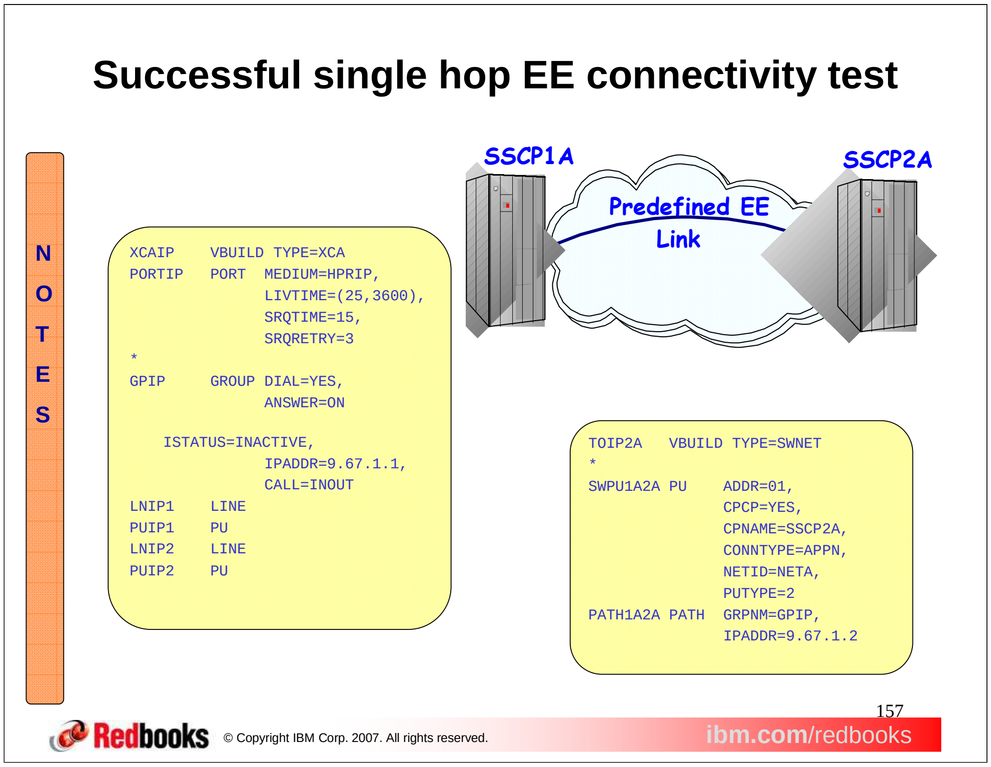
Redbooks

The Display EEDIAG,TEST=YES command, or Enterprise Extender (EE) connectivity test command, is useful in debugging various network problems and was introduced in z/OS Communication Server V1 R8. This command can be used to test an existing Enterprise Extender connection, or it can be used to assist in diagnosing why an EE connection cannot be established.

With policy-based routing, each of the EE ports can be associated with a unique routing rule. The EE traffic utilizing each port could be routed using different route tables. Due to these changes, the command required modifications for the support of policy-based routing.

The MAXROUTE value specifies the maximum number of valid TCP/IP routes that will be tested between two Enterprise Extender (EE) endpoints. Multiple routes may exist when MULTIPATH support or policy-based routing is being utilized in the route calculations between the EE endpoints. When the maximum routes to be tested is exceeded then all routes over the limit will not be tested.

Normally 16 routes (the default) should be sufficient to fully test connectivity between two EE endpoints. If message IST2139I indicates that all routes are not being tested, MAXROUTE can be used to increase the number of routes to be tested.



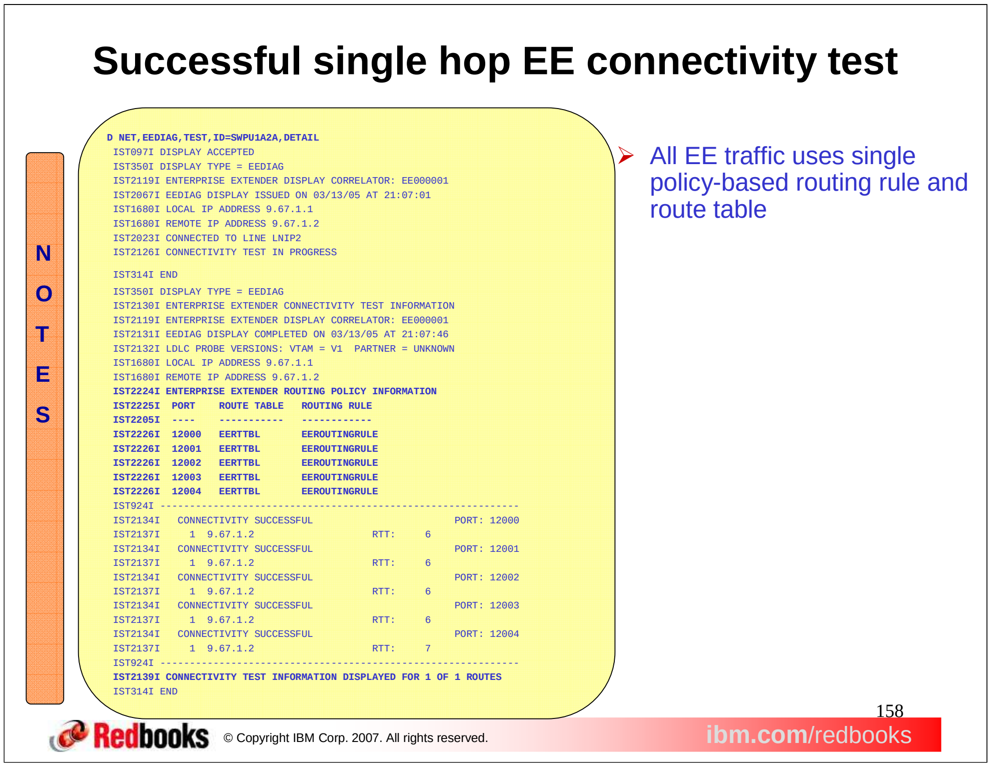
Successful single hop EE connectivity test

N
O
T
E
S
SSCP1A
Predefined EE
Link
SSCP2A

```
XCAIP    VBUILD TYPE=XCA
PORTIP   PORT   MEDIUM=HPRIP,
                LIVTIME=(25,3600),
                SRQTIME=15,
                SRQRETRY=3
*
GPIP     GROUP  DIAL=YES,
                ANSWER=ON

    ISTATUS=INACTIVE,
                IPADDR=9.67.1.1,
                CALL=INOUT
LNIP1    LINE
PUIP1    PU
LNIP2    LINE
PUIP2    PU
```

```
TOIP2A   VBUILD TYPE=SWNET
*
SWPU1A2A PU     ADDR=01,
                CPCP=YES,
                CPNAME=SSCP2A,
                CONNTYPE=APPN,
                NETID=NETA,
                PUTYPE=2
PATH1A2A PATH   GRPNM=GPIP,
                IPADDR=9.67.1.2
```

This is a sample of a single hop predefined Enterprise Extender (EE) link. This example illustrates a case where connectivity being tested between two EE endpoints is successful.



Successful single hop EE connectivity test

```
D NET,EEDIAG,TEST,ID=SWPU1A2A,DETAIL
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = EEDIAG
 IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
 IST2067I EEDIAG DISPLAY ISSUED ON 03/13/05 AT 21:07:01
 IST1680I LOCAL IP ADDRESS 9.67.1.1
 IST1680I REMOTE IP ADDRESS 9.67.1.2
 IST2023I CONNECTED TO LINE LNIP2
 IST2126I CONNECTIVITY TEST IN PROGRESS

 IST314I END
 IST350I DISPLAY TYPE = EEDIAG
 IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
 IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
 IST2131I EEDIAG DISPLAY COMPLETED ON 03/13/05 AT 21:07:46
 IST2132I LDLC PROBE VERSIONS: VTAM = V1  PARTNER = UNKNOWN
 IST1680I LOCAL IP ADDRESS 9.67.1.1
 IST1680I REMOTE IP ADDRESS 9.67.1.2
 IST2224I ENTERPRISE EXTENDER ROUTING POLICY INFORMATION
 IST2225I  PORT    ROUTE TABLE   ROUTING RULE
 IST2205I  ----    -----------   ------------
 IST2226I  12000   EERTTBL       EEROUTINGRULE
 IST2226I  12001   EERTTBL       EEROUTINGRULE
 IST2226I  12002   EERTTBL       EEROUTINGRULE
 IST2226I  12003   EERTTBL       EEROUTINGRULE
 IST2226I  12004   EERTTBL       EEROUTINGRULE
 IST924I -------------------------------------------------------------
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000
 IST2137I     1  9.67.1.2                    RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12001
 IST2137I     1  9.67.1.2                    RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12002
 IST2137I     1  9.67.1.2                    RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12003
 IST2137I     1  9.67.1.2                    RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12004
 IST2137I     1  9.67.1.2                    RTT:     7
 IST924I -------------------------------------------------------------
 IST2139I CONNECTIVITY TEST INFORMATION DISPLAYED FOR 1 OF 1 ROUTES
 IST314I END
```

- All EE traffic uses single policy-based routing rule and route table

In this example, the operator on SSCP1A performs the EE connectivity test to assist in determining the connectivity to the remote EE endpoint located on SSCP2A.

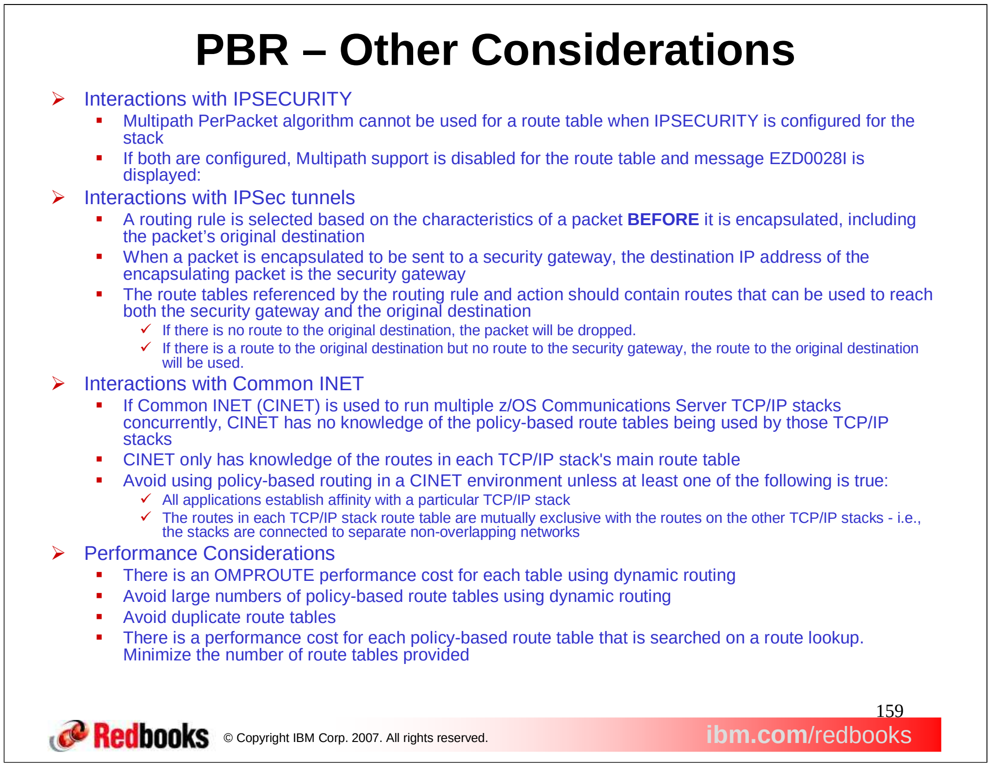
A single policy-based routing rule (EEROUTINGRULE) is being utilized for all EE traffic between the EE endpoints being tested. The policy-based routing rule has indicated that Multipath is disabled and that there is a RouteTable defined for EE traffic (the route table name is EERTTBL).

When a policy-based routing rule is defined for any EE traffic between the EE endpoints, you will receive additional messages. Message IST2224I is a header message for the routing policy information.

Message IST2225I is a header for the display of the EE UDP ports, route tables, and policy routing rules.

Message IST2226I displays the EE UDP ports and each port’s associated route table and policy routing rule. If a policy-based routing rule is not defined for an EE UDP port, the policy routing rule will be displayed as NONE. When the main route table is being utilized (either no policy routing rule or the routing action indicates the use of the main routing table), the route table that is displayed is EZBMAIN.

Message IST2139I has been modified to indicate the number of routes tested and the overall number of routes found for the EE connectivity test. If this message indicates that all routes are not being tested then the D NET,EEDIAG,TEST command can be re-issued with a MAXROUTE operand value sufficiently large enough to test all routes.



PBR – Other Considerations

- Interactions with IPSECURITY
  - Multipath PerPacket algorithm cannot be used for a route table when IPSECURITY is configured for the stack
  - If both are configured, Multipath support is disabled for the route table and message EZD0028I is displayed:
- Interactions with IPSec tunnels
  - A routing rule is selected based on the characteristics of a packet **BEFORE** it is encapsulated, including the packet's original destination
  - When a packet is encapsulated to be sent to a security gateway, the destination IP address of the encapsulating packet is the security gateway
  - The route tables referenced by the routing rule and action should contain routes that can be used to reach both the security gateway and the original destination
    - ✓ If there is no route to the original destination, the packet will be dropped.
    - ✓ If there is a route to the original destination but no route to the security gateway, the route to the original destination will be used.
- Interactions with Common INET
  - If Common INET (CINET) is used to run multiple z/OS Communications Server TCP/IP stacks concurrently, CINET has no knowledge of the policy-based route tables being used by those TCP/IP stacks
  - CINET only has knowledge of the routes in each TCP/IP stack's main route table
  - Avoid using policy-based routing in a CINET environment unless at least one of the following is true:
    - ✓ All applications establish affinity with a particular TCP/IP stack
    - ✓ The routes in each TCP/IP stack route table are mutually exclusive with the routes on the other TCP/IP stacks - i.e., the stacks are connected to separate non-overlapping networks
- Performance Considerations
  - There is an OMPROUTE performance cost for each table using dynamic routing
  - Avoid large numbers of policy-based route tables using dynamic routing
  - Avoid duplicate route tables
  - There is a performance cost for each policy-based route table that is searched on a route lookup. Minimize the number of route tables provided

159

ibm.com/redbooks

As with the main route table, the Multipath PerPacket algorithm cannot be used with a policy-based route table when IPSECURITY is configured for the stack. If these two are configured together, the Multipath function will be disabled.

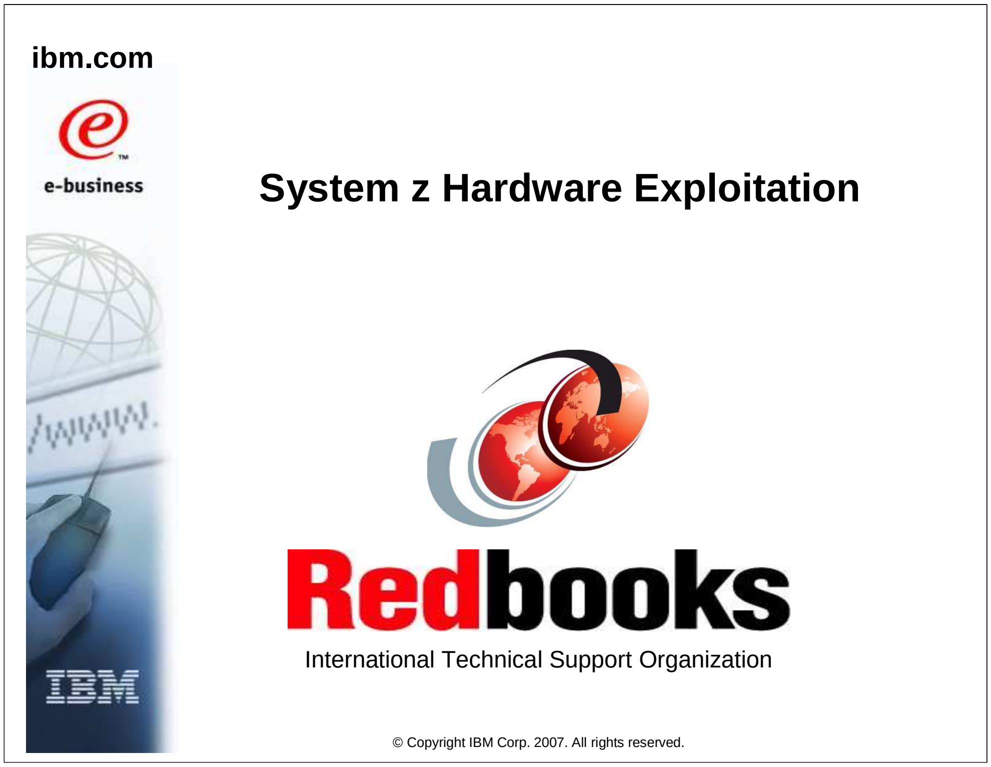
If IPSECURITY is in use on a stack, use either no Multipath algorithm or the Multipath PerConnection algorithm to distribute traffic routed using your policy-based route tables. If Multipath PerPacket is configured in the policy-based route table when IPSECURITY is configured for the stack then the following message is displayed:

EZD0028I IPV4 MULTIPATH PERPACKET NOT VALID WITH IPV4 SECURITY – MULTIPATH SUPPORT DISABLED FOR ROUTE TABLE *table*

When policy-based routing is used on a stack that is also using IP Security, special care must be taken to ensure that policy-based route tables that will be used for IP Security traffic contain the necessary routes. The route tables to be searched will be selected based on the characteristics of the unencapsulated packet. The destination of the encapsulated packet may differ from the destination of the original packet. Routes to both destinations should appear in the route tables to be searched.

The use of policy-based routing on a stack that is in a Common INET environment should be avoided except in very limited scenarios. Those scenarios are described on this page. The problem with this combination in other scenarios is due to the fact that Common INET has no knowledge of the policy-based route tables being used by each stack. Common INET only knows the contents of each stack's main route table and will select a stack to used for a connection based on that information. Since the contents of policy-based route tables will likely differ from the contents of the main route table, Common INET may not select the best stack for the connection and may even select a stack on which the connection will fail.

Each policy-based route table that is configured for dynamic routing adds additional processing to OMPROUTE. Duplicate route tables should be avoided and this is ensured if the Configuration Assistant is used to create your policy agent flat-files. In addition, you should avoid having large numbers of policy-based route tables that use dynamic routing.



ibm.com

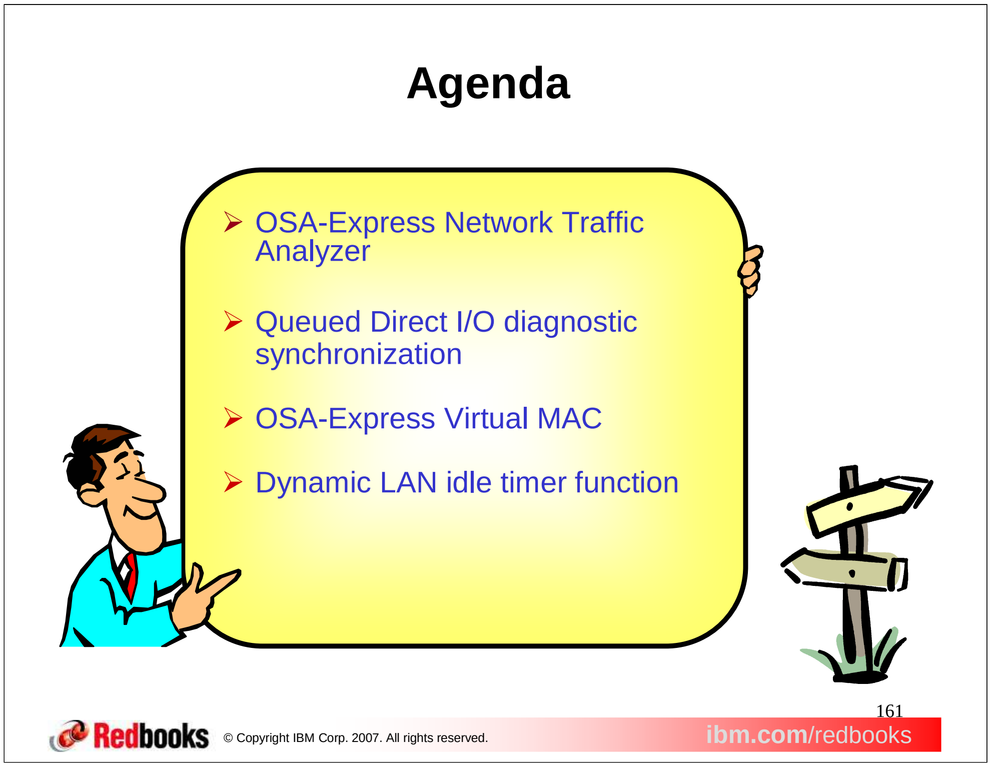
e-business

System z Hardware Exploitation

Redbooks

International Technical Support Organization

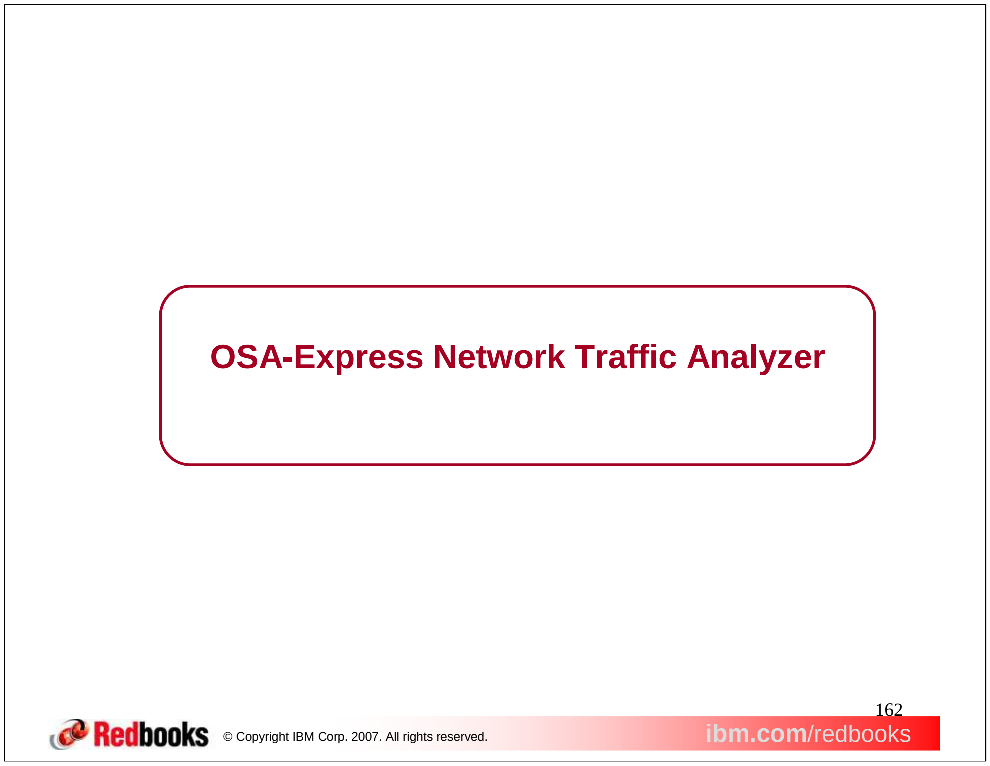
This presentation describes the changes to exploit the system z hardware.



Agenda

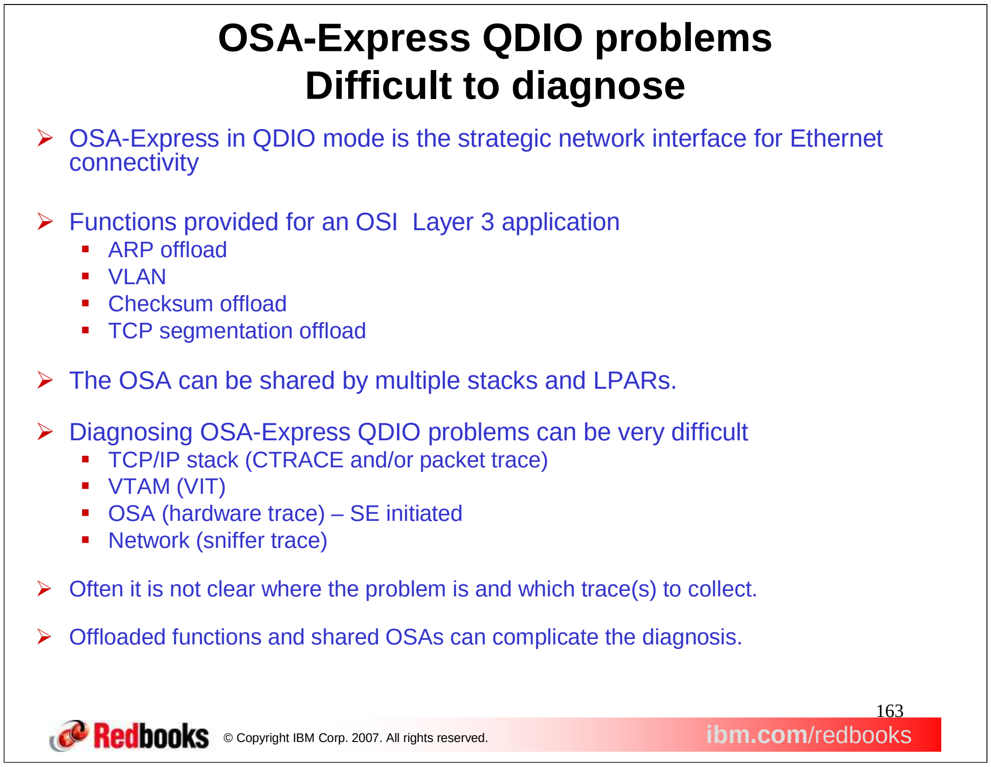
- OSA-Express Network Traffic Analyzer
- Queued Direct I/O diagnostic synchronization
- OSA-Express Virtual MAC
- Dynamic LAN idle timer function

In this presentation we will be discussing the functions that exploit the system z hardware.



OSA-Express Network Traffic Analyzer

This section describes the z/OS Communications Server implementation of the OSA-Express Network Traffic Analyzer.



OSA-Express QDIO problems Difficult to diagnose

- OSA-Express in QDIO mode is the strategic network interface for Ethernet connectivity
- Functions provided for an OSI Layer 3 application
  - ARP offload
  - VLAN
  - Checksum offload
  - TCP segmentation offload
- The OSA can be shared by multiple stacks and LPARs.
- Diagnosing OSA-Express QDIO problems can be very difficult
  - TCP/IP stack (CTRACE and/or packet trace)
  - VTAM (VIT)
  - OSA (hardware trace) – SE initiated
  - Network (sniffer trace)
- Often it is not clear where the problem is and which trace(s) to collect.
- Offloaded functions and shared OSAs can complicate the diagnosis.

163

Redbooks

ibm.com/redbooks

The Open System Adapter (OSA) Express operating in Queued Direct Input/Output (QDIO) mode is the strategic network interface for Internet Protocol (IP) communications for the z/Series line of processors.

The z/OS Communications Server provides the interface to the OSA-Express for IP when a TRLE definition is configured for VTAM and the IPv4 Device and LINK statements or the IPv6 INTERFACE statement is configured for IP. The VTAM component of Communications Server provides the device driver interface between the OSA and IP component with a TRLE definition. The TRLE entry in a VTAM TRL major node needs at least one DATAPATH address for each TCPIP stack on the LPAR. The PORTNAME value on the TRLE statement is the name that is the same value used on the DEVICE and INTERFACE configuration statements.

The OSI Layer 3 functions of ARP offload, VLAN, Checksum offload and TCP segmentation offload can be utilized by IP. By moving these function to the OSA, the Communications Server reduces the CPU load on the main processors.

In addition, multiple instances of the Communication Server in one LPAR or in multiple LPARs can share an OSA.

IP network problems can be complicated to resolve. Is the problem outside of the OSA? Is the network router not receiving packets from the OSA or not sending packets to the OSA? Is the problem inside the LPAR? Is the application or communications server not sending packets to the OSA or not receiving packets from the OSA? Tracking down the where in the network path a packet is lost may require traces and logs from many different sources:

•Application logs showing that a network session is active and processing network data.

•NETSTAT displays of active connections and the routes that are active for those connections.

•Packet traces and system traces taken by the communications server.

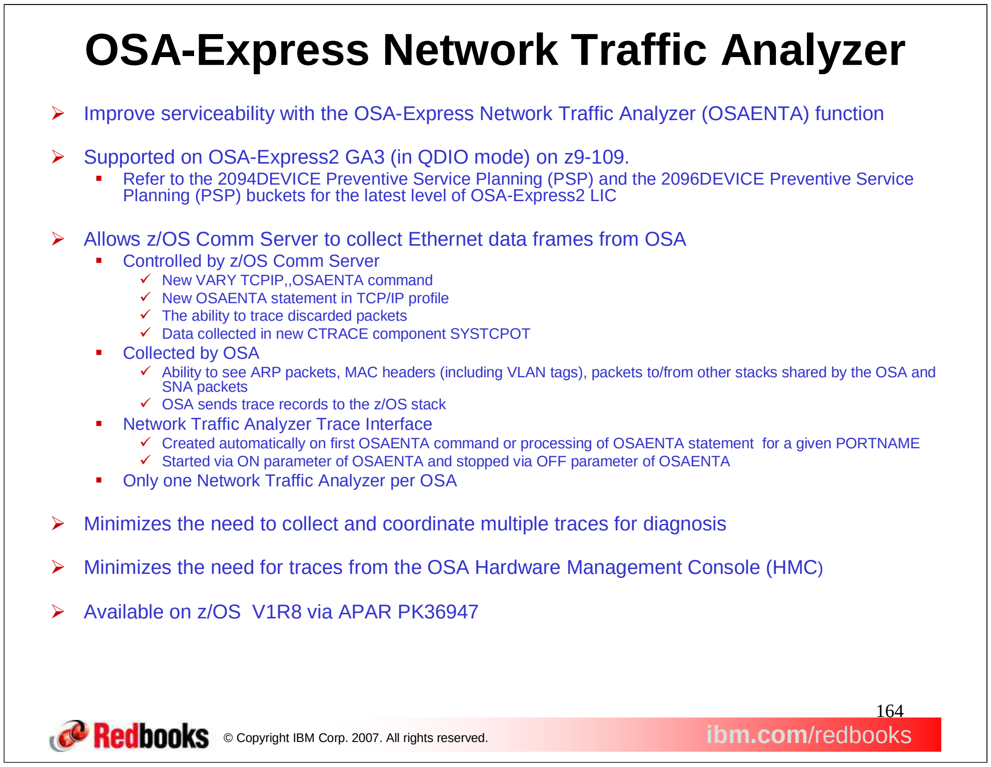
•VTAM traces of the device driver processing the network data.

•OSA hardware traces and logs

•Sniffer traces taken from routers and switches along the network path.

•And the above traces at the other end of the network path.

In addition the offloaded functions and shared OSAs cause further complications. The OSA hardware trace requires IBM SE personnel to be on-site to initiate the trace function from the Hardware Maintenance Console (HMC).



OSA-Express Network Traffic Analyzer

- Improve serviceability with the OSA-Express Network Traffic Analyzer (OSAENTA) function
- Supported on OSA-Express2 GA3 (in QDIO mode) on z9-109.
  - Refer to the 2094DEVICE Preventive Service Planning (PSP) and the 2096DEVICE Preventive Service Planning (PSP) buckets for the latest level of OSA-Express2 LIC
- Allows z/OS Comm Server to collect Ethernet data frames from OSA
  - Controlled by z/OS Comm Server
    - ✓ New VARY TCPIP,,OSAENTA command
    - ✓ New OSAENTA statement in TCP/IP profile
    - ✓ The ability to trace discarded packets
    - ✓ Data collected in new CTRACE component SYSTCPOT
  - Collected by OSA
    - ✓ Ability to see ARP packets, MAC headers (including VLAN tags), packets to/from other stacks shared by the OSA and SNA packets
    - ✓ OSA sends trace records to the z/OS stack
  - Network Traffic Analyzer Trace Interface
    - ✓ Created automatically on first OSAENTA command or processing of OSAENTA statement for a given PORTNAME
    - ✓ Started via ON parameter of OSAENTA and stopped via OFF parameter of OSAENTA
  - Only one Network Traffic Analyzer per OSA
- Minimizes the need to collect and coordinate multiple traces for diagnosis
- Minimizes the need for traces from the OSA Hardware Management Console (HMC)
- Available on z/OS V1R8 via APAR PK36947

The OSA-Express Network Traffic Analyzer (OSAENTA) function is designed to provide the serviceability function for OSA-Express by collecting packet traces between the z/Series processors and the LAN connected to the OSA. OSAENTA is supported on OSA-Express 2 GA3 on the z9-109 class of processors. It will also require an upgrade of the OSA-Express LIC. To enable the OSA-Express network traffic analyzer, which may be referred to as NTA or OSAENTA, you must be running at least an IBM System z9 EC or z9 BC and OSA-Express2 in QDIO mode (CHPID type OSD). Refer to the 2094DEVICE Preventive Service Planning (PSP) and the 2096DEVICE Preventive Service Planning (PSP) buckets for further information.

The z/OS Communications Server OSAENTA function is used to control the trace process in an OSA. The OSAENTA statement may be in the profile data set or in a Vary OBEYFILE data set or issued as console command using VARY TCPIP,,OSAENTA. There are control parameters which start and stop the trace, tell the OSA how much data to collect out of each packet and when to automatically stop the trace. The filter parameters tell the OSA which packets to capture.

Both the OSAENTA statement in the TCPIP profile and the VARY TCPIP,,OSAENTA command provide equivalent function with common syntax. The only difference in syntax is that the command parameters are separated by commas while the profile statement parameters can be separated by blanks. This presentation uses the term "OSAENTA command" to refer to either the OSAENTA profile statement or the VARY TCPIP,,OSAENTA command.

Packets that are being discarded are filtered only by the DISCARD parameter. The filters provided have no effect of the collection of discarded packets. A discard reason code is associated with each discarded packet. A packet may be discarded for exceptional reasons or as part of the usual discard processing. The reason a packet is discarded is broken into two groups, exceptional reasons and the usual reasons. An exceptional reason can be no buffers available, the destination IP address is not registered or there is no default router stack. A usual reason can be Ethernet protocol not supported by the OSA or a VLAN is not registered.

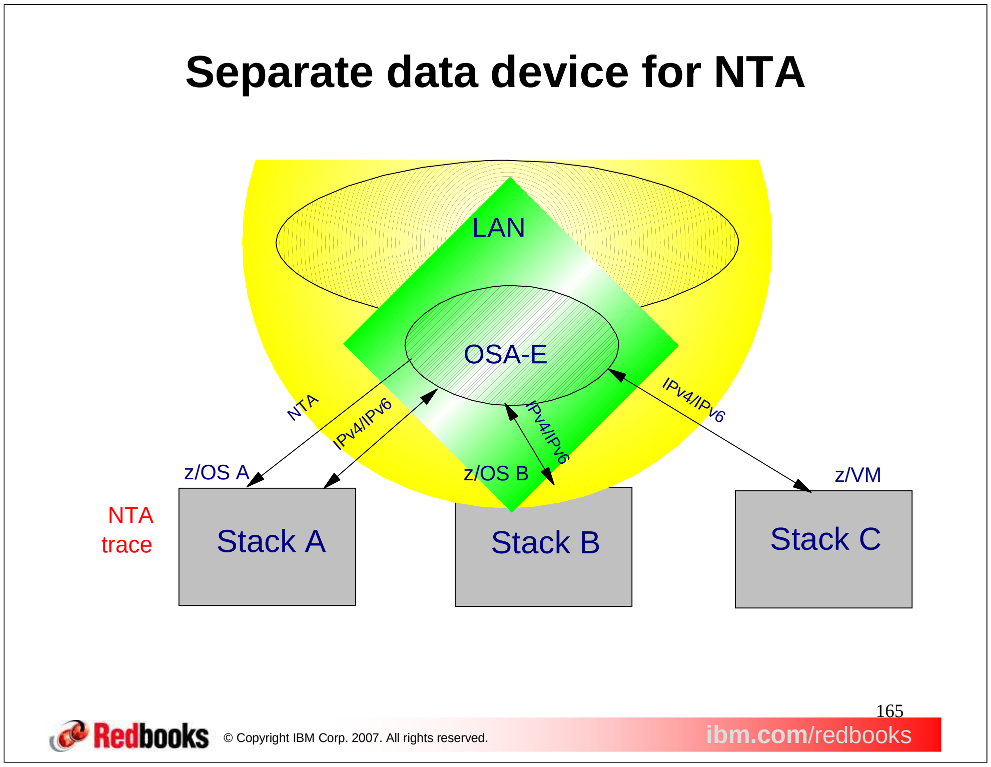
Using the currently available z/OS facilities (CTRACE and IPCS) the trace data can be written to z/OS data sets and formatted. In addition the OSAENTA facility captures the Ethernet header which is not available with the current PKTTRACE command. The Ethernet header includes the MAC addresses, the VLAN tag, and the other 802.3 fields. Packets for other protocols not currently seen by the z/OS IP Communications Server such as ARP and SNA packets can be captured. Packets sent and received from other devices shared by the OSA can also be captured. These include IP stacks in the same LPAR, in other LPARs running z/VM, z/Linux and z/OS. This also includes other z/OS images with different releases of z/OS.

For the z/OS Communications Server an internal interface for NTA is created the first time a new PORTNAME is encountered on a OSAENTA statement or command. The interface appears as a TCP/IP interface. A home IP address is not associated with this interface. The interface name is EZANTA concatenated with the port name. This EZANTA interface is displayed at the end of the NETSTAT DEV output. When the ON keyword of the OSAENTA command is specified VTAM allocates the next available TRLE data path associated with the port. This data path is used only for inbound trace data. When the OFF keyword is used (or the trace limits of the TIME, DATA or FRAMES keyword are reached) the data path is released. The NTA trace interface is neither IPv4 nor IPv6. The stack automatically sets READSTORAGE to MAX(4 MB) and INBERF to MINCPU for the NTA trace interface.

There are some restrictions using the OSAENTA command. The OSA does not support multiple stacks activating the trace at the same time. Once the OSAENTA OFF command has been issued, another stack may start trace. The security authorization in the HMC is required to be set to CHPID to see packets from other operating system images. LPAR to LPAR traffic does not go over the LAN but is handled directly by the OSA. As such there are no MAC headers for these packets. This function applies only to OSA-Express2 Ethernet-type adapters configured in QDIO mode. Data to and from an OSA-Express2 adapter configured in Network Control Program (OSN) mode cannot be traced. SNA data tracing is currently limited to Enterprise Extender data when the OSA-Express adapter is configured in QDIO Layer 3 mode, and data to and from Communication Controller for Linux (CCL) on System z (TM) when the OSA-Express adapter is configured in QDIO Layer 2 mode. Data sent or received over the control devices are not traced. These include the IP assist commands and the OSA-Express SNMP subagent packets.

The OSA collects the data when it is sent across the PCI to the physical port (sometimes referred to as the NIC). The OSA also collects data for LPAR-LPAR packets which do not go onto the LAN. The SNA data collected is limited to Enterprise Extender data when OSA is configured in QDIO layer 3 mode and data to/from Communication Controller for Linux (CCL) on System z when OSA is configured in QDIO layer 2 mode. OSA supports only one stack sharing the OSA to perform NTA tracing. One stack can perform NTA tracing for multiple OSAs.

By collecting the Ethernet data frames OSAENTA make it easier to collect trace data from multiple sources. The Hardware Management Console (HMC) will only be used to the set the security setting for collecting trace data. Each OSA can have one of three security levels



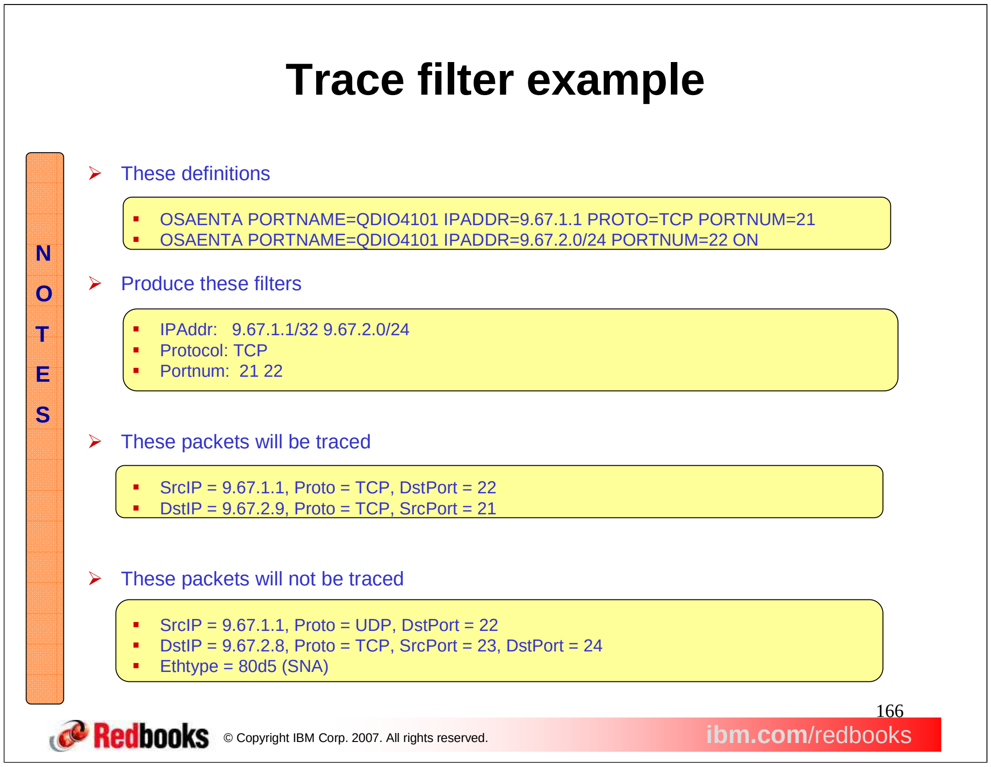
Separate data device for NTA

This diagram shows OSAENTA collecting data from an OSA-Express2 over a separate data path from a z/OS Communications Server LPAR. This z/OS A LPAR has a data path for the IPv4 and IPv6 network traffic. In addition the OSA is shared with another z/OS LPAR, z/OS B. and a z/VM LPAR. OSAENTA can collect packets flowing to and from each of these LPARs and to and from the LAN.

In this configuration Stack A has defined two DATAPATH addresses in its TRLE definition. One will be used for IP communications and the other for OSAENTA. If there are multiple IP stacks on z/OS A, then there must be one DATAPATH address for each IP stack and one for OSAENTA.

This example shows a stack which is using the OSA for IPv4/IPv6 data and also for NTA tracing. This configuration requires a TRLE with at least two data devices. Stack A activates the OSA for IPv4 and IPv6. This causes z/OS Communications Sever to allocate one data device which is shared for IPv4 and IPv6 data. Stack A also activates the OSA for NTA tracing. This causes z/OS CS to allocate another data device which is used exclusively for NTA.

Another alternative is to have a dedicated stack for NTA. In this configuration, Stack A could be a a test system with a single DATAPATH address used exclusively by OSAENTA to capture packets from the other LPARs sharing the OSA, Stack B and Stack C. In this way Stack A will absorb the CPU cycles needed to process and write the trace data and minimize the CPU impact to the other LPARs. Of course, the impact to the OSA for the overhead of capturing and forwarding the trace data will remain the same in this configuration as in the previous configuration.



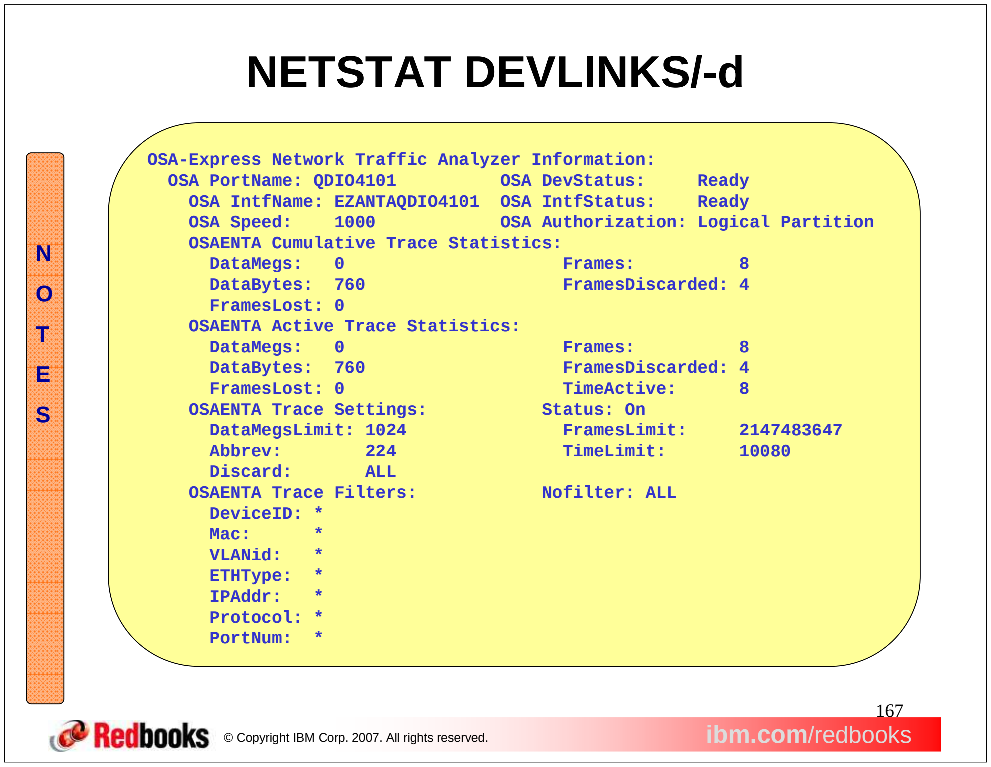
Trace filter example

N
O
T
E
S

- These definitions
  - OSAENTA PORTNAME=QDIO4101 IPADDR=9.67.1.1 PROTO=TCP PORTNUM=21
  - OSAENTA PORTNAME=QDIO4101 IPADDR=9.67.2.0/24 PORTNUM=22 ON
- Produce these filters
  - IPAddr: 9.67.1.1/32 9.67.2.0/24
  - Protocol: TCP
  - Portnum: 21 22
- These packets will be traced
  - SrcIP = 9.67.1.1, Proto = TCP, DstPort = 22
  - DstIP = 9.67.2.9, Proto = TCP, SrcPort = 21
- These packets will not be traced
  - SrcIP = 9.67.1.1, Proto = UDP, DstPort = 22
  - DstIP = 9.67.2.8, Proto = TCP, SrcPort = 23, DstPort = 24
  - Ethtype = 80d5 (SNA)

Redbooks

In this example we show the effects of using the different filters. Two OSAENTA commands specify selection based upon two IP addresses, the TCP protocol and two port numbers. If a packet matches all the criteria of the filters then it would be traced. Since in this example, the IPADDR, PORTOCOL and PORTNUM filters were used, then packets that were not ethernet type of IPv4 would not be traced.



NETSTAT DEVLINKS/-d

```
OSA-Express Network Traffic Analyzer Information:
  OSA PortName: QDIO4101          OSA DevStatus:     Ready
    OSA IntfName: EZANTAQDIO4101  OSA IntfStatus:    Ready
    OSA Speed:    1000            OSA Authorization: Logical Partition
    OSAENTA Cumulative Trace Statistics:
      DataMegs:   0                     Frames:          8
      DataBytes:  760                   FramesDiscarded: 4
      FramesLost: 0
    OSAENTA Active Trace Statistics:
      DataMegs:   0                     Frames:          8
      DataBytes:  760                   FramesDiscarded: 4
      FramesLost: 0                     TimeActive:      8
    OSAENTA Trace Settings:           Status: On
      DataMegsLimit: 1024               FramesLimit:     2147483647
      Abbrev:        224                TimeLimit:       10080
      Discard:       ALL
    OSAENTA Trace Filters:            Nofilter: ALL
      DeviceID: *
      Mac:      *
      VLANid:   *
      ETHType:  *
      IPAddr:   *
      Protocol: *
      PortNum:  *
```

167

This slide shows the partial output from a NETSTAT DEVLINKS/-d command relating to the Network Traffic Analyzer.

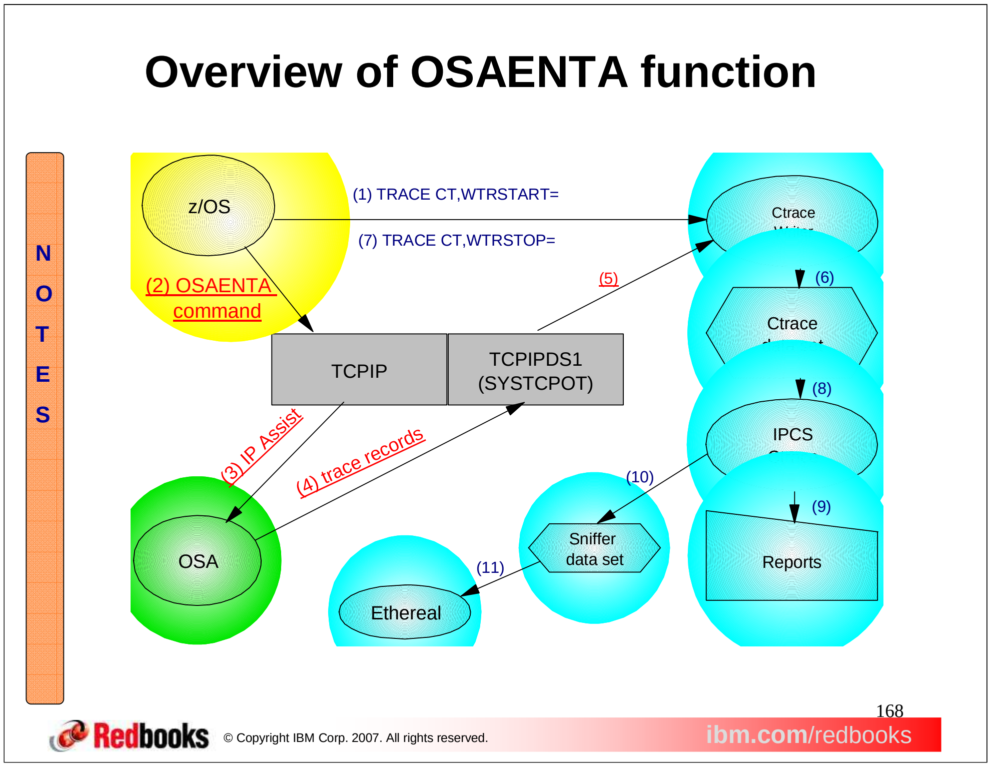
The display is divided into five sections. The first section shows the OSA portname, the interface name and status of the interface. In addition the last known security setting associated with the OSA is shown. If the interface has never been active, then UNKNOWN will be shown.

The second section is the statistics since the first time the interface was defined to TCPIP. Note that when the OSAENTA DELETE command is executed then these values are lost.

The third section is the statistics since the last time a OSAENTA ON command was issued.

The fourth section is the current trace settings for DATA, FRAMES, ABBREV, TIME and DISCARD.

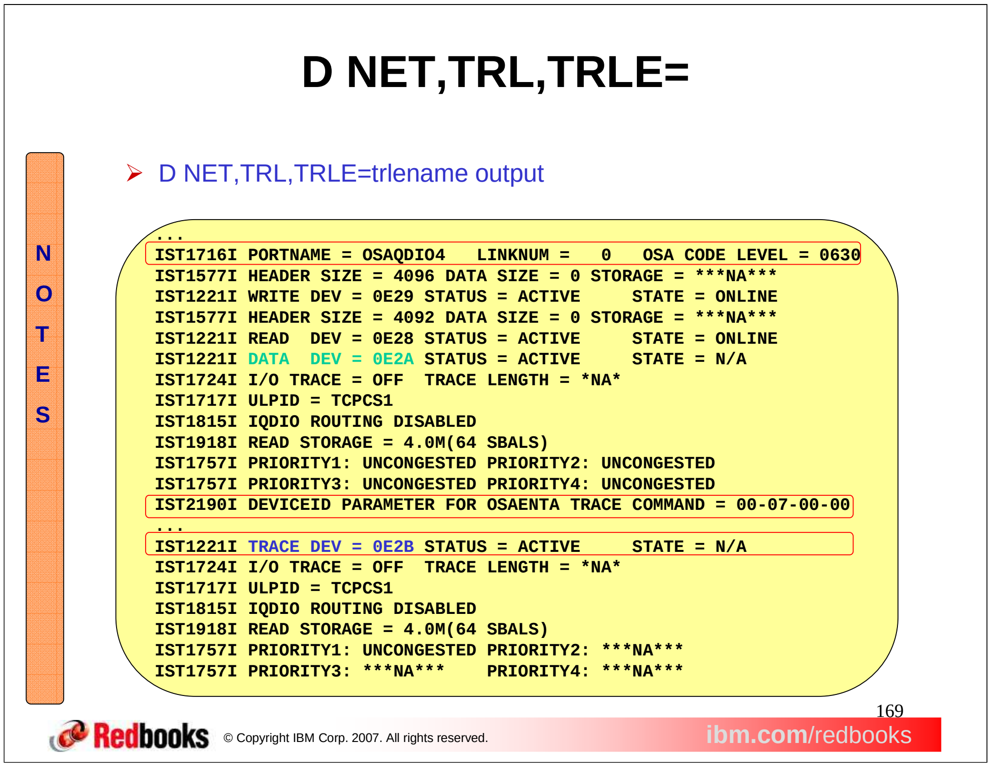
The fifth section is the current filter values. If NOFILTER has been set to NONE, or left to default, and all the filters are set *, then the filter values are not displayed.



Overview of OSAENTA function

This diagram shows the process of collecting trace data.

1. A MVS Ctrace external writer is started with the TRACE CT,WTRSTART=*wtrprocname*
2. The VARY OSAENTA,PORTNAME=*osaname*,ON command is issued to start the collection process
3. TCP/IP creates the required control blocks for the OSAENTA command. The interface name is EZANTA*osaname* . Then IP Assist orders are issued to create the data channel for the trace records and request the tracing commence using filters and parameters from the OSAENTA command (or commands).
4. The OSA collects the trace records into buffers for transmission to TCPIP. The same mechanism for regular data transfer is used to transfer the trace buffers.
5. TCPIP moves the trace data into the TCPIPDS1 data space buffers reserved for the SYSTCPOT Ctrace component. As the Ctrace buffers are filled they are written by the Ctrace external writer.
6. The Ctrace external writer copies the buffers into the Ctrace data sets allocated in the Ctrace writer procedure.
7. The operator disconnects the writer from TCPIP and stops the writer.
8. The IPCS CTRACE subcommand can now be used to select packets and format reports.
9. The same reports available for packet trace (SYSTCPDA) are available for OSA trace (SYSTCPOT).
10. The packets can also be copied to a sniffer formatted data set (OPTIONS ((SNIFFER))).
11. Once the sniffer data set has been downloaded to a PC, programs such as Ethereal can be used to further analyze the packet data.



D NET,TRL,TRLE=

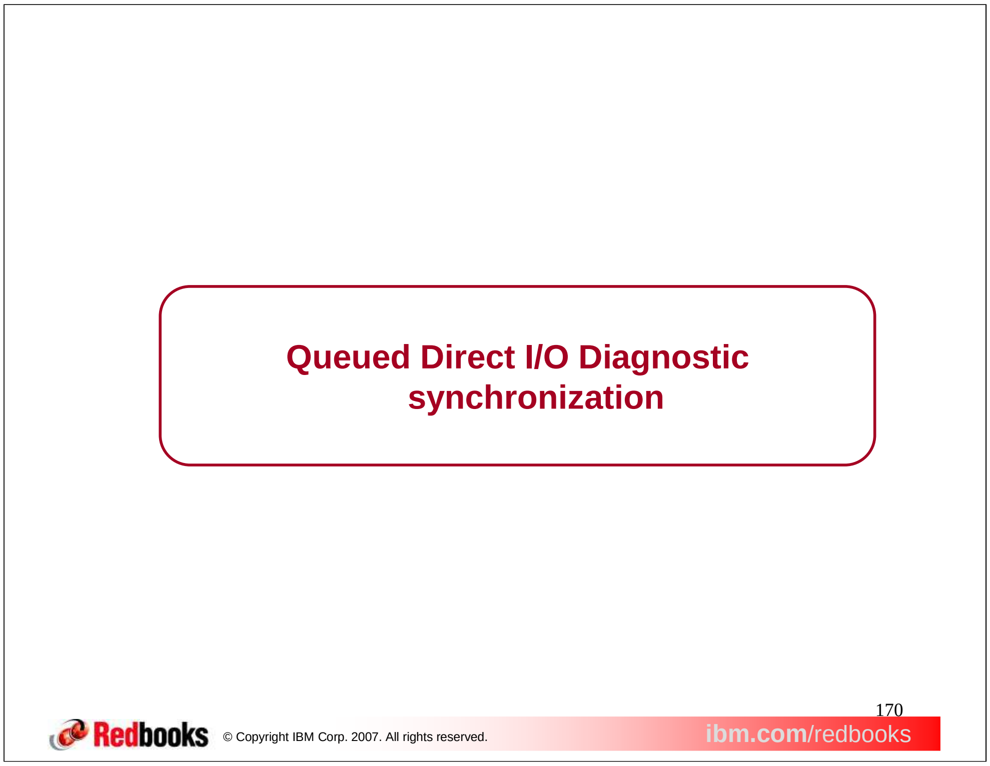
- D NET,TRL,TRLE=trlename output

```
...
IST1716I PORTNAME = OSAQDIO4   LINKNUM =   0   OSA CODE LEVEL = 0630
IST1577I HEADER SIZE = 4096 DATA SIZE = 0 STORAGE = ***NA***
IST1221I WRITE DEV = 0E29 STATUS = ACTIVE     STATE = ONLINE
IST1577I HEADER SIZE = 4092 DATA SIZE = 0 STORAGE = ***NA***
IST1221I READ  DEV = 0E28 STATUS = ACTIVE     STATE = ONLINE
IST1221I DATA  DEV = 0E2A STATUS = ACTIVE     STATE = N/A
IST1724I I/O TRACE = OFF  TRACE LENGTH = *NA*
IST1717I ULPID = TCPCS1
IST1815I IQDIO ROUTING DISABLED
IST1918I READ STORAGE = 4.0M(64 SBALS)
IST1757I PRIORITY1: UNCONGESTED PRIORITY2: UNCONGESTED
IST1757I PRIORITY3: UNCONGESTED PRIORITY4: UNCONGESTED
IST2190I DEVICEID PARAMETER FOR OSAENTA TRACE COMMAND = 00-07-00-00
...
IST1221I TRACE DEV = 0E2B STATUS = ACTIVE     STATE = N/A
IST1724I I/O TRACE = OFF  TRACE LENGTH = *NA*
IST1717I ULPID = TCPCS1
IST1815I IQDIO ROUTING DISABLED
IST1918I READ STORAGE = 4.0M(64 SBALS)
IST1757I PRIORITY1: UNCONGESTED PRIORITY2: ***NA***
IST1757I PRIORITY3: ***NA***    PRIORITY4: ***NA***
```

169

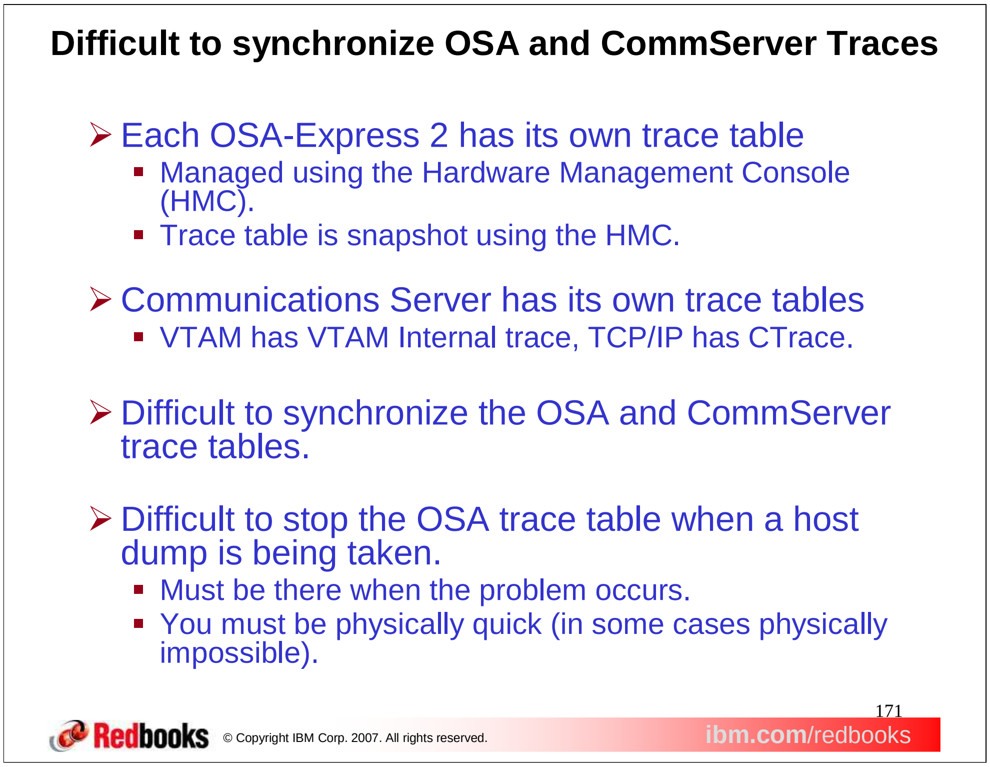
Redbooks ibm.com/redbooks

This is the output of the VTAM display for an active OSA TRLE definition. Message IST1716I shows the OSA code level. Note that 0630 is **NOT** the code level that supports OSAENTA. Message IST2190I shows the DEVICEID value. Message IST1221I shows the device address and status of the trace DATAPATH device.



Queued Direct I/O Diagnostic synchronization

This solution is also informally referred to as ‘OSA Trap’. This solution was part of z/OS **V1R8** Communications Server and subsequent releases. It is being presented here because OSA support is now available.



Difficult to synchronize OSA and CommServer Traces

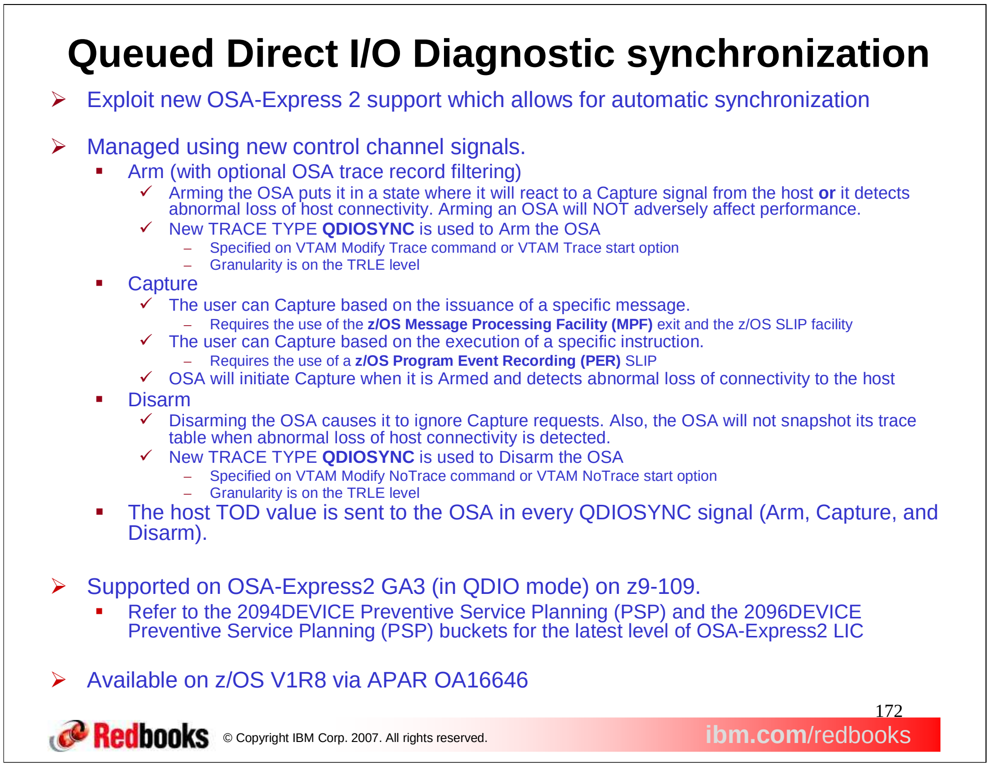
- Each OSA-Express 2 has its own trace table
  - Managed using the Hardware Management Console (HMC).
  - Trace table is snapshot using the HMC.
- Communications Server has its own trace tables
  - VTAM has VTAM Internal trace, TCP/IP has CTrace.
- Difficult to synchronize the OSA and CommServer trace tables.
- Difficult to stop the OSA trace table when a host dump is being taken.
  - Must be there when the problem occurs.
  - You must be physically quick (in some cases physically impossible).

171

Redbooks ibm.com/redbooks

All references to OSA in this presentation implies OSA-Express2 in QDIO mode. Diagnostic information refers primarily to OSA and Communications Server (host) trace tables. It does not preclude other diagnostic information such as counters, error logs, etc. The OSA and the host maintain their own diagnostic information separately and each product's trace tables often are the most important piece of diagnostic information for that product.

This solution originated as a requirement from z/OS Communications Server System Verification Test (SVT). SVT wanted a way to automatically capture diagnostic information from multiple products simultaneously, hoping to minimize recreates.



Queued Direct I/O Diagnostic synchronization

- Exploit new OSA-Express 2 support which allows for automatic synchronization
- Managed using new control channel signals.
  - Arm (with optional OSA trace record filtering)
    - ✓ Arming the OSA puts it in a state where it will react to a Capture signal from the host **or** it detects abnormal loss of host connectivity. Arming an OSA will NOT adversely affect performance.
    - ✓ New TRACE TYPE **QDIOSYNC** is used to Arm the OSA
      - Specified on VTAM Modify Trace command or VTAM Trace start option
      - Granularity is on the TRLE level
  - Capture
    - ✓ The user can Capture based on the issuance of a specific message.
      - Requires the use of the **z/OS Message Processing Facility (MPF)** exit and the z/OS SLIP facility
    - ✓ The user can Capture based on the execution of a specific instruction.
      - Requires the use of a **z/OS Program Event Recording (PER)** SLIP
    - ✓ OSA will initiate Capture when it is Armed and detects abnormal loss of connectivity to the host
  - Disarm
    - ✓ Disarming the OSA causes it to ignore Capture requests. Also, the OSA will not snapshot its trace table when abnormal loss of host connectivity is detected.
    - ✓ New TRACE TYPE **QDIOSYNC** is used to Disarm the OSA
      - Specified on VTAM Modify NoTrace command or VTAM NoTrace start option
      - Granularity is on the TRLE level
  - The host TOD value is sent to the OSA in every QDIOSYNC signal (Arm, Capture, and Disarm).
- Supported on OSA-Express2 GA3 (in QDIO mode) on z9-109.
  - Refer to the 2094DEVICE Preventive Service Planning (PSP) and the 2096DEVICE Preventive Service Planning (PSP) buckets for the latest level of OSA-Express2 LIC
- Available on z/OS V1R8 via APAR OA16646

172

Redbooks

ibm.com/redbooks

This solution is effective only if supported by the OSA-Express 2 feature and enabled on z/OS Communications Server.

New control signals are used between z/OS Communications Server and OSA to facilitate implementation of this solution. z/OS Communications Server uses one of these signals to tell OSA to snapshot its trace table to the HMC hardfile. This new set of control signals is collectively known as SetDiagAsst.

Host initiated refers to z/OS Communications Server sending a trace synchronization command to the OSA. OSA initiated refers to action taken by the OSA without a specific command being received from the host.

Two synchronization states are defined for the OSA Armed and Disarmed. When Armed, if the OSA receives a capture request from the host *or* the OSA loses host connectivity, the OSA will snapshot its diagnostic information (trace table) to the HMC hardfile. When disarmed, the OSA will act no different than prior to this solution. Arming an OSA will NOT adversely affect performance because it simply causes the OSA to change an internal state and this internal state is not interrogated by the OSA during normal data transmission.

There are host initiated captures and an OSA initiated capture . For host initiated captures, z/OS Communications Server tells OSA to snapshot its trace table. For OSA initiated captures, OSA decides on its own to snapshot its trace table. There are 2 methods a user can use to initiate a Capture request from z/OS Communications Server. Note that a Capture is sent to all Armed OSAs. A user can Capture based on the issuance of a specific message. This requires the use of the z/OS Message Processing Facility (MPF) to drive the new V1R8 MPF exit (IUTLLCMP). The user will also need to use the z/OS SLIP facility on the same message(s) to initiate a host dump. The user can also Capture based on the execution of a specific instruction. This requires the use of a z/OS PER type SLIP specifying ACTION=(RECOVERY). In this case the user can use the same PER SLIP to also get a host dump. Of the two host initiated capture mechanisms the MPF and SLIP mechanism is the most useful. The PER SLIP is the less useful and could significantly affect performance. The OSA will initiate a Capture when it is Armed and detects abnormal loss of connectivity to the host (includes any type of Halt subchannel (ex. InOp)).

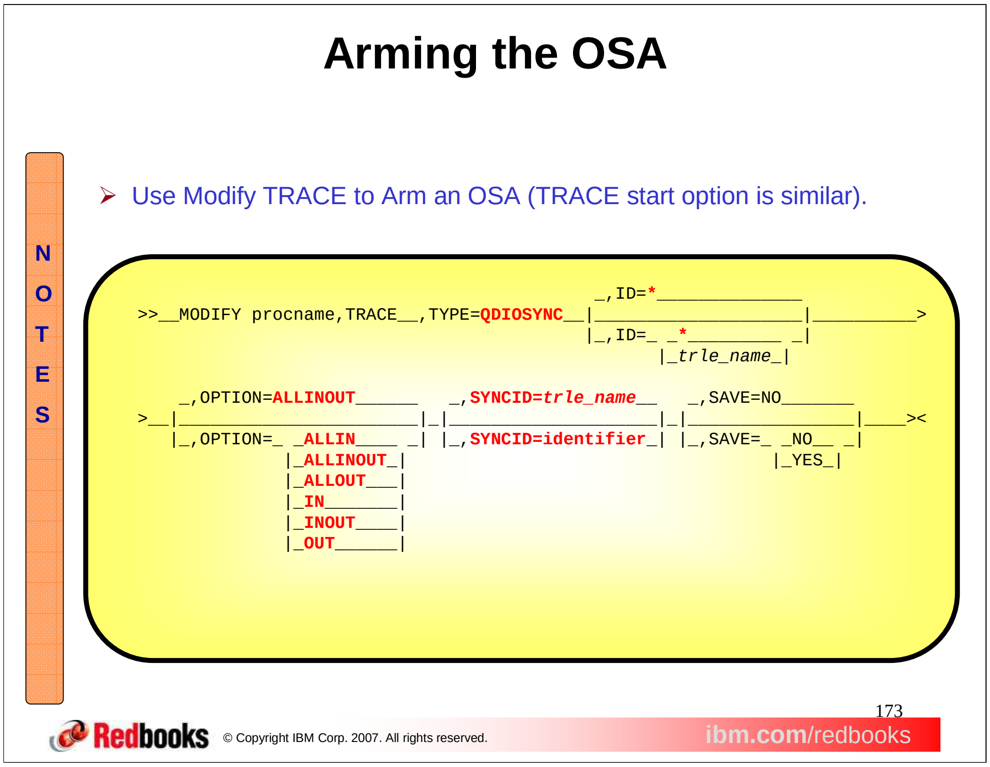
The existing VTAM Trace command and Trace facilities provided a good infrastructure for external control of this solution and was therefore modified to support this solution. VTAM TRACE infrastructure is modified to manage OSA diagnostic synchronization. A new TRACE TYPE **QDIOSYNC** is used to Arm, Disarm, and Display traces. Both the VTAM TRACE/NOTRACE start Option and command are supported. The Arm/Disarm granularity is on the TRLE level, meaning a user can Arm or Disarm ALL devices defined in the TRLE. When Arming, the user can optionally specify which trace records OSA will cut. A word of caution, specifying which trace records OSA should cut should only be used when directed to do so. When Arming the OSA, a user can optionally specify a synchronization correlator used by OSA when it writes its trace table to the HMC hardfile.

If a failure occurs attempting to Arm one of the devices in the TRLE, any other devices in the same TRLE that were previously Armed are forced Disarmed. Subsequent Arm attempts (commands) are rejected once it's discovered the OSA does not support SetDiagAsst or an Arm failure occurred. Only a TRLE recycle will allow a subsequent Arm attempt.

In addition to ID=*trlename*, ID=* is also supported for TYPE=QDIOSYNC on the MODIFY TRACE/NOTRACE command and the TRACE/NOTRACE start option. ID=* Arms or Disarms all OSAs. SAVE=YES is supported which saves the TRACE command and applies it when the TRL major node is activated.

The host TOD value is sent to the OSA in every QDIOSYNC signal (Arm, Capture, and Disarm). This intended to be exposed by the OSA and used to correlate the host and OSA diagnostics.

This solution leverages z/OS facilities as a trigger for the host to initiate an OSA snapshot of its trace table.



Arming the OSA

N
O
T
E
S

- Use Modify TRACE to Arm an OSA (TRACE start option is similar).

```
                                            _,ID=*______________
>>__MODIFY procname,TRACE__,TYPE=QDIOSYNC__|____________________|_________>
                                           |_,ID=_ _*_________ _|
                                                  |_trle_name_|

    _,OPTION=ALLINOUT______   _,SYNCID=trle_name__   _,SAVE=NO_______
>__|_______________________|_|____________________|_|________________|____><
   |_,OPTION=_ _ALLIN____ _| |_,SYNCID=identifier_| |_,SAVE=_ _NO__ _|
              |_ALLINOUT_|                                   |_YES_|
              |_ALLOUT___|
              |_IN_______|
              |_INOUT____|
              |_OUT______|
```

173

Redbooks ibm.com/redbooks

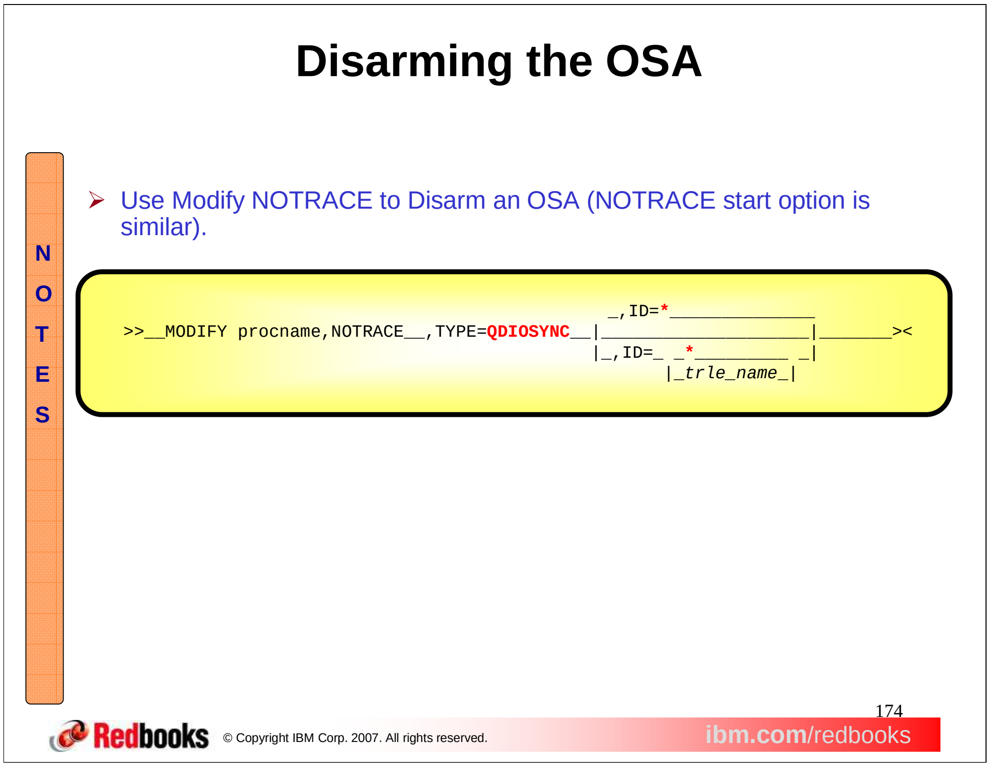
New options and values are highlighted in red. There are new values for the OPTION parameter. ALLINOUT should be used unless directed to do otherwise. ALLIN directs OSA to collect only inbound diagnostic information for all devices. ALLOUT directs OSA to collect only outbound diagnostic information for all devices. ALLINOUT directs OSA to collect inbound and outbound diagnostic information for all devices. IN directs OSA to collect only inbound diagnostic information for devices defined to this VTAM. OUT directs OSA to collect only outbound diagnostic information for devices defined to this VTAM. INOUT directs OSA to collect inbound and outbound diagnostic information for devices defined to this VTAM. Note that OSA currently does not support IN, OUT and INOUT. They can be specified, but OSA will convert them to ALLIN, ALLOUT, and ALLINOUT respectively.

SyncID is simply a (user chosen) EBCDIC correlator value passed to OSA on an Arm request (OSA will convert to ASCII and use it in its diagnostic information).

The asterisk value is now accepted for the ID keyword when TYPE=QDIOSYNC and means all TRLEs. The asterisk is NOT a wildcard variable and if used must be the only character specified.

You can issue Modify Trace even if the OSA is already Armed, which effectively updates the values.

Please note that SAVE=NO is the default for the modify trace command where SAVE=YES is the default for the trace start option.



Disarming the OSA

N O T E S

- Use Modify NOTRACE to Disarm an OSA (NOTRACE start option is similar).

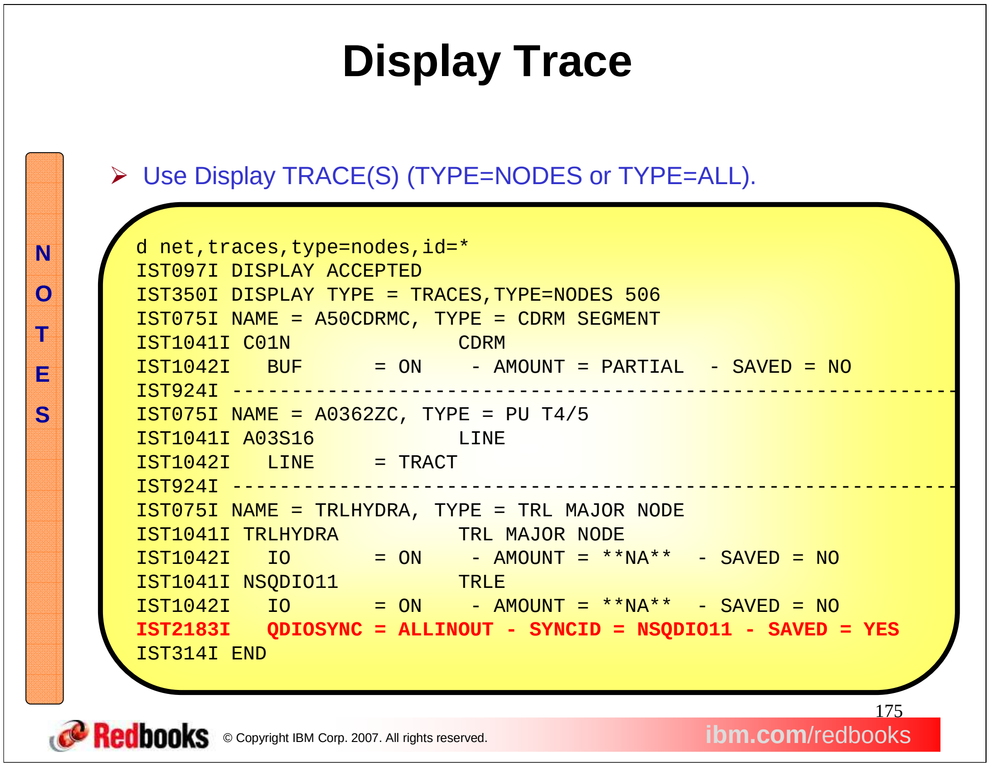
```
                                              _,ID=*_______________
>>__MODIFY procname,NOTRACE__,TYPE=QDIOSYNC__|____________________|________><
                                             |_,ID=_ _*__________ _|
                                                    |_trle_name_|
```

174

ibm.com/redbooks

New options and values are highlighted in red.

The asterisk value is now accepted for the ID keyword when TYPE=QDIOSYNC and means all TRLEs. The asterisk is NOT a wildcard variable and if used must be the only character specified.
The Vary TCPIP,tcpprocname,STOP command results in an automatic Disarm if the OSA is Armed.



Display Trace

N
O
T
E
S

- Use Display TRACE(S) (TYPE=NODES or TYPE=ALL).

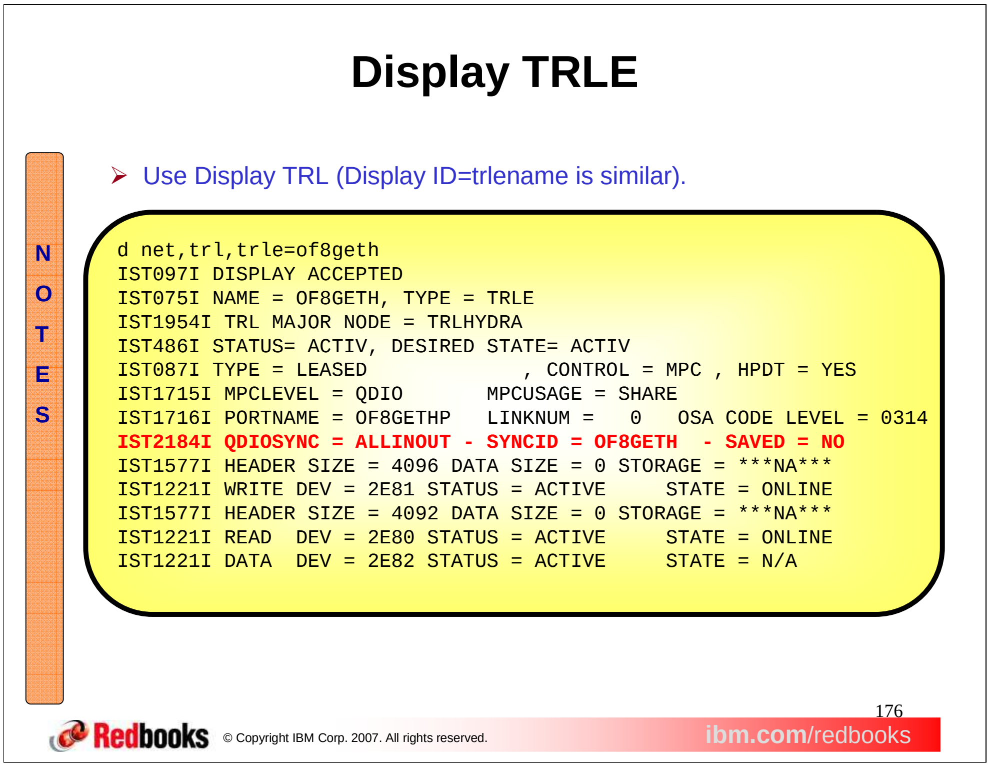
```
d net,traces,type=nodes,id=*
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TRACES,TYPE=NODES 506
IST075I NAME = A50CDRMC, TYPE = CDRM SEGMENT
IST1041I C01N              CDRM
IST1042I   BUF      = ON     - AMOUNT = PARTIAL  - SAVED = NO
IST924I ------------------------------------------------------------
IST075I NAME = A0362ZC, TYPE = PU T4/5
IST1041I A03S16            LINE
IST1042I   LINE     = TRACT
IST924I ------------------------------------------------------------
IST075I NAME = TRLHYDRA, TYPE = TRL MAJOR NODE
IST1041I TRLHYDRA          TRL MAJOR NODE
IST1042I   IO       = ON     - AMOUNT = **NA**  - SAVED = NO
IST1041I NSQDIO11          TRLE
IST1042I   IO       = ON     - AMOUNT = **NA**  - SAVED = NO
IST2183I   QDIOSYNC = ALLINOUT - SYNCID = NSQDIO11 - SAVED = YES
IST314I END
```

175

Redbooks ibm.com/redbooks

A new message is added to the Display response and is highlighted in red. The message is only issued if the TRLE is Armed (or will be armed when activated).

In this case the OPTION specified on (or defaulted for) the Trace command/start option is ALLINOUT. ALLINOUT means OSA is to collect diagnostic data pertaining to ALL LPARs and in both directions. The SYNCID is NSQDIO11, which is being retained by the OSA in case a capture occurs (the SYNCID value will be apparent in the OSA diagnostic data). The QDIOSYNC trace is also saved meaning it is effected when the TRLE or any of its devices are activated.



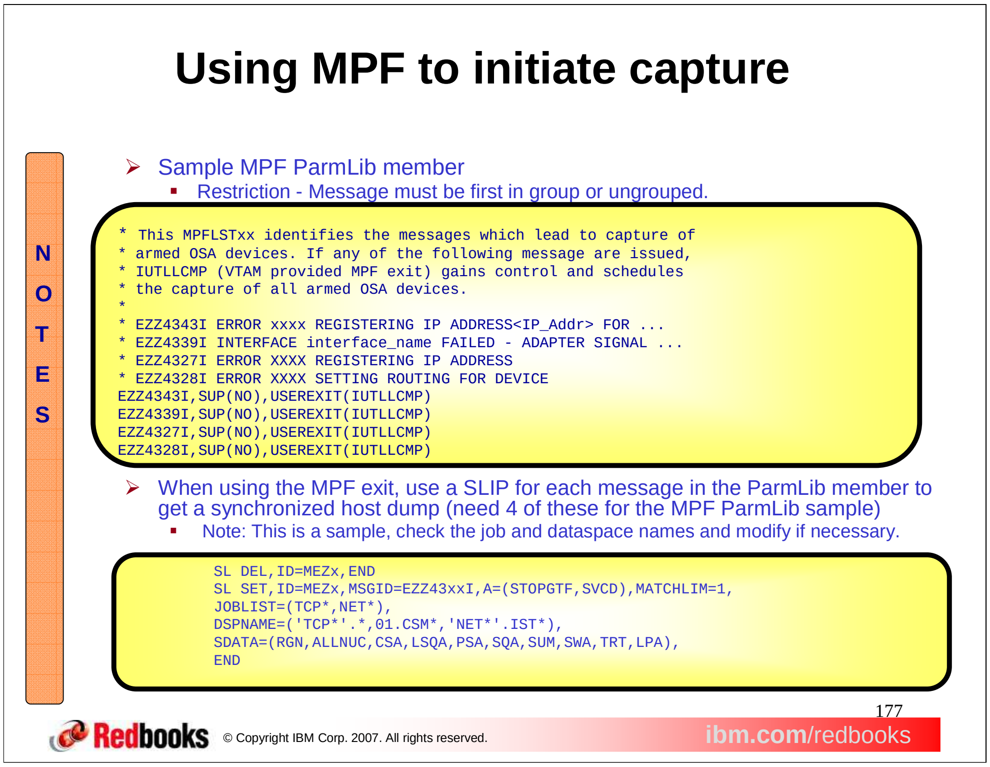
Display TRLE

N O T E S

- Use Display TRL (Display ID=trlename is similar).

```
d net,trl,trle=of8geth
IST097I DISPLAY ACCEPTED
IST075I NAME = OF8GETH, TYPE = TRLE
IST1954I TRL MAJOR NODE = TRLHYDRA
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST087I TYPE = LEASED             , CONTROL = MPC , HPDT = YES
IST1715I MPCLEVEL = QDIO       MPCUSAGE = SHARE
IST1716I PORTNAME = OF8GETHP   LINKNUM =    0   OSA CODE LEVEL = 0314
IST2184I QDIOSYNC = ALLINOUT - SYNCID = OF8GETH  - SAVED = NO
IST1577I HEADER SIZE = 4096 DATA SIZE = 0 STORAGE = ***NA***
IST1221I WRITE DEV = 2E81 STATUS = ACTIVE     STATE = ONLINE
IST1577I HEADER SIZE = 4092 DATA SIZE = 0 STORAGE = ***NA***
IST1221I READ  DEV = 2E80 STATUS = ACTIVE     STATE = ONLINE
IST1221I DATA  DEV = 2E82 STATUS = ACTIVE     STATE = N/A
```

A new message is added to the Display response and is highlighted in red. The message is only issued if the TRLE is Armed (or will be armed when activated).



N O T E S

Using MPF to initiate capture

- Sample MPF ParmLib member
  - Restriction - Message must be first in group or ungrouped.

```
* This MPFLSTxx identifies the messages which lead to capture of
* armed OSA devices. If any of the following message are issued,
* IUTLLCMP (VTAM provided MPF exit) gains control and schedules
* the capture of all armed OSA devices.
*
* EZZ4343I ERROR xxxx REGISTERING IP ADDRESS<IP_Addr> FOR ...
* EZZ4339I INTERFACE interface_name FAILED - ADAPTER SIGNAL ...
* EZZ4327I ERROR XXXX REGISTERING IP ADDRESS
* EZZ4328I ERROR XXXX SETTING ROUTING FOR DEVICE
EZZ4343I,SUP(NO),USEREXIT(IUTLLCMP)
EZZ4339I,SUP(NO),USEREXIT(IUTLLCMP)
EZZ4327I,SUP(NO),USEREXIT(IUTLLCMP)
EZZ4328I,SUP(NO),USEREXIT(IUTLLCMP)
```

- When using the MPF exit, use a SLIP for each message in the ParmLib member to get a synchronized host dump (need 4 of these for the MPF ParmLib sample)
  - Note: This is a sample, check the job and dataspace names and modify if necessary.

```
SL DEL,ID=MEZx,END
SL SET,ID=MEZx,MSGID=EZZ43xxI,A=(STOPGTF,SVCD),MATCHLIM=1,
JOBLIST=(TCP*,NET*),
DSPNAME=('TCP*'.*,01.CSM*,'NET*'.IST*),
SDATA=(RGN,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT,LPA),
END
```

177

Redbooks ibm.com/redbooks

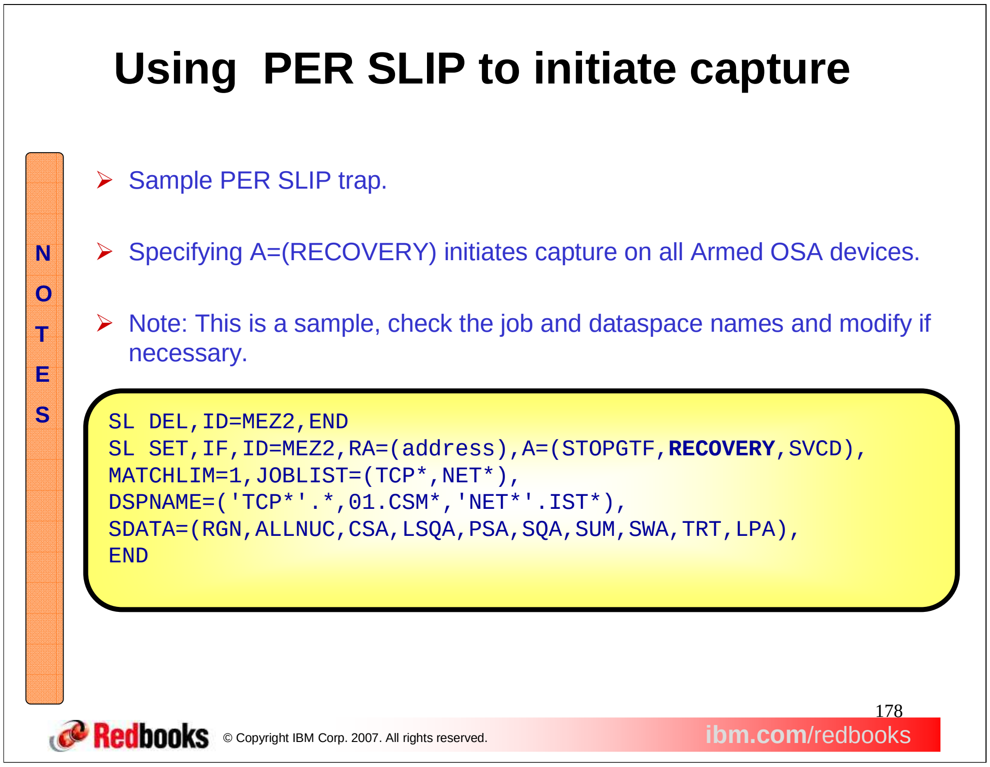
Using the MPF exit will tend to have an insignificant effect on performance.

Additional information on the MPFLSTxx ParmLib member can be found in the z/OS MVS publications. Search on ‘MPFLSTxx’.

The messages in this list are indicative of OSA errors commonly seen by our System Verification Test (SVT) group.

When z/OS Communications Server issues any message in this ParmLib, z/OS generates a call to IUTLLCMP which locates all Armed OSAs and builds and sends each of them a Capture request.

In order to get ‘matching’ host documentation for the OSA trace table, create a message SLIP for each of the 4 messages. This set should be repeated 4 times using MEZ1, MEZ2, MEZ3, and MEZ4 with the corresponding message number change. Order of the SLIPs and order of the ParmLib messages need not match, as log as there’s a SLIP for each message in the ParmLib.



Using PER SLIP to initiate capture

N O T E S

- Sample PER SLIP trap.
- Specifying A=(RECOVERY) initiates capture on all Armed OSA devices.
- Note: This is a sample, check the job and dataspace names and modify if necessary.

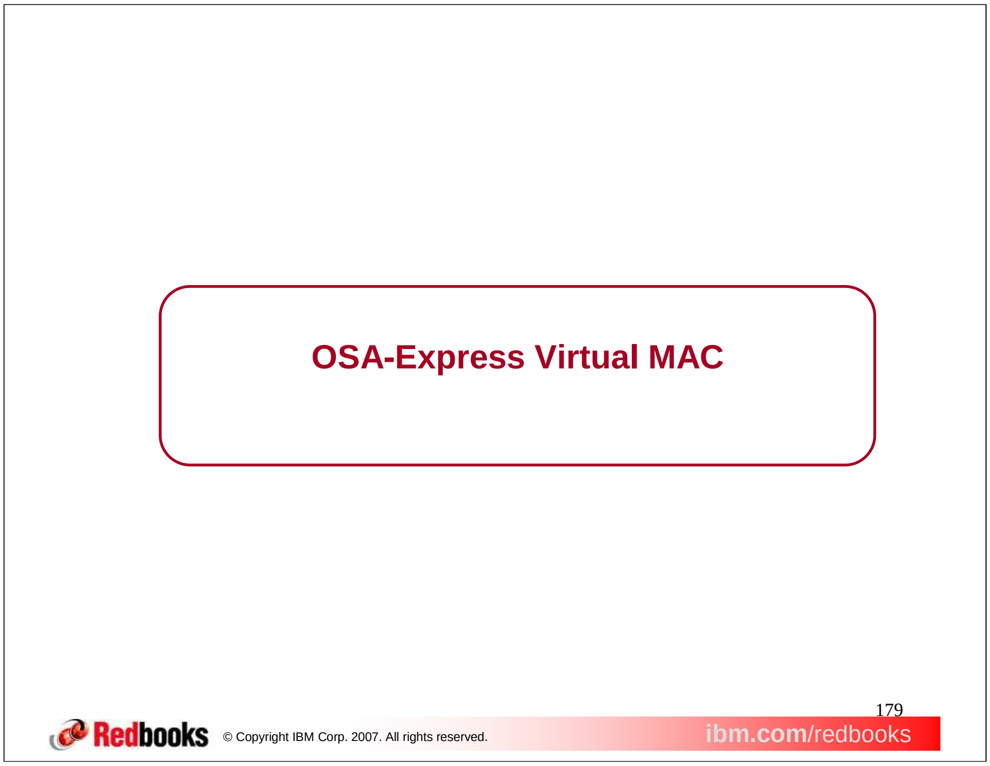
```
SL DEL,ID=MEZ2,END
SL SET,IF,ID=MEZ2,RA=(address),A=(STOPGTF,RECOVERY,SVCD),
MATCHLIM=1,JOBLIST=(TCP*,NET*),
DSPNAME=('TCP*'.*,01.CSM*,'NET*'.IST*),
SDATA=(RGN,ALLNUC,CSA,LSQA,PSA,SQA,SUM,SWA,TRT,LPA),
END
```

178

Redbooks ibm.com/redbooks

Unlike message SLIPs, only 1 PER SLIP can be active at any time (which restricts its usefulness). Also, using a PER SLIP trap can have a significant adverse effect on performance.

This is not intended to be used with the MPF ParmLib member but instead by itself.

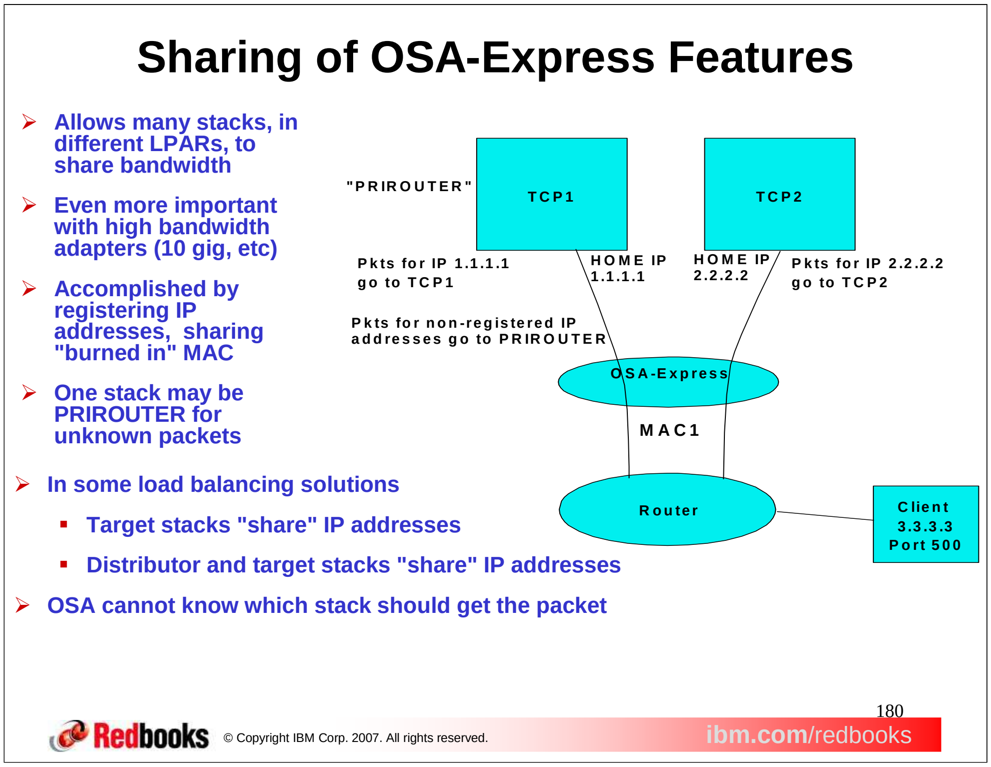


OSA-Express Virtual MAC

179

ibm.com/redbooks

This section describes the new OSA-Express virtual MAC function.



Sharing of OSA-Express Features

- Allows many stacks, in different LPARs, to share bandwidth
- Even more important with high bandwidth adapters (10 gig, etc)
- Accomplished by registering IP addresses, sharing "burned in" MAC
- One stack may be PRIROUTER for unknown packets
- In some load balancing solutions
  - Target stacks "share" IP addresses
  - Distributor and target stacks "share" IP addresses
- OSA cannot know which stack should get the packet

"PRIROUTER"

TCP1

TCP2

Pkts for IP 1.1.1.1 go to TCP1

HOME IP 1.1.1.1

HOME IP 2.2.2.2

Pkts for IP 2.2.2.2 go to TCP2

Pkts for non-registered IP addresses go to PRIROUTER

OSA-Express

MAC1

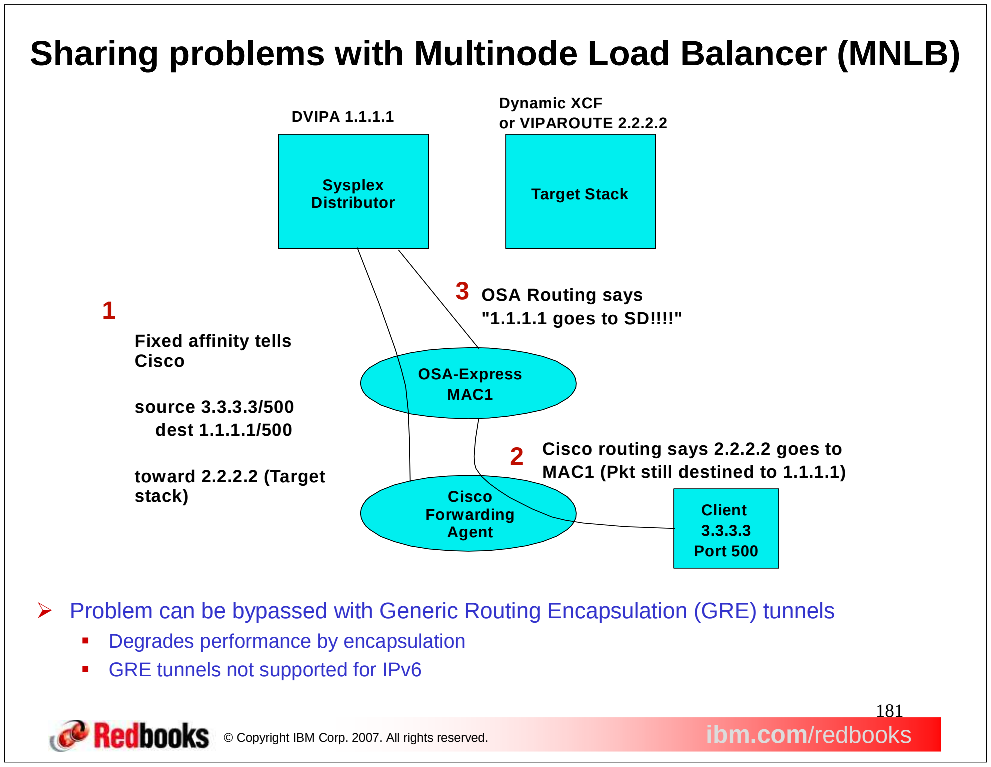
Router

Client 3.3.3.3 Port 500

With high bandwidth adapters, one stack on one LPAR usually does not send or receive enough traffic to fully utilize all the bandwidth of the OSA. To get the money for your investment, you want multiple stacks on multiple LPARs using, or sharing, the same OSA.

The figure on the right is an example of how sharing works today. In this example, stack 1 registers its home IP addresses for this OSA, such as 1.1.1.1, while stack 2 registers its home IP addresses, including 2.2.2.2. The OSA ARPs all these addresses using its one physical burned in MAC, so everyone on the LAN knows to get to any of these addresses, use that MAC. Then the OSA routes to the correct stack using the IP address. It knows that everything to 1.1.1.1 goes to TCP1, and everything to 2.2.2.2 goes to TCP2.

The stack that is PRIROUTER will get any packet sent to an IP address that is not registered in the OSA.



Sharing problems with Multinode Load Balancer (MNLB)

DVIPA 1.1.1.1

Dynamic XCF
or VIPAROUTE 2.2.2.2

Sysplex Distributor

Target Stack

1

Fixed affinity tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2 (Target stack)

3 OSA Routing says "1.1.1.1 goes to SD!!!!"

OSA-Express MAC1

2 Cisco routing says 2.2.2.2 goes to MAC1 (Pkt still destined to 1.1.1.1)

Cisco Forwarding Agent

Client
3.3.3.3
Port 500

- Problem can be bypassed with Generic Routing Encapsulation (GRE) tunnels
  - Degrades performance by encapsulation
  - GRE tunnels not supported for IPv6

181

MNLB is Multinode Load Balancer. In Multinode Load Balancing, the Sysplex Distributor registers load balancing information with the routers acting as forwarding agents. In particular, it registers a 4-tuple connection to the Cisco router, and which target IP address the Cisco should use to find the target stack MAC to send all data for that connection.

The problem is that the Cisco forwarding agents use the same MAC address for packets destined to the Distributor stack or the Target stack. So the OSA has know way to know that some packets destined to the DVIPA should go to the distributor because there is no current affinity to a target stack, and some should go to the target stack because an affinity is already established.

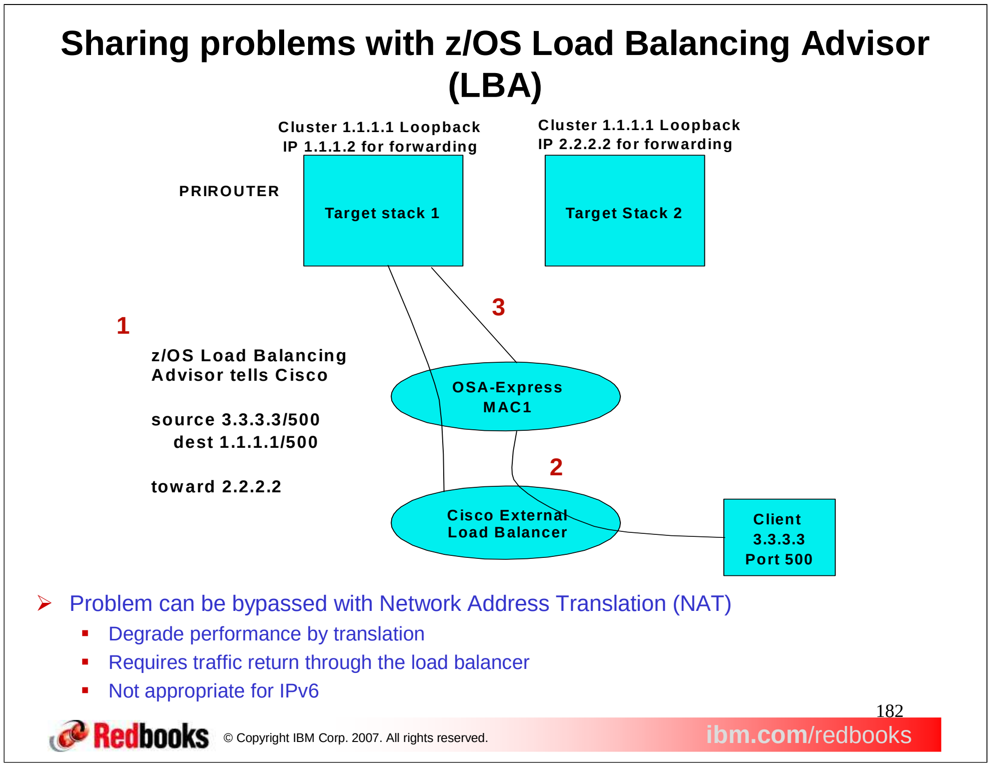
In an MNLB configuration, TCP/IP's Sysplex Distributor informs the Cisco Forwarding Agents that a given 4-tuple should be sent "toward" a particular target stack IP address - either the dynamic XCF address or the VIPAROUTE address of the target stack. The Sysplex Distributor does this by knowing which target stack it assigned to this connection. In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to DVIPA 1.1.1.1, port 500, toward destination IP address 2.2.2.2.

Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address, but does not change the contents of this packet. In this example, the Cisco would see target destination IP address 2.2.2.2 can be reached through MAC1. So it would send the packet to MAC1, but the packet still has the original 4-tuple, and in particular, the DVIPA1 address as the destination IP address.

Since OSA gets the packet with the original 4-tuple, it sees the destination IP address as DVIPA1. Since the Distributor stack has registered DVIPA1, the OSA will forward the packet to the distributor, even though the Cisco intended the packet to go directly to the target stack.

The conclusion is that because the 2 target stacks share the same MAC, with MNLB there is no way for the OSA to know which packets are truly destined to the distributor and which should go to the target stack.

This problem can be solved by using GRE tunnels. For MNLB, GRE tunnels are configured from the Cisco Forwarding Agents to the target stacks. Thus, Cisco will imbed the packets for a given 4-tuple in another packet (GRE encapsulated packet) with the destination IP address of the target stack. GRE tunnels, however, are not architected in IPv6.



Sharing problems with z/OS Load Balancing Advisor (LBA)

Cluster 1.1.1.1 Loopback IP 1.1.1.2 for forwarding

Cluster 1.1.1.1 Loopback IP 2.2.2.2 for forwarding

PRIROUTER

Target stack 1

Target Stack 2

3

1

z/OS Load Balancing Advisor tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2

OSA-Express MAC1

2

Cisco External Load Balancer

Client 3.3.3.3 Port 500

- Problem can be bypassed with Network Address Translation (NAT)
  - Degrade performance by translation
  - Requires traffic return through the load balancer
  - Not appropriate for IPv6

z/OS LBA is Communication Server’s load balancing advisor. In LBA, the Load Balancing Advisor registers the load balancing information with the routers acting as external load balancers, such as a Cisco Content Switching Module (CSM). In particular, in dispatch mode, the Load Balancing Advisor registers a 4-tuple connection to the Cisco external load balancer, and which target IP address the Cisco should use to find the target stack MAC to send all data for that connection.

The problem is that the Cisco external load balancer configured in dispatch mode uses the same MAC address for packets destined to the two target stacks. So the OSA has know way to know that some packets destined to the cluster IP address 1.1.1.1 already have an affinity to target stack 1 and should go there, and some have an affinity with target stack to and should go there.

z/OS Load Balancing Advisor is subject to the same problems with shared OSAs as MNLB. This is particularly true when the external load balancer is in dispatch mode. In dispatch mode, as with MNLB, the external load balancer does not alter the packet, but forwards the packet to the MAC of a destination IP address. In a shared OSA configuration, the following problem can occur.

TCP/IP's z/OS Load Balancing Advisor informs the Cisco external load balancer that a given 4-tuple should be sent "toward" a particular target stack IP address. That IP address is the IP address of a target stack it learned from the z/OS Load Balancing agent. In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to cluster IP address 1.1.1.1, port 500, toward destination IP address 2.2.2.2

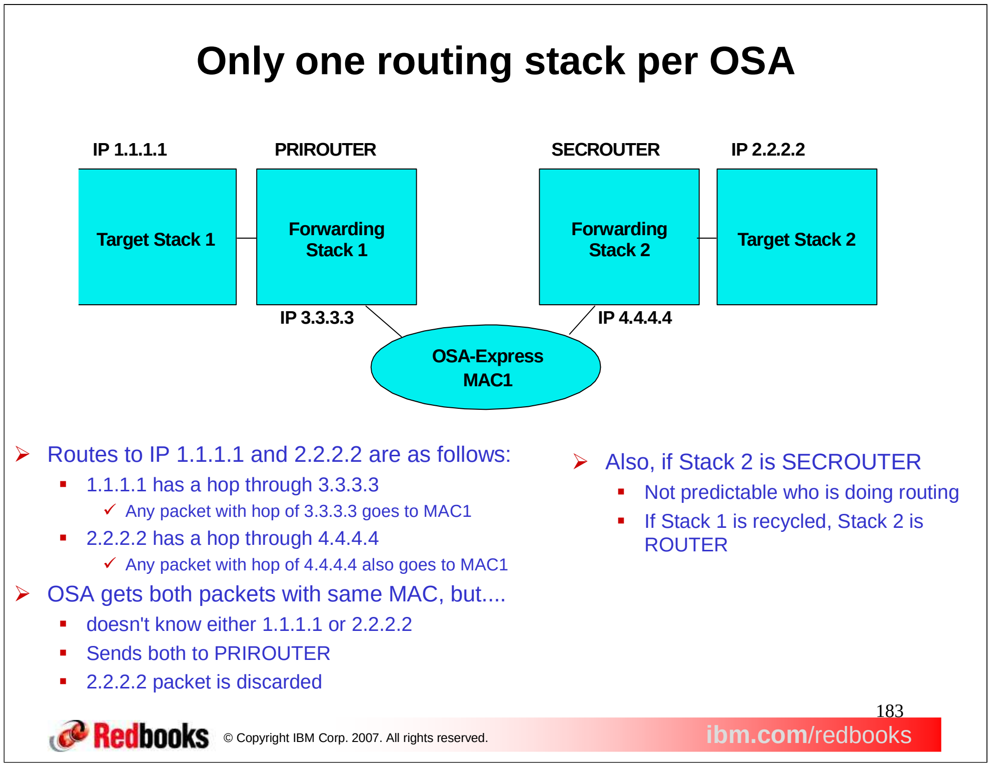
Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address, but does not change the contents of this packet. In this example, the Cisco would see target destination IP address 2.2.2.2 can be reached through MAC1. So it would send the packet to MAC1, but the packet still has the original 4-tuple, and in particular, the cluster IP address as the destination IP address.

Since OSA gets the packet with the original 4-tuple, it sees the destination IP address as a cluster IP address that has not been registered. The OSA will therefore forward the packet to the stack defined as the PRIROUTER, even though the Cisco intended the packet to go directly to target stack 2.

The conclusion is that because the 2 target stacks share the same MAC, with LBA in dispatch mode there is no way for the OSA to know to which target stack the packet is truly destined. For this reason, any customers sharing OSAs and running LBA usually use directed mode. This has the overhead of doing Network Address Translation.

The problem can be solved by using load balancing advisors configured for directed mode. For z/OS LBA, either OSAs are not shared, or the external load balancer is configured in directed mode. In directed mode, the destination IP addresses are converted using Network Address Translation (NAT) to IP addresses that belong to the given target stack. The bypass of using GRE tunnels is not supported in Cisco CSM, and thus cannot be used for LBA. Even if an external load balancer did support GRE tunnels for IPv4, GRE tunnels are not architected in IPv6.

Network Address Translation (NAT) can be used for LBA by defining the external load balancer in directed mode. This allows the external load balancer to actually change the destination IP address from the cluster IP address to an IP address on the correct target stack. However, NAT requires that the return traffic go through the external load balancer, which is not always the best route for the return traffic, and burdens the router acting as the external load balancer. With NAT, there is also an overhead associated with changing the IP address of every packet. And one of the advantages of going to IPv6 is that NAT is not required, like it sometimes is when IPv4 addresses are exhausted in a given customer shop. When LBA and IPv6 are used together, however, we again require NAT with shared OSAs.



Only one routing stack per OSA

IP 1.1.1.1
PRIROUTER
SECROUTER
IP 2.2.2.2
Target Stack 1
Forwarding Stack 1
Forwarding Stack 2
Target Stack 2
IP 3.3.3.3
IP 4.4.4.4
OSA-Express MAC1

- ➢ Routes to IP 1.1.1.1 and 2.2.2.2 are as follows:
  - 1.1.1.1 has a hop through 3.3.3.3
    - ✓ Any packet with hop of 3.3.3.3 goes to MAC1
  - 2.2.2.2 has a hop through 4.4.4.4
    - ✓ Any packet with hop of 4.4.4.4 also goes to MAC1
- ➢ OSA gets both packets with same MAC, but....
  - doesn't know either 1.1.1.1 or 2.2.2.2
  - Sends both to PRIROUTER
  - 2.2.2.2 packet is discarded

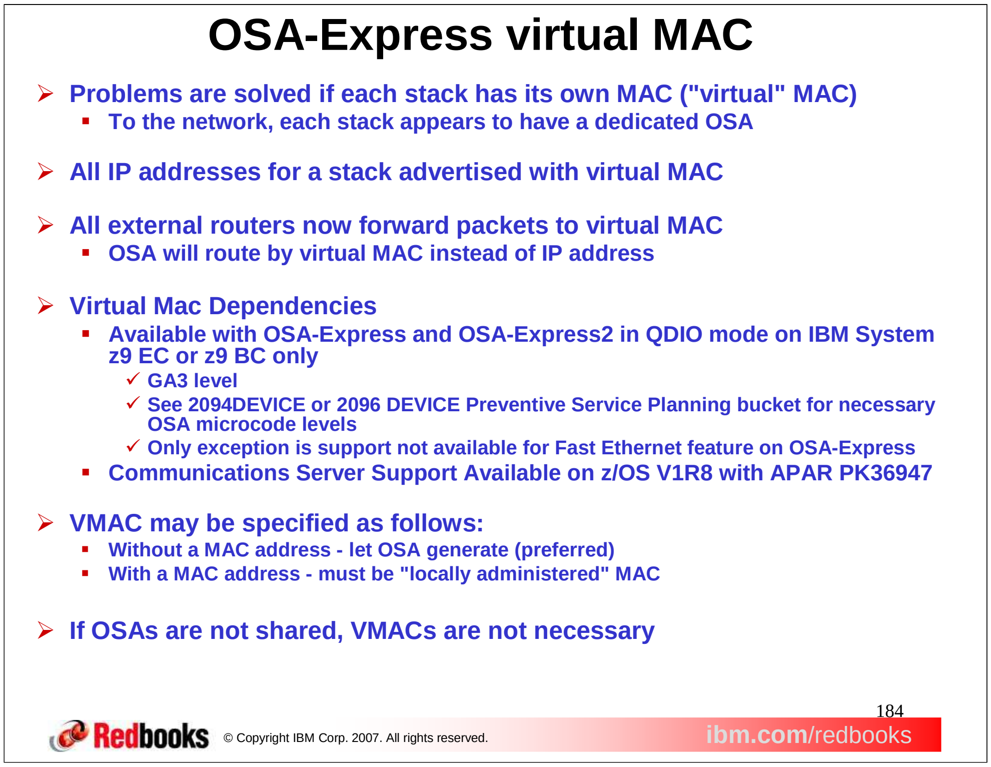
- ➢ Also, if Stack 2 is SECROUTER
  - Not predictable who is doing routing
  - If Stack 1 is recycled, Stack 2 is ROUTER

183

Redbooks ibm.com/redbooks

Though not related to load balancing, there is another problem with OSAs shared by multiple stacks. This is if two stacks sharing an OSA both act as routing stacks.

Note that in this example the OSA does not have either the address of target stack 1 or the address of target stack 2 registered, because neither stack is directly connected to it. Therefore, the OSA forwards packets destined to either target stack to the PRIROUTER stack called forwarding stack 1. Because of this, packets destined to target stack 2 will never reach target stack 2.



OSA-Express virtual MAC

- Problems are solved if each stack has its own MAC ("virtual" MAC)
  - To the network, each stack appears to have a dedicated OSA
- All IP addresses for a stack advertised with virtual MAC
- All external routers now forward packets to virtual MAC
  - OSA will route by virtual MAC instead of IP address
- Virtual Mac Dependencies
  - Available with OSA-Express and OSA-Express2 in QDIO mode on IBM System z9 EC or z9 BC only
    - ✓ GA3 level
    - ✓ See 2094DEVICE or 2096 DEVICE Preventive Service Planning bucket for necessary OSA microcode levels
    - ✓ Only exception is support not available for Fast Ethernet feature on OSA-Express
  - Communications Server Support Available on z/OS V1R8 with APAR PK36947
- VMAC may be specified as follows:
  - Without a MAC address - let OSA generate (preferred)
  - With a MAC address - must be "locally administered" MAC
- If OSAs are not shared, VMACs are not necessary

184

Redbooks ibm.com/redbooks

All these problems are resolved if each stack has its own MAC address. All IP addresses for a stack are advertised with its virtual MAC by OSA using ARP for IPv4 and by the stack using Neighbor Discovery (ND) for IPv6. The VMACs for IPv4 are defined on the LINK statement representing the OSA-Express. The VMACs for IPv6 are defined on the INTERFACE statement representing the OSA-Express. You can specify a VMAC on the LINK statement for IPv4, and use the same VMAC or a different VMAC on the INTERFACE statement for IPv6. You can also specify a VMAC on one statement (LINK or INTERFACE), and not on the other, thus using a VMAC for one protocol and the physical MAC for the other. Also, one stack can use a VMAC for its connection to the OSA, and another stack can use the physical MAC. VMACs may be assigned to a particular VLAN id, just like the physical MAC could.

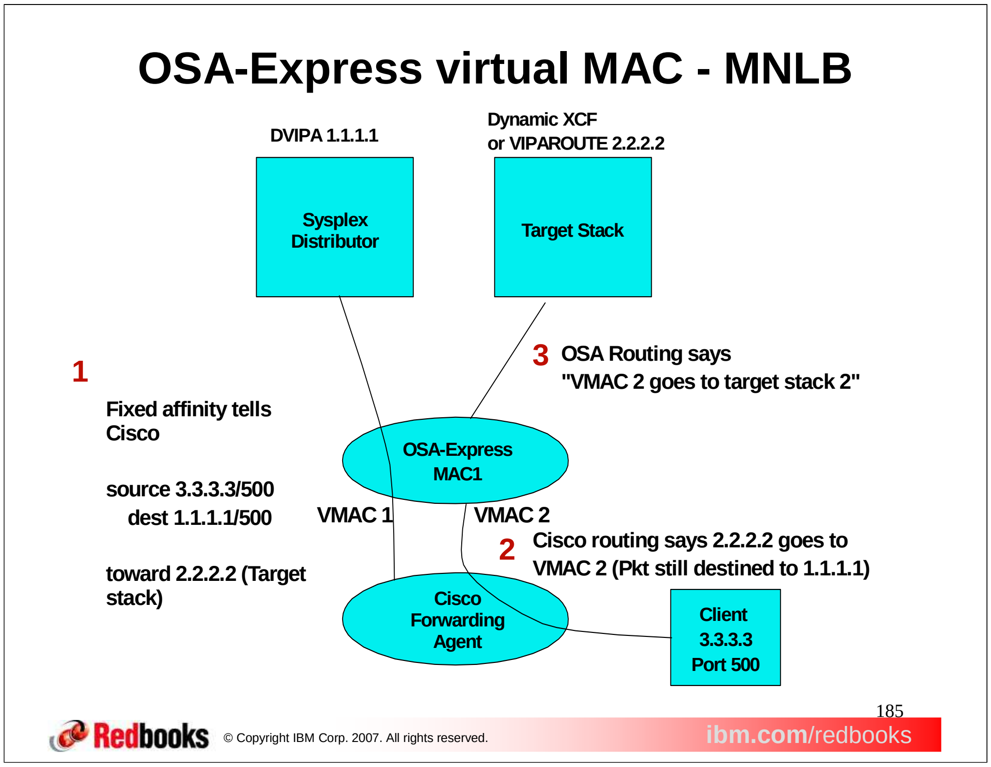
This virtual MAC function is only available on a z9 at the GA3 level, but both the OSA-Express and OSA-Express2 platforms have the virtual MAC function available. Note that the function and the publications were available in z/OS V1R8, but until GA3 of the z9 EC and BC, the OSA code was not available. The virtual MAC function is available with Communications Server V1R8 when APAR PK36947 is applied.

When VMAC is specified without a MAC address, the OSA will generate one. OSA's VMAC generation scheme, to guarantee uniqueness, is as follows: First byte of VMAC will be a constant 02. The 2 bit indicates this is a locally administered MAC address. This will guarantee it is unique from all physical "burned-in" MACs, since the 2 bit for these adapters is off, indicating they are "universal" addresses. The last 3 bytes will be the last 3 bytes of the physical MAC address. This will guarantee all VMACs on one OSA will be unique from all other VMACs on any other OSA. To guarantee stacks sharing an OSA will get unique addresses, the second and third bytes of the VMAC will be an instance count, incremented each time OSA gives out a VMAC address.

OSA will generate a different VMAC for IPv6 versus IPv4. The same generation rules apply as applied to the LINK statement. The VMAC generated for the INTERFACE statement should differ from that generated for the DEVICE/LINK statement only by the instance count. If INTFID is not defined by the user, on the IPv6 INTERFACE statement, the generated INTFID is different for VMAC versus non-VMAC INTERFACES. For non-VMAC INTERFACES, it is first 3 bytes of MAC, followed by OSA generated instance count, followed by last 3 bytes of MAC. This is because the MAC is shared by multiple INTERFACEs, and the INTFID needs to be unique. For VMAC INTERFACEs, the VMAC is unique for each stack. So the standard form of an interface id is generated. This is the first 3 bytes of the VMAC, followed by X'FFFE', followed by the last 3 bytes of the VMAC.

TCP/IP will reuse the same generated VMAC address when a device becomes inactive and is reactivated. A new VMAC address will be generated for a given OSA if the stack is stopped and restarted. If the VMAC is defined by the user, it must be a 12 digit hexadecimal number, with the **X'02'** bit in the first byte of the VMAC on, indicating this is a locally administered MAC address. It is up to the user to ensure the uniqueness of the VMAC on the local LAN on which this OSA resides. It is recommended VMACs be used anytime the OSA is shared.

When ROUTEALL is specified or defaulted on the LINK or INTERFACE statement, then all packets destined for the VMAC are routed to this stack. This is done even if the IP address is not registered. When ROUTELCL is specified on the LINK or INTERFACE statement that only packets for registered IP addresses will be routed to this stack. This parameter



OSA-Express virtual MAC - MNLB

DVIPA 1.1.1.1

Sysplex Distributor

Dynamic XCF
or VIPAROUTE 2.2.2.2

Target Stack

1

Fixed affinity tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2 (Target stack)

3 OSA Routing says "VMAC 2 goes to target stack 2"

OSA-Express MAC1

VMAC 1

VMAC 2

2 Cisco routing says 2.2.2.2 goes to VMAC 2 (Pkt still destined to 1.1.1.1)

Cisco Forwarding Agent

Client
3.3.3.3
Port 500

185

Redbooks

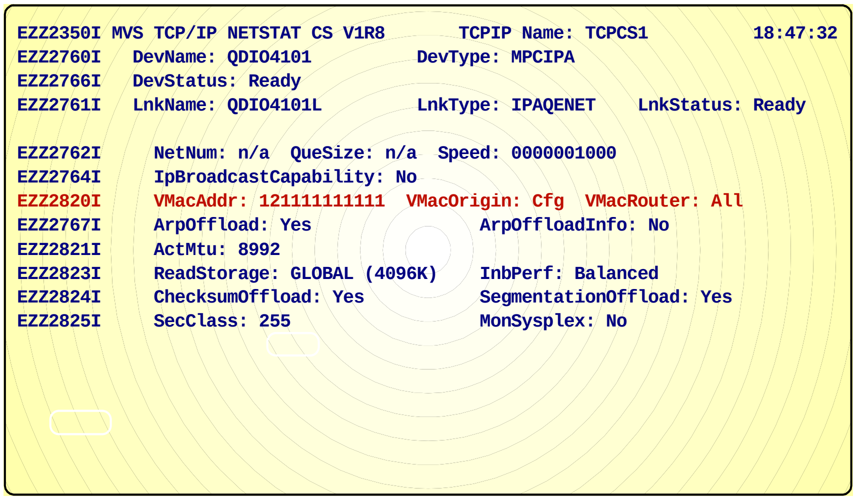
ibm.com/redbooks

Let’s see how VMAC fixes the MNLB problems. In an MNLB configuration with VMACs, TCP/IP's Sysplex Distributor informs the Cisco Forwarding Agents that a given 4-tuple should be sent "toward" a particular target stack IP address - either the dynamic XCF address or the VIPAROUTE address of the target stack. The Sysplex Distributor does this by knowing which target stack it assigned to this connection. In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to DVIPA 1.1.1.1, port 500, toward destination IP address 2.2.2.2.

Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address. However, with VMAC in place, the OSA has advertised that destination IP address 2.2.2.2 can be reached not through shared MAC1, but through unique VMAC2. So it would send the packet to VMAC2, still with the original 4-tuple, and in particular, the DVIPA1 address as the destination IP address.

OSA gets the packet with the original 4-tuple, but the destination is not shared MAC1, but VMAC2. Because the packet is to a VMAC, it will route the packet directly to the target stack owning that VMAC, even though the destination IP address of DVIPA1 is registered to the distributor stack.
VMAC provides these same advantages for z/OS Load Balancing Advisor solutions, and in some configurations allows for using dispatch mode instead of directed mode for the external load balancer. Dispatch mode will still be subject to some special considerations when the load balancer is more than one hop away from the target systems. See the IP Configuration Guide, section "External IP workload balancing solutions," for more details on these considerations.

Netstat DEVLINKS/-d



```
EZZ2350I MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPCS1          18:47:32
EZZ2760I   DevName: QDIO4101          DevType: MPCIPA
EZZ2766I   DevStatus: Ready
EZZ2761I   LnkName: QDIO4101L         LnkType: IPAQENET    LnkStatus: Ready

EZZ2762I     NetNum: n/a  QueSize: n/a  Speed: 0000001000
EZZ2764I     IpBroadcastCapability: No
EZZ2820I     VMacAddr: 121111111111  VMacOrigin: Cfg  VMacRouter: All
EZZ2767I     ArpOffload: Yes                ArpOffloadInfo: No
EZZ2821I     ActMtu: 8992
EZZ2823I     ReadStorage: GLOBAL (4096K)    InbPerf: Balanced
EZZ2824I     ChecksumOffload: Yes           SegmentationOffload: Yes
EZZ2825I     SecClass: 255                  MonSysplex: No
```

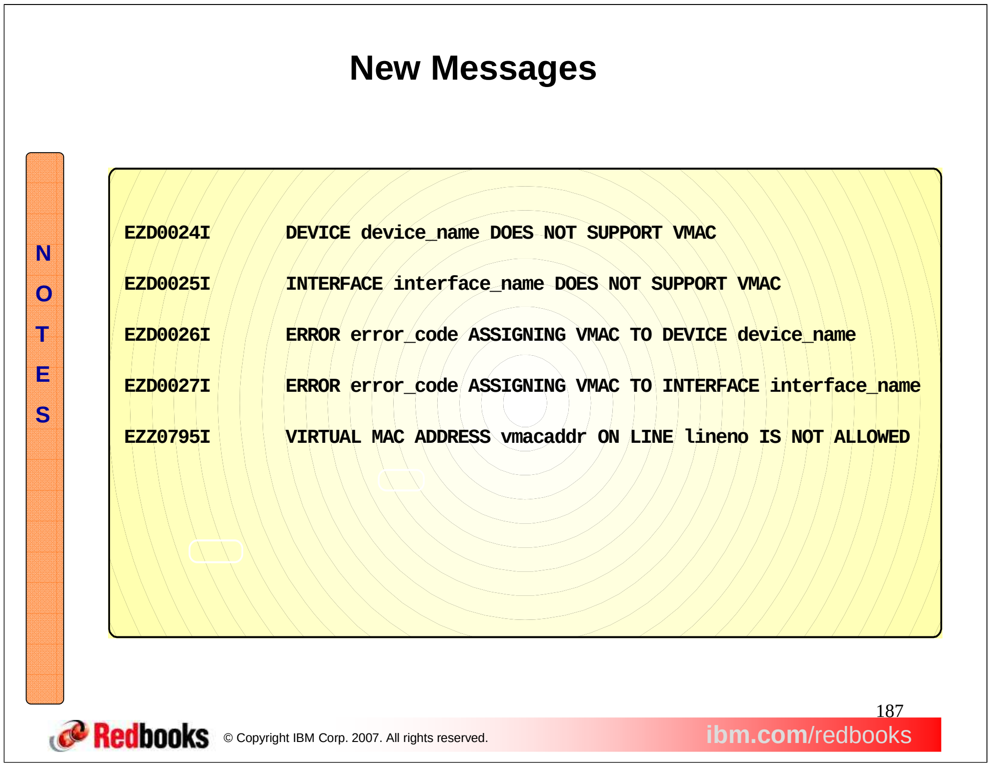
Redbooks

The VMAC fields are displayed on the netstat devlinks command.

The VMacAddr field will be either the predefined VMAC address if VMAC xxxxxxxx was defined, or the OSA generated VMAC address if VMAC was defined without a specific MAC address.

The VMacOrigin field will be Cfg if the MAC address is defined by the user, or OSA if the MAC address is generated by the OSA.

The VMacRouter field will be ALL if ROUTEALL was specified or defaulted for the VMAC, or LCL if ROUTELCL was specified for this VMAC.



New Messages

N
O
T
E
S

EZD0024I	DEVICE device_name DOES NOT SUPPORT VMAC
EZD0025I	INTERFACE interface_name DOES NOT SUPPORT VMAC
EZD0026I	ERROR error_code ASSIGNING VMAC TO DEVICE device_name
EZD0027I	ERROR error_code ASSIGNING VMAC TO INTERFACE interface_name
EZZ0795I	VIRTUAL MAC ADDRESS vmacaddr ON LINE lineno IS NOT ALLOWED

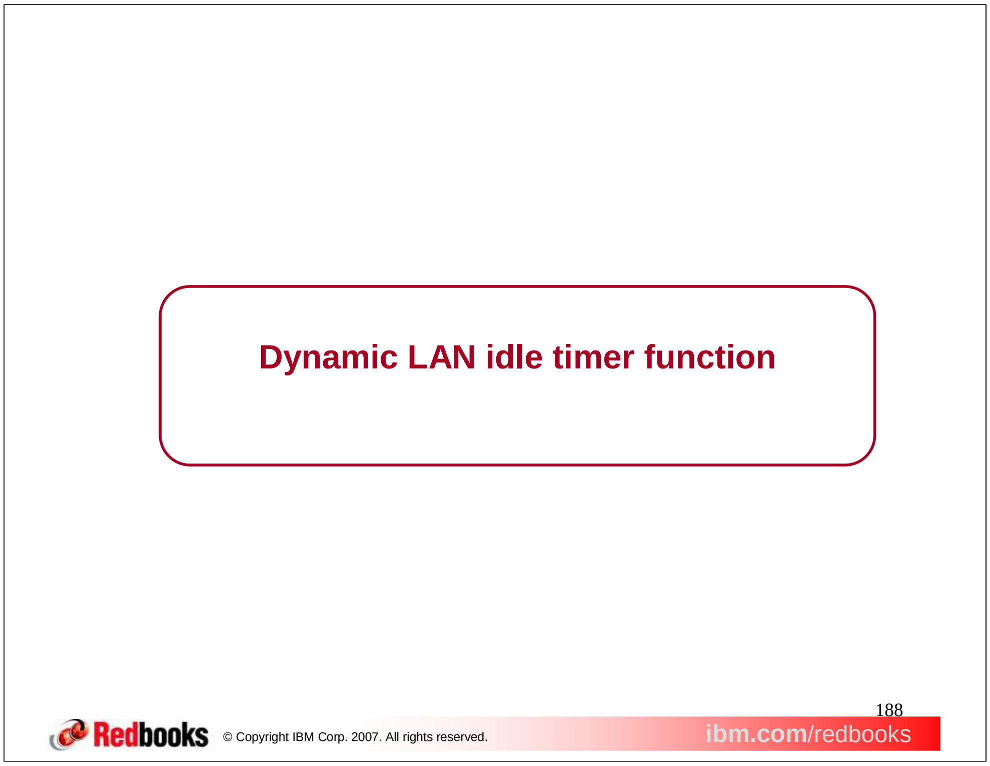
Redbooks

These are the new messages introduced with this function.

Note that if VMAC is defined, and either the OSA does not support VMAC, or an error was reported attempting to assign the VMAC, Device or Interface activation fails. This is because it is assumed if VMAC was configured, other configurations were altered to use it. Those altered configurations will likely fail without VMAC. For example, the external load balancer may have been reconfigured to use dispatch mode, or GRE tunnels may have been removed from Cisco forwarding agents for MNLB. In either case, load balancing will now fail.

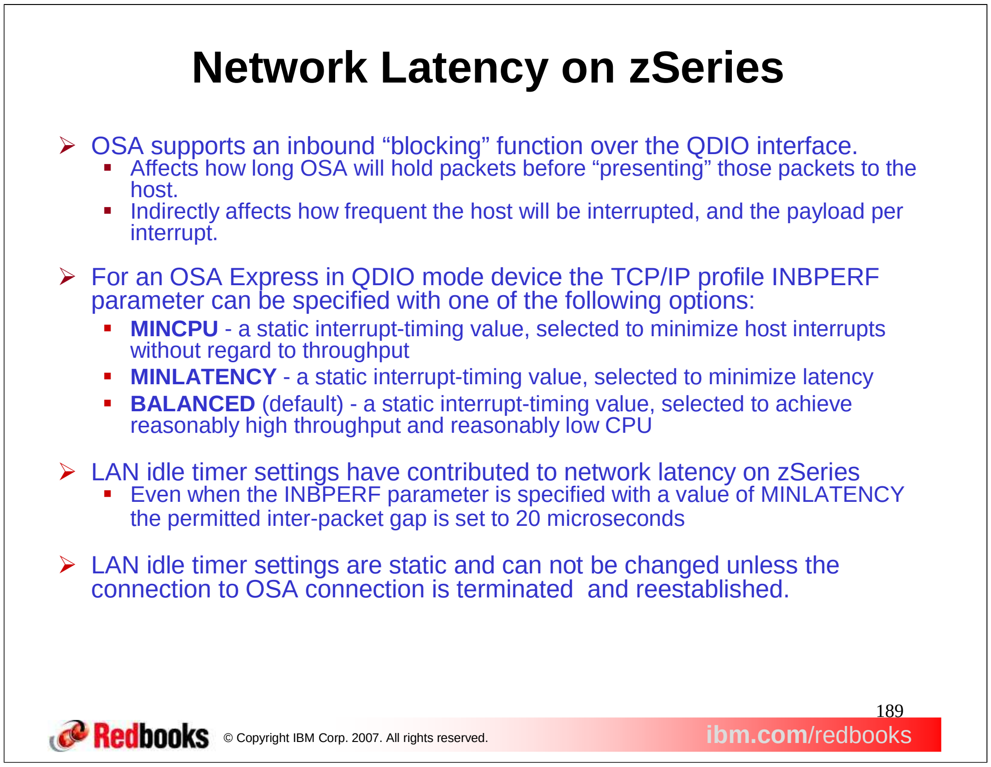
Messages EZD0026I and EZD0027I are expected to be issued with only one code - the code indicating the VMAC attempting to be assigned was already defined.

Message EZZ0795I is issued when the user attempts to configure a virtual MAC address with the local/universal bit set off, meaning universal MAC address. Only locally administered MAC addresses should be defined for VMAC. Device or interface configuration processing fails, for the same reasons stated with messages EZD0024I-EZD0027I.



Dynamic LAN idle timer function

This section describes enhancements to the setting of OSA LAN idle timer settings for Communications Server in V1R9.



Network Latency on zSeries

- OSA supports an inbound “blocking” function over the QDIO interface.
  - Affects how long OSA will hold packets before “presenting” those packets to the host.
  - Indirectly affects how frequent the host will be interrupted, and the payload per interrupt.
- For an OSA Express in QDIO mode device the TCP/IP profile INBPERF parameter can be specified with one of the following options:
  - **MINCPU** - a static interrupt-timing value, selected to minimize host interrupts without regard to throughput
  - **MINLATENCY** - a static interrupt-timing value, selected to minimize latency
  - **BALANCED** (default) - a static interrupt-timing value, selected to achieve reasonably high throughput and reasonably low CPU
- LAN idle timer settings have contributed to network latency on zSeries
  - Even when the INBPERF parameter is specified with a value of MINLATENCY the permitted inter-packet gap is set to 20 microseconds
- LAN idle timer settings are static and can not be changed unless the connection to OSA connection is terminated and reestablished.

189

Redbooks ibm.com/redbooks

OSA supports an inbound “blocking” (or packing) function over the QDIO interface. This function affects how long OSA will hold packets before “presenting” those packets to the host. Here “presenting” means assigning the read buffer to the host, which is a matter of updating the state of the host buffer to host owned. In most cases this same action will result in an interrupt to the host for this QDIO data device. Therefore, this function indirectly affects the QDIO interrupt processing (i.e. how frequent the host will be interrupted, and the payload per interrupt).

This function is referred to as the OSA “LAN Idle timer”. Today the host can pass various time intervals to OSA when the QDIO data device is activated. In the z/OS case, the system administrator can adjust this setting. However, the setting is static and can not be changed unless the connection to OSA is terminated (device is stopped) and reestablished (restart the device).

Currently the user can not directly configure explicit LAN Idle settings. Instead the user provides a constant value to OSA which represented the best “compromise” setting. In the TCP/IP profile the user can define a LAN Idle setting for an OSA Express in QDIO mode device. This is performed by specifying the TCP/IP profile INBPERF parameter with one of the following options:

MINCPU setting

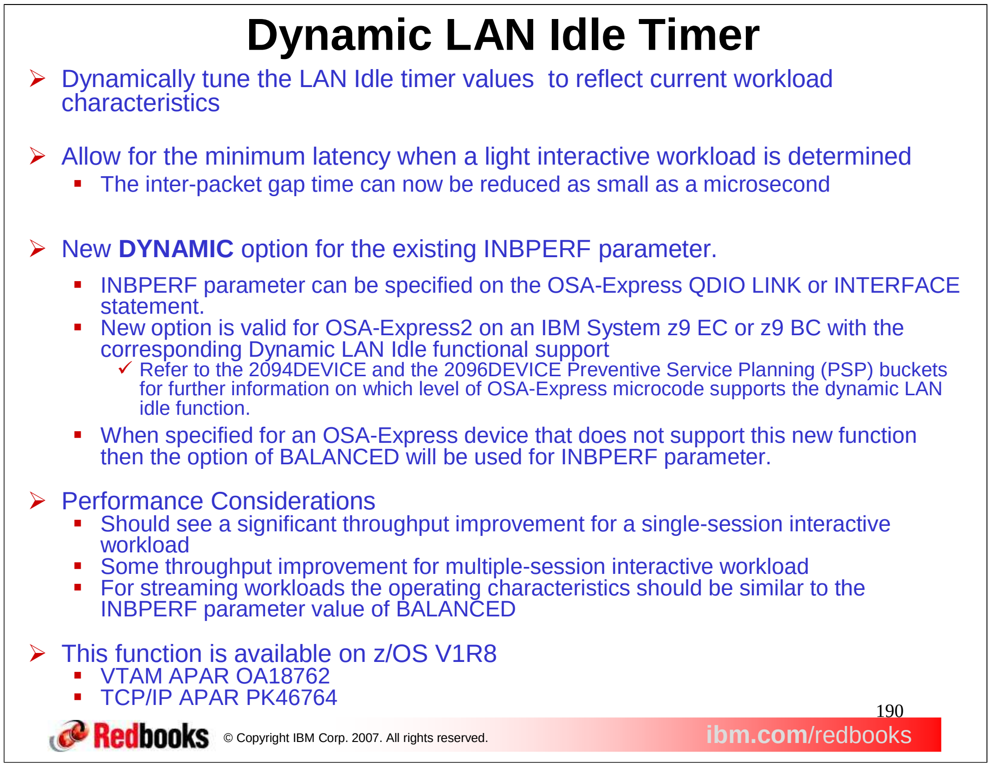
OSA should increase packet hold time. Holding packets longer minimizes CPU utilization by decreasing interrupt frequency and providing a better payload per interrupt)

MINLATENCY setting

OSA should decrease packet hold time. Presenting packets to host sooner reduce network latency (but drive up CPU utilization by causing more frequent interrupts and a smaller payload per interrupt)

BALANCED Setting (default)

CS would compromise and set timer interval values that split the difference in the above two settings Our current LAN idle timers are having OSA hold the packets to save CPU when sometimes CPU is not an issue. Note that at higher utilizations, dispatch delay becomes a determining factor in network turnaround time rather than LAN idle delay.



Dynamic LAN Idle Timer

- Dynamically tune the LAN Idle timer values to reflect current workload characteristics
- Allow for the minimum latency when a light interactive workload is determined
  - The inter-packet gap time can now be reduced as small as a microsecond
- New **DYNAMIC** option for the existing INBPERF parameter.
  - INBPERF parameter can be specified on the OSA-Express QDIO LINK or INTERFACE statement.
  - New option is valid for OSA-Express2 on an IBM System z9 EC or z9 BC with the corresponding Dynamic LAN Idle functional support
    - ✓ Refer to the 2094DEVICE and the 2096DEVICE Preventive Service Planning (PSP) buckets for further information on which level of OSA-Express microcode supports the dynamic LAN idle function.
  - When specified for an OSA-Express device that does not support this new function then the option of BALANCED will be used for INBPERF parameter.
- Performance Considerations
  - Should see a significant throughput improvement for a single-session interactive workload
  - Some throughput improvement for multiple-session interactive workload
  - For streaming workloads the operating characteristics should be similar to the INBPERF parameter value of BALANCED
- This function is available on z/OS V1R8
  - VTAM APAR OA18762
  - TCP/IP APAR PK46764

190

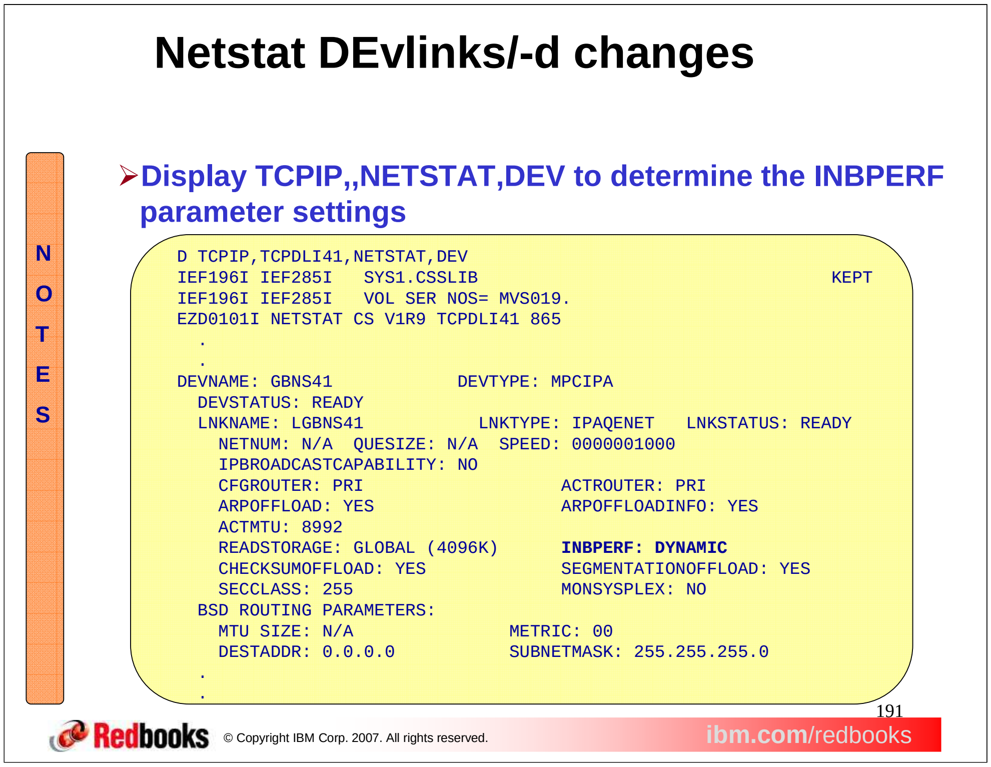
Performance studies have shown network latency improvements in environments where the CEC is under low utilization of up to 35% by tuning the Lan Idle timer within the OSA Express2 using a dynamic algorithm that takes workload characteristics. This dynamic algorithm involves taking the current default inter-packet gap of 40 microseconds to as low as 1 microsecond.

A new INBPERF parameter option of DYNAMIC will now be permitted. This new configurable setting allows the TCP/IP stack to dynamically calculate the best values for the LAN idle timer settings. These settings will indirectly determine how frequently the OSA adapter will interrupt the host for inbound traffic.

The new dynamic LAN idle algorithm will be utilized to compute the optimal OSA inter-packet gap timers to be utilized. These LAN Idle Timers will be updated by this algorithm to attempt to optimize throughput. The new algorithm should be effective for all protocols.

The new DYNAMIC option for the existing INBPERF parameter is only valid for OSA-Express2 on an IBM System z9 EC or z9 BC with the corresponding Dynamic LAN Idle functional support. You should see a significant throughput improvement for a single-session interactive workload. A Latency improvement of 30% or more with a reduction in the CPU cost per transaction.

The dynamic LAN idle timer algorithm will adjust the LAN idle timer settings to best fit the traffic characteristics.



Netstat DEvlinks/-d changes

➢Display TCPIP,,NETSTAT,DEV to determine the INBPERF parameter settings

```
D TCPIP,TCPDLI41,NETSTAT,DEV
IEF196I IEF285I   SYS1.CSSLIB                                  KEPT
IEF196I IEF285I   VOL SER NOS= MVS019.
EZD0101I NETSTAT CS V1R9 TCPDLI41 865
  .
  .
DEVNAME: GBNS41            DEVTYPE: MPCIPA
  DEVSTATUS: READY
  LNKNAME: LGBNS41           LNKTYPE: IPAQENET   LNKSTATUS: READY
    NETNUM: N/A  QUESIZE: N/A  SPEED: 0000001000
    IPBROADCASTCAPABILITY: NO
    CFGROUTER: PRI                   ACTROUTER: PRI
    ARPOFFLOAD: YES                  ARPOFFLOADINFO: YES
    ACTMTU: 8992
    READSTORAGE: GLOBAL (4096K)      INBPERF: DYNAMIC
    CHECKSUMOFFLOAD: YES             SEGMENTATIONOFFLOAD: YES
    SECCLASS: 255                    MONSYSPLEX: NO
  BSD ROUTING PARAMETERS:
    MTU SIZE: N/A               METRIC: 00
    DESTADDR: 0.0.0.0           SUBNETMASK: 255.255.255.0
  .
  .
```

191

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

The **Netstat DEvlinks/-d** command displays information about devices and defined interfaces or links defined to the TCP/IP stack.

The **INBPERF** field is significant only for active IPAQENET links, IPAQTR links, and IPAQENET6 interfaces. This field indicates how frequently the adapter should interrupt the host. The possible values are:

MinCPU

Indicates that the adapter is using a static interrupt-timing value that minimizes host interrupts, and therefore minimizes host CPU consumption.

MinLatency

Indicates that the adapter is using a static interrupt-timing value that minimizes latency delay by more aggressively presenting received packets to the host.

Balanced

Indicates that the adapter is using a static interrupt-timing value that strikes a balance between MinCPU and MinLatency.

Dynamic

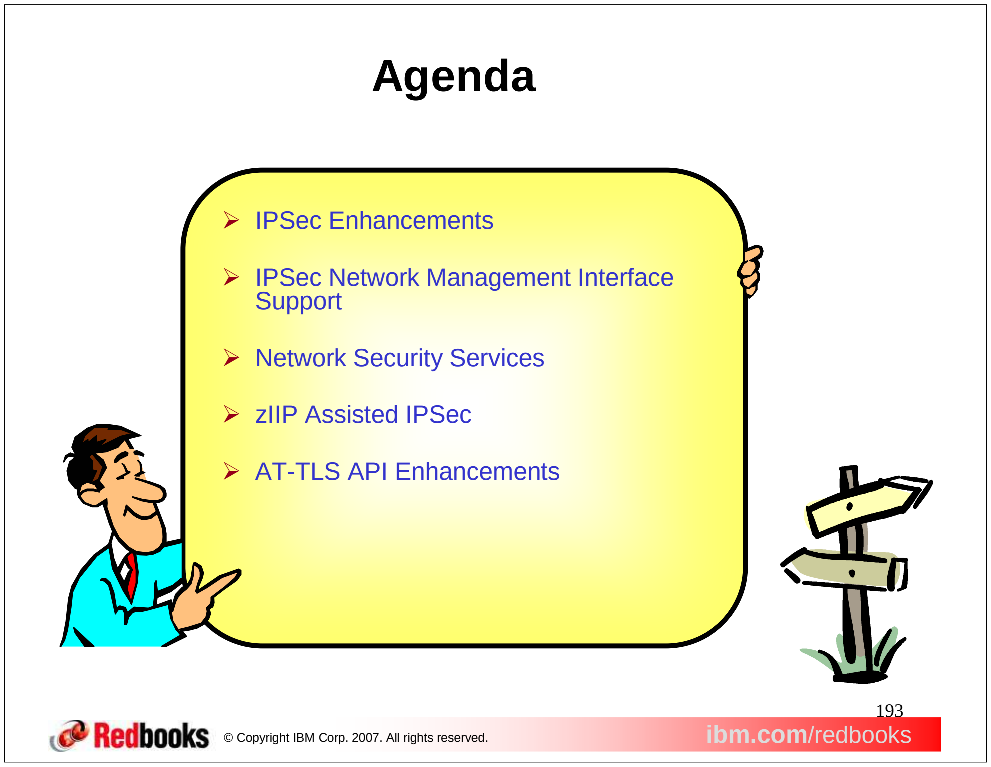
Indicates that the stack and the adapter are dynamically updating the frequently with which the adapter should interrupt the host for inbound traffic.

ibm.com

Networking Security

International Technical Support Organization

This presentation describes enhancements and new functions added to the Communications Server in the area of network security in z/OS V1R9.



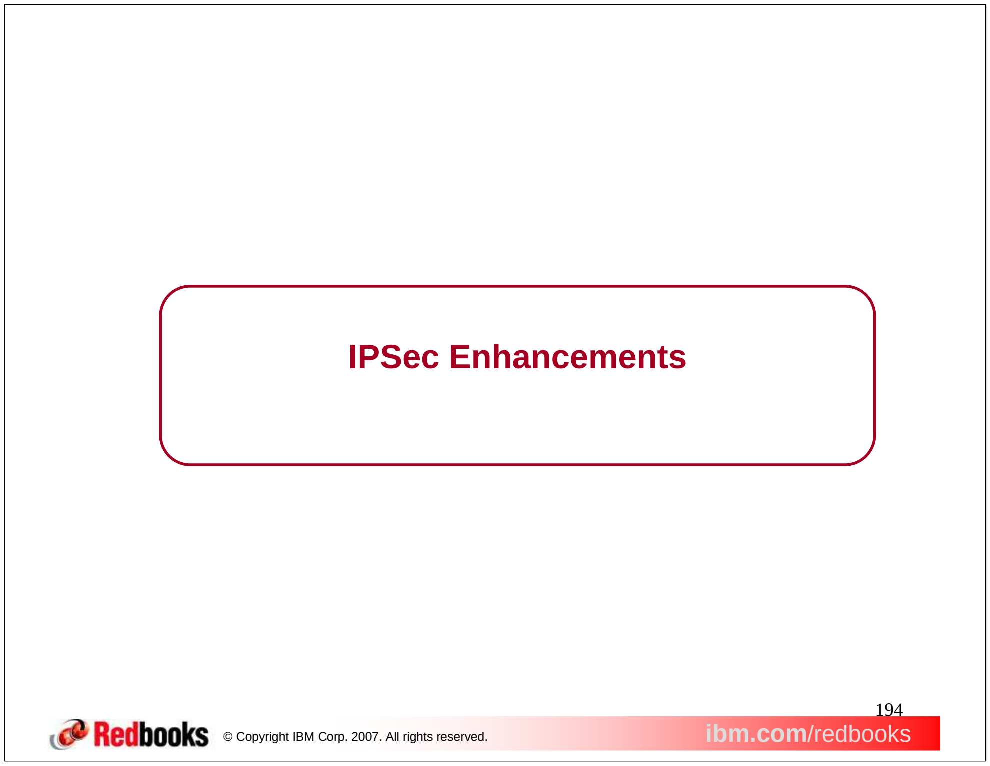
Agenda

- IPSec Enhancements
- IPSec Network Management Interface Support
- Network Security Services
- zIIP Assisted IPSec
- AT-TLS API Enhancements

There are some enhancements as well as new function added for IP Security.

Network Security Services is a new function in V1R9 that provides a set of services for IPSec.

Some enhancements were made to the existing AT-TLS API.

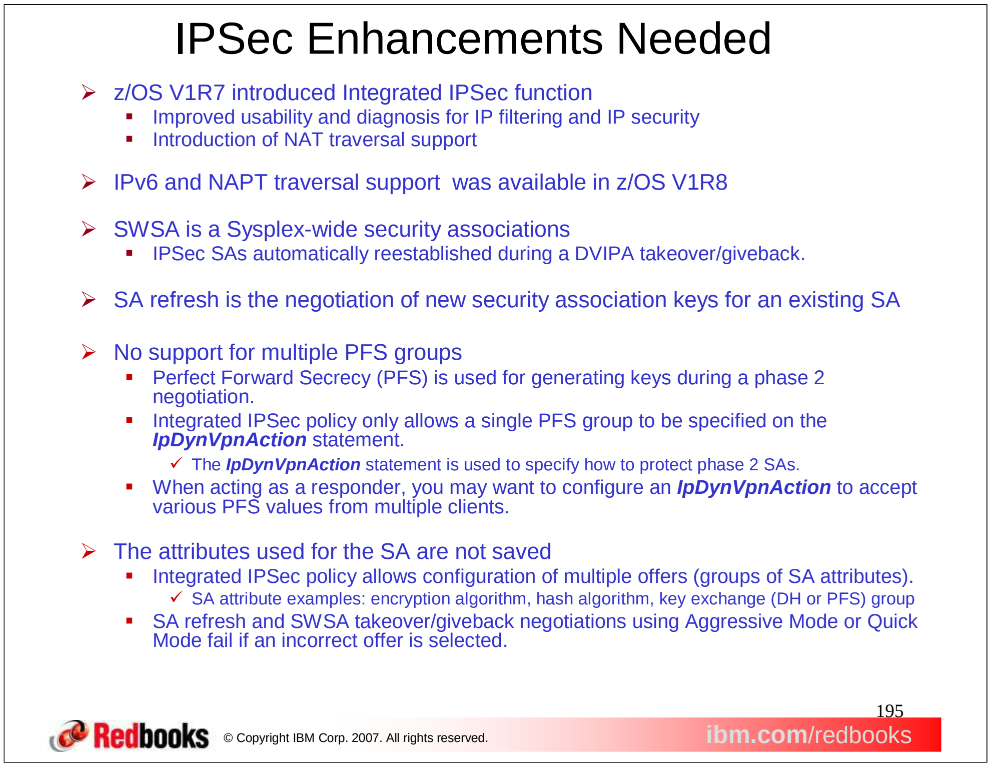


IPSec Enhancements

194

ibm.com/redbooks

This section describes the enhancements to the existing IPSec function in V1R9.



IPSec Enhancements Needed

- z/OS V1R7 introduced Integrated IPSec function
  - Improved usability and diagnosis for IP filtering and IP security
  - Introduction of NAT traversal support
- IPv6 and NAPT traversal support was available in z/OS V1R8
- SWSA is a Sysplex-wide security associations
  - IPSec SAs automatically reestablished during a DVIPA takeover/giveback.
- SA refresh is the negotiation of new security association keys for an existing SA
- No support for multiple PFS groups
  - Perfect Forward Secrecy (PFS) is used for generating keys during a phase 2 negotiation.
  - Integrated IPSec policy only allows a single PFS group to be specified on the ***IpDynVpnAction*** statement.
    - ✓ The ***IpDynVpnAction*** statement is used to specify how to protect phase 2 SAs.
  - When acting as a responder, you may want to configure an ***IpDynVpnAction*** to accept various PFS values from multiple clients.
- The attributes used for the SA are not saved
  - Integrated IPSec policy allows configuration of multiple offers (groups of SA attributes).
    - ✓ SA attribute examples: encryption algorithm, hash algorithm, key exchange (DH or PFS) group
  - SA refresh and SWSA takeover/giveback negotiations using Aggressive Mode or Quick Mode fail if an incorrect offer is selected.

Redbooks ibm.com/redbooks

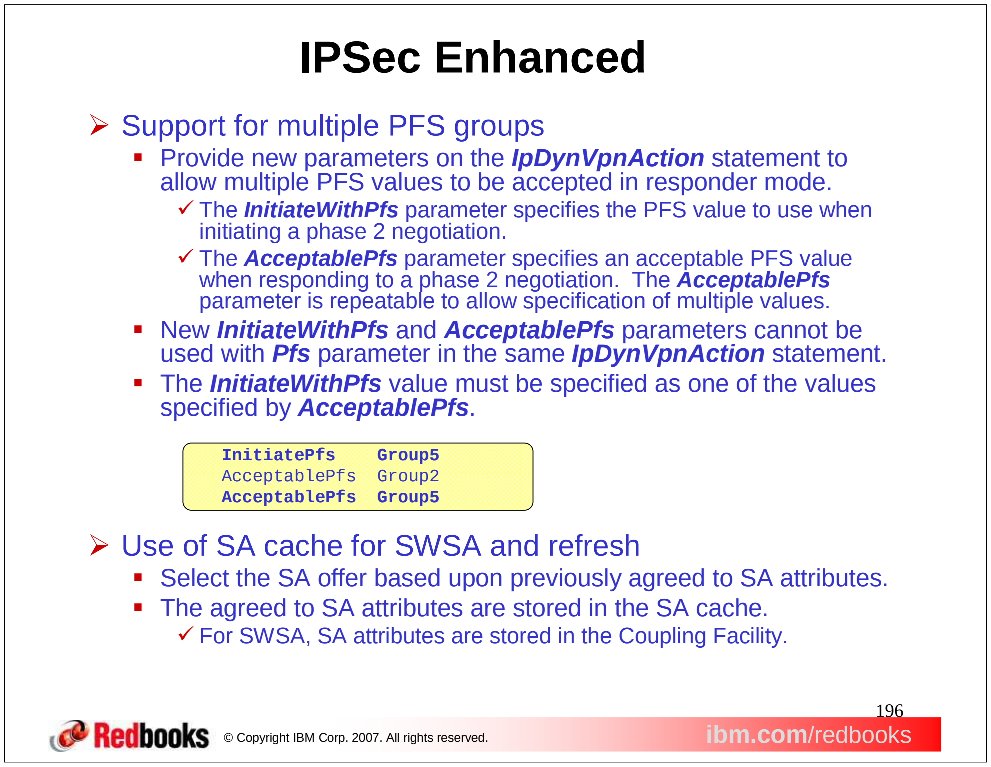
The Integrated IPSec function was introduced in CS V1R7 as a replacement for the Security Server Firewall Technologies IP filtering and IP security function. The Integrated IPSec function has improved usability over Firewall Technologies. The Integrated IPSec function has Network Address Translation (NAT) support. It also has Network Address Port Translation (NAPT) traversal support. NAPT translates multiple internal IP addresses to a single public address and translates the TCP or UDP port to make the connection unique.

The IKE daemon negotiates the dynamic IP security associations. Security associations are negotiated in two phases. Phase 1 SAs are established first and they protect the phase 2 negotiations. You have a choice of two modes: Aggressive Mode or Main Mode. Phase 2 SAs protect data traffic and uses Quick Mode.

SWSA allows for IPSec SAs to be automatically reestablished during a DVIPA takeover or giveback. SA refreshes are performed prior to expiration of the existing SA.

The usage of PFS is optional in a phase 2 negotiation. For maximum interoperability, servers should be able to accept multiple PFS values. Some clients may only be able to support lower PFS groups, while other clients may support higher PFS groups for higher security.

During a takeover/giveback or refresh using Aggressive or quick mode, the first offer is used which may not be the offer used in the establishment of the SA when multiple offers are configured. In Main mode all of the SA offers are sent to the peer and the peer can select an acceptable offer.



IPSec Enhanced

- Support for multiple PFS groups
  - Provide new parameters on the ***IpDynVpnAction*** statement to allow multiple PFS values to be accepted in responder mode.
    - ✓ The ***InitiateWithPfs*** parameter specifies the PFS value to use when initiating a phase 2 negotiation.
    - ✓ The ***AcceptablePfs*** parameter specifies an acceptable PFS value when responding to a phase 2 negotiation. The ***AcceptablePfs*** parameter is repeatable to allow specification of multiple values.
  - New ***InitiateWithPfs*** and ***AcceptablePfs*** parameters cannot be used with ***Pfs*** parameter in the same ***IpDynVpnAction*** statement.
  - The ***InitiateWithPfs*** value must be specified as one of the values specified by ***AcceptablePfs***.

```
InitiatePfs    Group5
AcceptablePfs  Group2
AcceptablePfs  Group5
```

- Use of SA cache for SWSA and refresh
  - Select the SA offer based upon previously agreed to SA attributes.
  - The agreed to SA attributes are stored in the SA cache.
    - ✓ For SWSA, SA attributes are stored in the Coupling Facility.

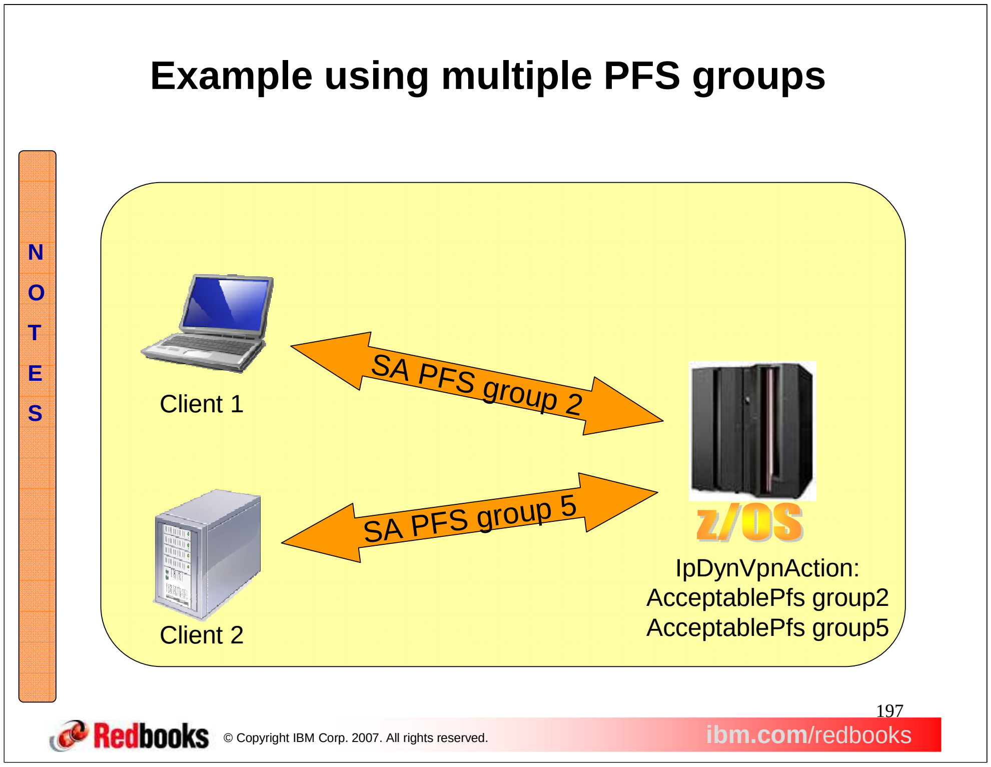
196

ibm.com/redbooks

New parameters are added to the IpDynVpnAction statement to allow the user to configure multiple PFS values to be accepted in responder mode. The InitiateWIthPfs and AcceptablePfs parameters should now be used instead of the Pfs parameter.

With a SA cache the correct offer will be used for SWSA takeover and giveback and a SA refresh. The SA cache is located in the IKED private storage.

Policy agent enforces that the *InitiateWithPfs* value must be specified as one of the values specified by *AcceptablePfs*. Otherwise, SA refreshes and SWSA takeovers would fail.



Example using multiple PFS groups

N
O
T
E
S

Client 1

SA PFS group 2

Client 2

SA PFS group 5

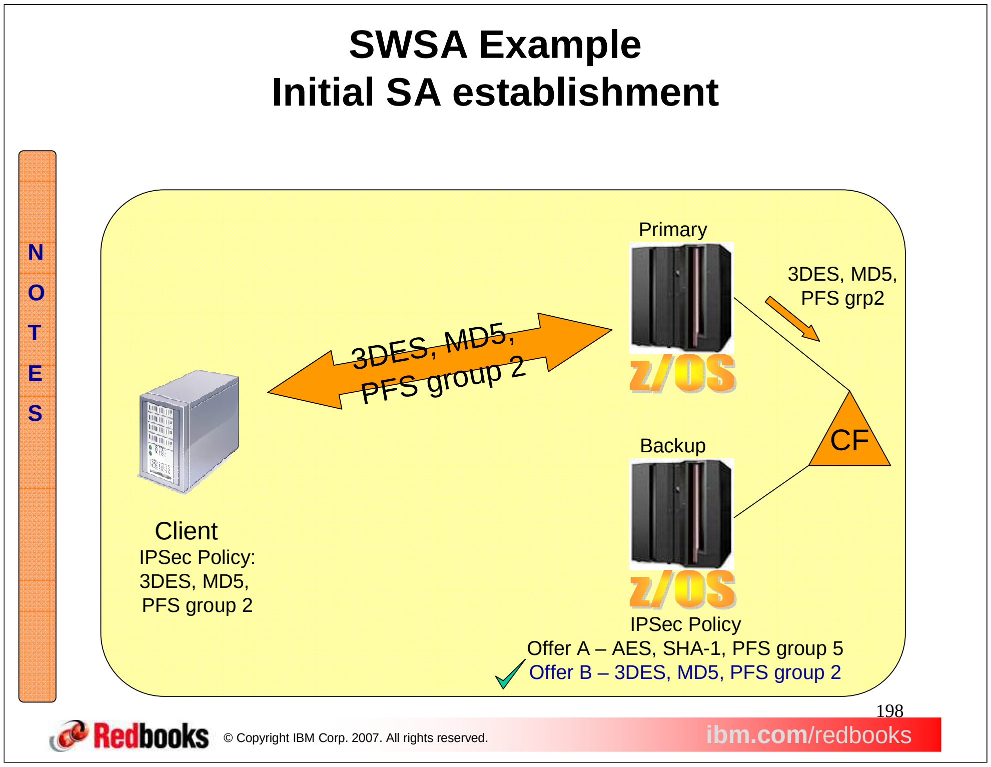
z/OS

IpDynVpnAction:
AcceptablePfs group2
AcceptablePfs group5

197

Redbooks ibm.com/redbooks

The two clients use different PFS groups when establishing a SA with the z/OS server. The z/OS server is able to respond to both clients by configuring a single IpDynVpnAction to support multiple PFS groups.



SWSA Example
Initial SA establishment

N O T E S

Primary

3DES, MD5,
PFS grp2

3DES, MD5,
PFS group 2

CF

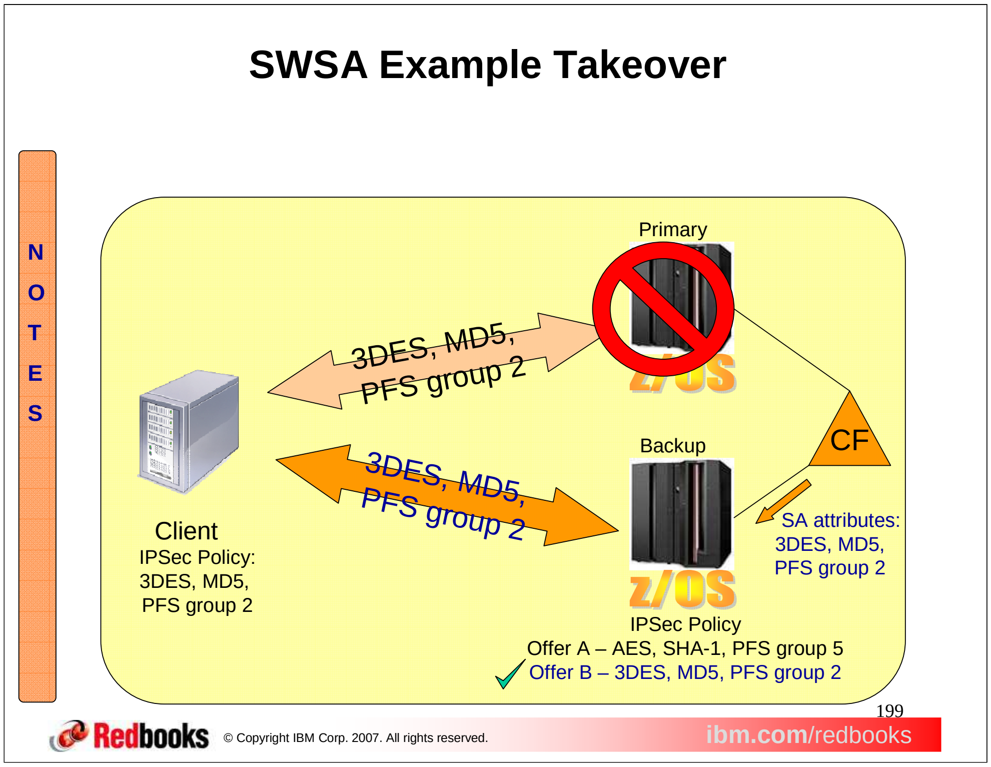
Backup

Client
IPSec Policy:
3DES, MD5,
PFS group 2

IPSec Policy
Offer A – AES, SHA-1, PFS group 5
✓ Offer B – 3DES, MD5, PFS group 2

198

The client on the left establishes an SA with the z/OS sysplex. The SA information is stored in the coupling facility.



SWSA Example Takeover

N O T E S

Primary

3DES, MD5,
PFS group 2

Backup

CF

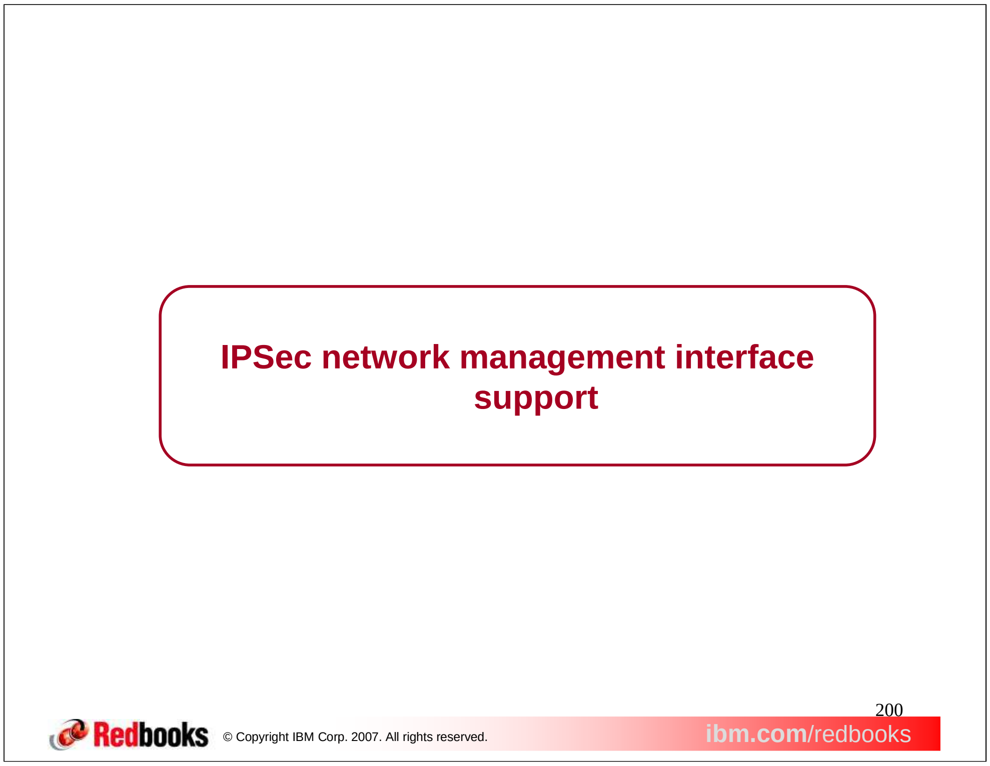
3DES, MD5,
PFS group 2

Client
IPSec Policy:
3DES, MD5,
PFS group 2

SA attributes:
3DES, MD5,
PFS group 2

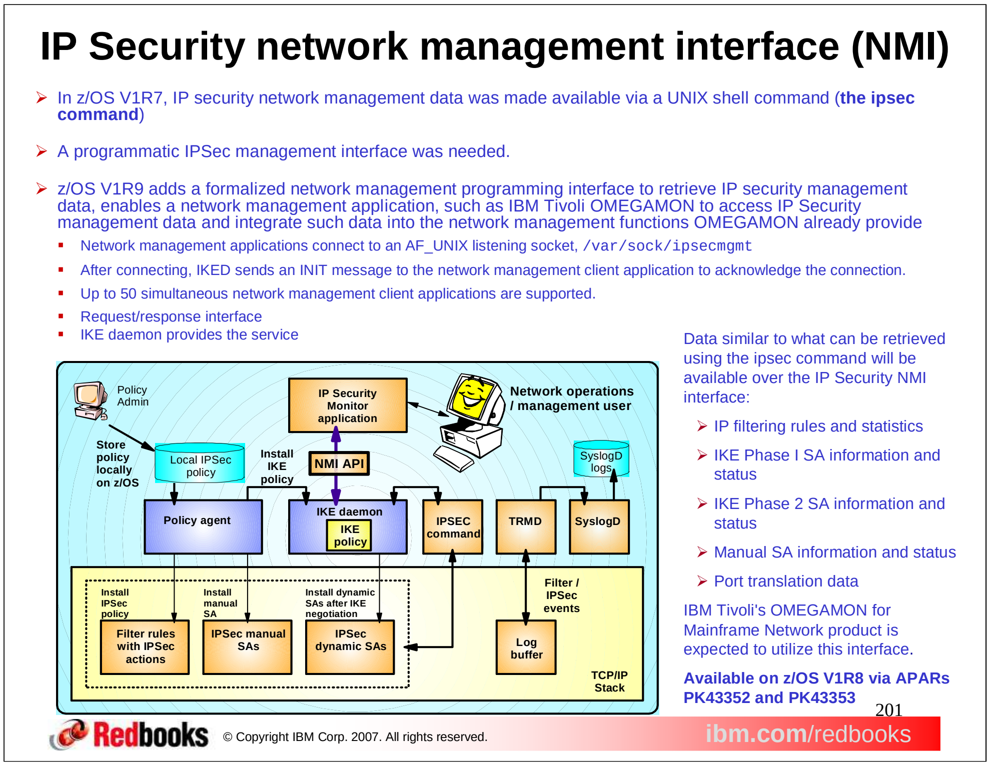
IPSec Policy
Offer A – AES, SHA-1, PFS group 5
✓ Offer B – 3DES, MD5, PFS group 2

When the primary z/OS box becomes unavailable, the backup performs a SWSA takeover to re-establish the SA with the client. The SA attributes are retrieved from the coupling facility and the correct SA offer is used for the negotiation.



IPSec network management interface support

This section discusses the IPSec network management interface support.



IP Security network management interface (NMI)

- In z/OS V1R7, IP security network management data was made available via a UNIX shell command (**the ipsec command**)
- A programmatic IPSec management interface was needed.
- z/OS V1R9 adds a formalized network management programming interface to retrieve IP security management data, enables a network management application, such as IBM Tivoli OMEGAMON to access IP Security management data and integrate such data into the network management functions OMEGAMON already provide
  - Network management applications connect to an AF_UNIX listening socket, `/var/sock/ipsecmgmt`
  - After connecting, IKED sends an INIT message to the network management client application to acknowledge the connection.
  - Up to 50 simultaneous network management client applications are supported.
  - Request/response interface
  - IKE daemon provides the service

Data similar to what can be retrieved using the ipsec command will be available over the IP Security NMI interface:

- IP filtering rules and statistics
- IKE Phase I SA information and status
- IKE Phase 2 SA information and status
- Manual SA information and status
- Port translation data

IBM Tivoli's OMEGAMON for Mainframe Network product is expected to utilize this interface.

Available on z/OS V1R8 via APARs PK43352 and PK43353

In z/OS V1R7, IP security network management data was made available via a UNIX shell command (the ipsec command). Adding a formalized network management programming interface to retrieve IP security management data, enables a network management application to access IP Security management data and integrate such data into the network management functions already provided.

The ipsec command displays and manages system information for Integrated IPSec, but a programmatic interface was needed for network management applications to perform these actions without needing to resort to screen-scraping.

A network management interface is implemented for Integrated IPSec in z/OS V1R9. The IKE daemon implements an AF_UNIX listening socket that accepts connections, and uses a request/response model for providing IPSec management data and control. Consequently, IKED must be running in order to make use of this service. Data similar to what can be retrieved using the ipsec command will be available over the IP Security NMI interface:

•IP filtering rules and statistics

•IKE Phase I SA information and status

•IKE Phase 2 SA information and status

•Manual SA information and status

•Port translation data

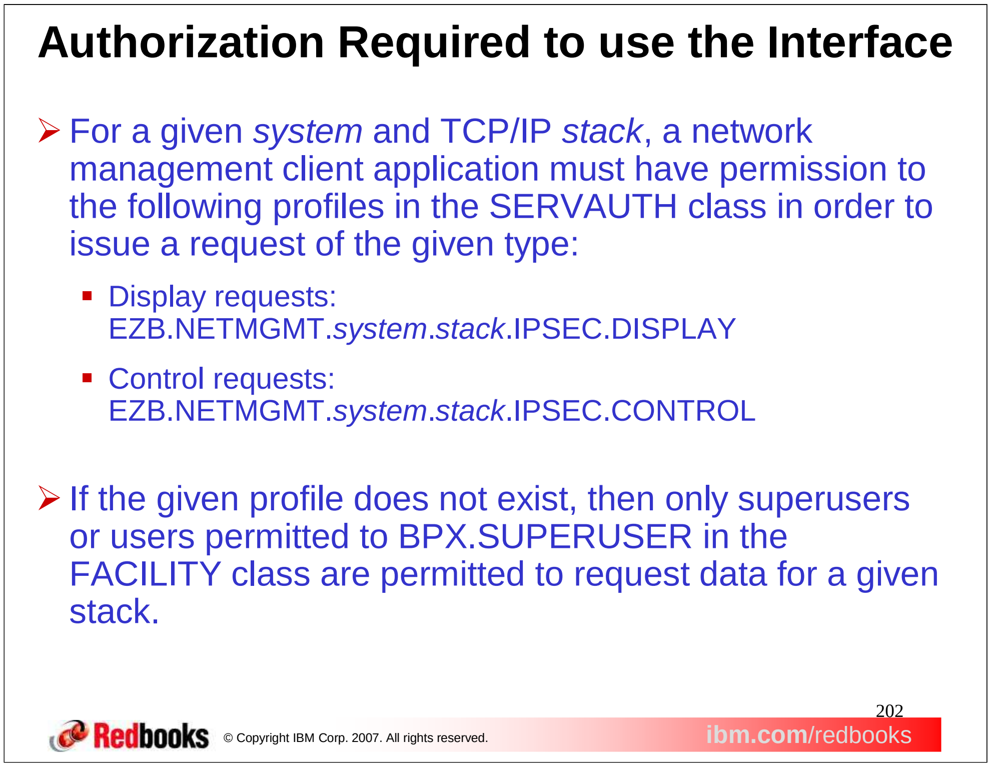
In addition the interface allows for the following:

•Activation or deactivation of manual tunnels

•Activation, deactivation, or refresh of IP tunnels

•Deactivation or refresh of IKE tunnels

•Load default IP filters or policy IP filters



Authorization Required to use the Interface

- For a given *system* and TCP/IP *stack*, a network management client application must have permission to the following profiles in the SERVAUTH class in order to issue a request of the given type:
  - Display requests: EZB.NETMGMT.*system*.*stack*.IPSEC.DISPLAY
  - Control requests: EZB.NETMGMT.*system*.*stack*.IPSEC.CONTROL
- If the given profile does not exist, then only superusers or users permitted to BPX.SUPERUSER in the FACILITY class are permitted to request data for a given stack.

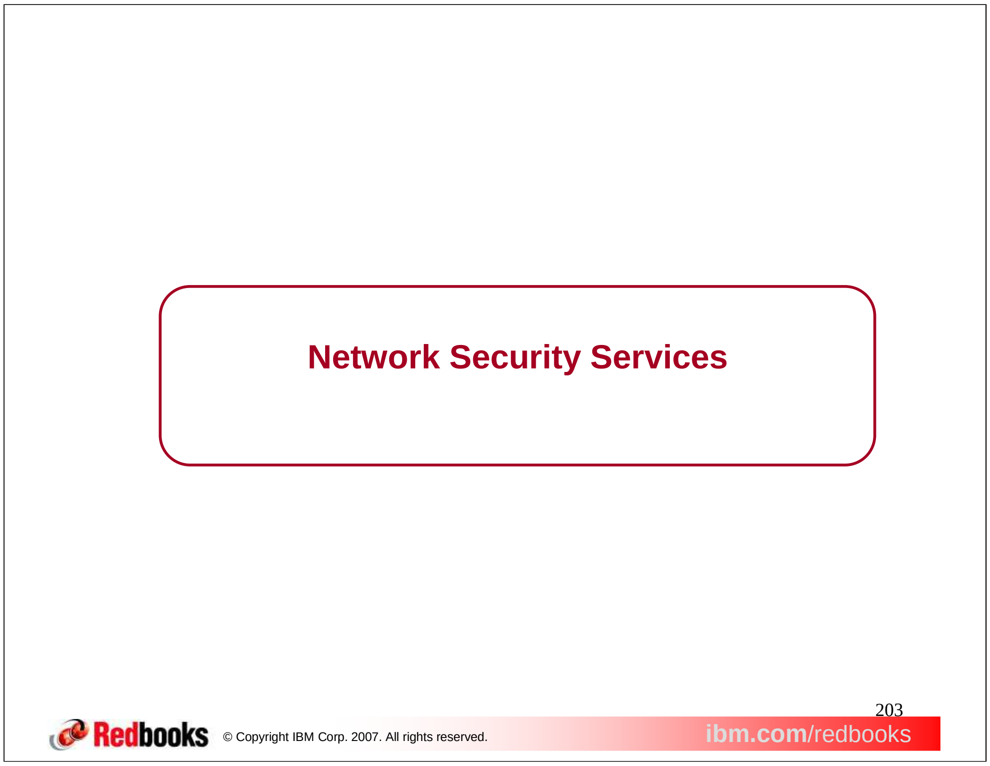
202

Redbooks ibm.com/redbooks

SAF authorization is required for most request types, to the resource EZB.NETMGMT.*system*.*stack*.IPSEC.*type,* in the SERVAUTH class, where *type* is DISPLAY or CONTROL. If this profile does not exist, then only superusers or users permitted to BPX.SUPERUSER in the FACILITY class are permitted to request data for a given stack. Authorization failure will be indicated by an EACCES return code in the response message.

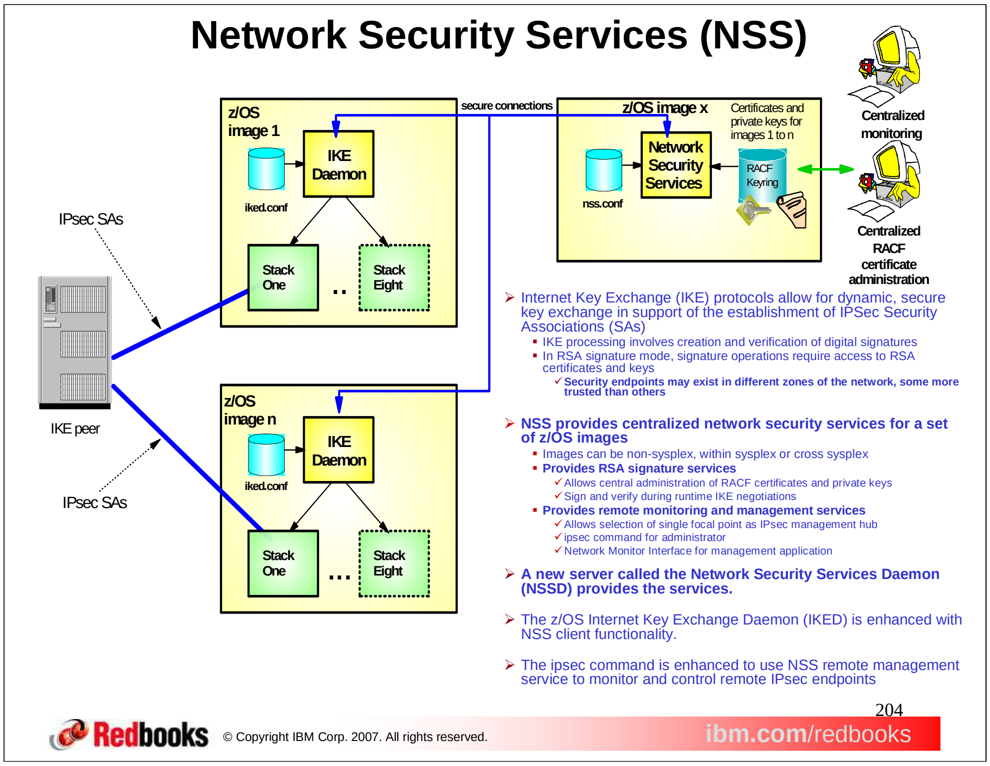
Other errors may occur in addition to authorization failure. Most such errors relate to improperly formed request messages. These errors will be indicated in the response message header's return code and reason code fields, and the possible values for these are documented in the *IP Programmer's Guide and Reference*. In certain error cases IKED will close the connection.

The set of errors that may be reflected in the response message's reason code are documented in the *IP Programmer's Guide and Reference*.



Network Security Services

This section describes a new z/OS Communications Server function called Network security services (NSS).



Network Security Services (NSS)

- Internet Key Exchange (IKE) protocols allow for dynamic, secure key exchange in support of the establishment of IPSec Security Associations (SAs)
  - IKE processing involves creation and verification of digital signatures
  - In RSA signature mode, signature operations require access to RSA certificates and keys
    - ✓ **Security endpoints may exist in different zones of the network, some more trusted than others**
- **NSS provides centralized network security services for a set of z/OS images**
  - Images can be non-sysplex, within sysplex or cross sysplex
  - **Provides RSA signature services**
    - ✓ Allows central administration of RACF certificates and private keys
    - ✓ Sign and verify during runtime IKE negotiations
  - **Provides remote monitoring and management services**
    - ✓ Allows selection of single focal point as IPsec management hub
    - ✓ ipsec command for administrator
    - ✓ Network Monitor Interface for management application
- **A new server called the Network Security Services Daemon (NSSD) provides the services.**
- The z/OS Internet Key Exchange Daemon (IKED) is enhanced with NSS client functionality.
- The ipsec command is enhanced to use NSS remote management service to monitor and control remote IPsec endpoints

204

Redbooks ibm.com/redbooks

Network Security Services (NSS) centralizes the sensitive keying material that would otherwise need to reside in less secure zones of the network onto a single location in the most secure zone of the network. In addition, NSS allows for centralized configuration and administration of certificates.

Network Security Services provide centralized certificate services, monitoring and management for IPSec security across z/OS systems within and across sysplexes. Network Security Services allow IPSec certificates to be kept in a single location, rather than having them reside on each z/OS node. The z/OS Communications Server IKE daemon is enhanced so that it can be configured to act as a Network Security client. Configuration is on a per-stack basis, such that each NSS-enabled stack will appear to the Network Security Server as an independent client. For TCP/IP stacks that are not configured to use Network Security Services, the IKE daemon will continue to manage certificates out of a local keyring.

Specifically, NSS provides a central SAF-enabled repository for RSA certificates along with signature services within the most trusted zones. It eliminates the need to distribute certificates to security endpoints. NSS centralizes and reduce configuration and deployment complexity, especially when used along with Centralized Policy Services. It offloads digital signature operations from IKE daemon (the NSS client) and it enables monitoring and management of remote IPSec endpoints through the ipsec command and a network management programming interface.

The network security services (NSS) server provides a set of network security services for IPSec. These include the certificate (and digital signature) service and the network management service. The certificate service and network management service are used by NSS clients. When an NSS client uses the NSS certificate service, the NSS server creates and verifies RSA signatures on the behalf of the NSS client using RSA certificates that are stored only at the NSS server. When an NSS client uses the network management service, the NSS server routes IPSec network management interface (NMI) requests to that NSS client, which enables the NSS client to be managed remotely. The NSS client provides the NSS server with responses to these requests.

As mention earlier, the IKE daemon can be configured to act as an NSS client on behalf of multiple TCP/IP stacks. A separate connection is maintained to the server for each NSS-enabled TCP/IP stack, so each TCP/IP stack appears as a separate NSS client to the NSS server. The -z option of the ipsec command or the IPSec NMI can be used to manage NSS clients that use the NSS network management service. For details about using the ipsec command to manage NSS clients, see *z/OS Communications Server: IP System Administrator’s Commands*. For details about using the IPSec NMI to manage NSS clients, see *z/OS Communications Server: IP Programmer’s Guide and Reference*.

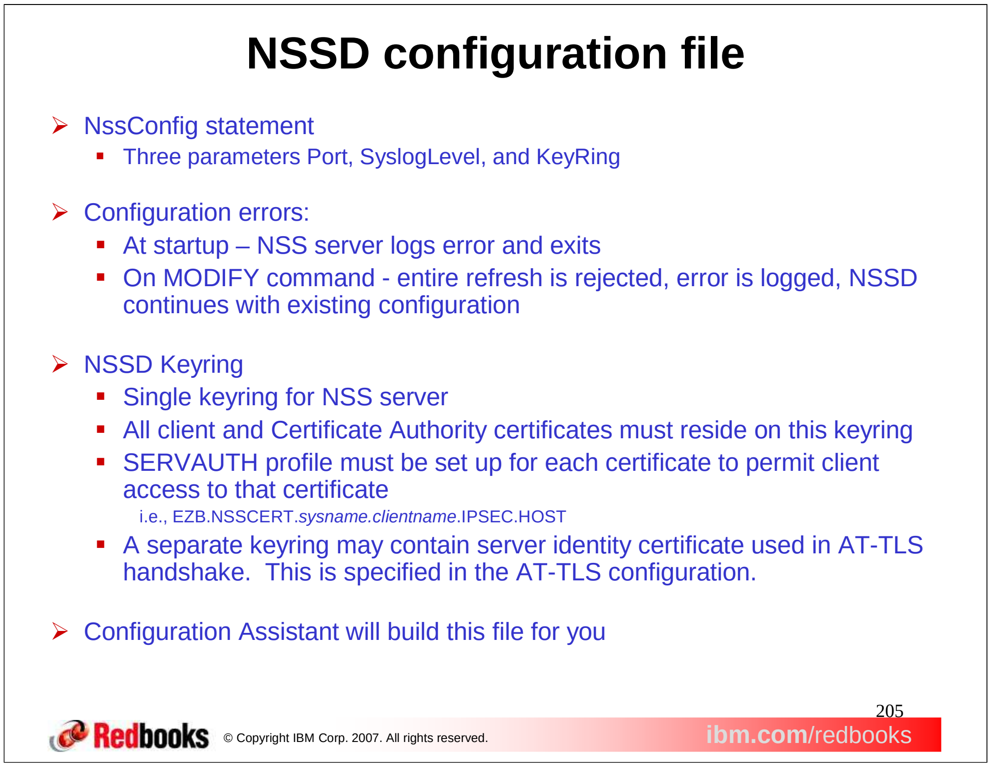
In a nutshell NSS does the following:

Provides a central SAF-enabled repository for RSA certificates along with signature services within the most trusted zones

Eliminates the need to distribute certificates to security endpoints
Centralizes and reduces configuration and deployment complexity, especially when used along with Centralized Policy Services

Offloads digital signature operations from IKE daemon (the NSS client)

Enables monitoring and management of remote IPsec endpoints through the ipsec command and a network management programming interface.



NSSD configuration file

- NssConfig statement
  - Three parameters Port, SyslogLevel, and KeyRing
- Configuration errors:
  - At startup – NSS server logs error and exits
  - On MODIFY command - entire refresh is rejected, error is logged, NSSD continues with existing configuration
- NSSD Keyring
  - Single keyring for NSS server
  - All client and Certificate Authority certificates must reside on this keyring
  - SERVAUTH profile must be set up for each certificate to permit client access to that certificate

    i.e., EZB.NSSCERT.*sysname.clientname*.IPSEC.HOST
  - A separate keyring may contain server identity certificate used in AT-TLS handshake. This is specified in the AT-TLS configuration.
- Configuration Assistant will build this file for you

205

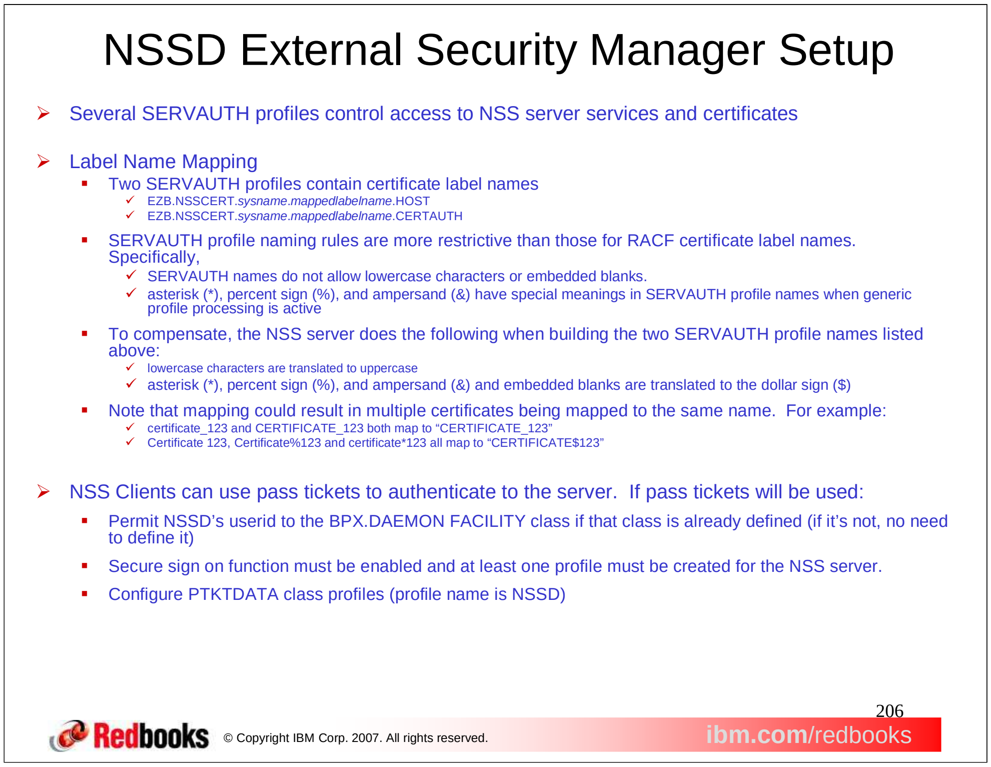
The form of the NSSD configuration file is similar to that of the IKE daemon. All of the parameters are contained within the NssConfig statement, enclosed by curly braces. Three parameters are available. The **Port** parameter identifies the TCP port to which the NSS server will bind. The default value is 4159, which is an IANA registered port. All client requests must come in through this TCP port. The **SyslogLevel** parameter specifies the level of logging to obtain from the NSS server. The default value is 1 which is the minimal NSS daemon syslog output. The **KeyRing** parameter specifies the SAF key ring database, which contains certificates and keys used when creating and verifying signatures for NSS clients. There is no default value for this parameter. NSS certificate services will not be activated if the KeyRing parameter is not specified.

The NSS server logs an error and exits when configuration errors are detected during startup. When errors in the configuration file, that is being refresh as a result of the Modify command, are detected then the error is logged and the refresh is rejected.

The NSS server's key ring serves a similar purpose as the IKE daemon's key ring. It contains certificates that are used in the process of creating and verifying signatures that are exchanged during RSA signature authentication. A personal certificate or site certificate contained on the NSS server's key ring represents the identity of an NSS client, whereas a certificate contained on the IKE daemon's key ring represents a local stack's identity. See chapter 18 of the IP Configuration Guide for more details.

Certificates for all NSS clients must reside on this one key ring. The same commands that are used to create and manage the IKE daemon's key ring also apply to the NSS server's key ring. For examples of how to create and manage the IKE daemon's key ring, see the IP Configuration Guide Appendix E, "Steps for preparing to run IP security."

You must create a SERVAUTH resource profile for each NSS client certificate that is added to the NSS server's key ring. For details, see chapter 18 of the IP Configuration Guide.



NSSD External Security Manager Setup

- Several SERVAUTH profiles control access to NSS server services and certificates
- Label Name Mapping
  - Two SERVAUTH profiles contain certificate label names
    - EZB.NSSCERT.*sysname.mappedlabelname*.HOST
    - EZB.NSSCERT.*sysname.mappedlabelname*.CERTAUTH
  - SERVAUTH profile naming rules are more restrictive than those for RACF certificate label names. Specifically,
    - SERVAUTH names do not allow lowercase characters or embedded blanks.
    - asterisk (*), percent sign (%), and ampersand (&) have special meanings in SERVAUTH profile names when generic profile processing is active
  - To compensate, the NSS server does the following when building the two SERVAUTH profile names listed above:
    - lowercase characters are translated to uppercase
    - asterisk (*), percent sign (%), and ampersand (&) and embedded blanks are translated to the dollar sign ($)
  - Note that mapping could result in multiple certificates being mapped to the same name. For example:
    - certificate_123 and CERTIFICATE_123 both map to “CERTIFICATE_123”
    - Certificate 123, Certificate%123 and certificate*123 all map to “CERTIFICATE$123”
- NSS Clients can use pass tickets to authenticate to the server. If pass tickets will be used:
  - Permit NSSD’s userid to the BPX.DAEMON FACILITY class if that class is already defined (if it’s not, no need to define it)
  - Secure sign on function must be enabled and at least one profile must be created for the NSS server.
  - Configure PTKTDATA class profiles (profile name is NSSD)

This slide describes the External Security Manager (ESM) setup required for NSS. The name of the SERVAUTH profiles contains a *sysname* which is the name of the z/OS system on which NSSD is running. Some profile names contain a *clientname* which is the symbolic name of the NSS client. A *mappedlabelname* is also present in some profile names and that is the mapped version of a certificate label. To allow access the userid under which the NSS client registers must be permitted READ access to the following SERVAUTH profiles.

EZB.NSS.*sysname.clientname*.IPSEC.CERT controls whether a NSS client can register with the NSS server for the NSS certificate service.

EZB.NSS.*sysname.clientname*.IPSEC.NETMGMT controls whether a NSS client can register with the NSS server for NSS network management service. **EZB.NSSCERT.*sysname.mappedlabelname*.CERTAUTH** controls whether a NSS client can access a given CERTAUTH certificate on the NSS server's key ring. This profile controls access to a single certificate, identified by the mappedlabelname.

EZB.NSSCERT.*sysname.mappedlabelname*.HOST controls whether a NSS client can access a given PERSONAL or SITE certificate on the NSS server's key ring. This profile controls access to a single certificate, identified by the mappedlabelname. Name mapping rules will be discussed later on.

To allow access, the userid under which the NMI application or the ipsec command will run must be permitted READ access to the following SERVAUTH profiles.

EZB.NETMGMT.*sysname.clientname*.IPSEC.DISPLAY controls whether a z/OS user can issue NMI monitoring requests to the NSS server on behalf of a NSS client (i.e. GET_xxx requests) or issue the ipsec command with the -z option to perform a display action for a NSS client (i.e. display options) .

EZB.NETMGMT.*sysname.clientname*.IPSEC.CONTROL controls whether a z/OS user can issue NMI management requests to the NSS server on behalf of a NSS client (e.g. activate/deactivate requests) or issue the ipsec command with the -z option to perform a management action to a NSS client (e.g. activate/deactivate options).

EZB.NETMGMT.*sysname.sysname*.NSS.DISPLAY controls whether a z/OS user can issue NMI requests to display connections to the NSS server or issue the ipsec command with the -x option to display connections to the NSS server.

During the processing of certificate operations, the NSS server validates that an NSS client is authorized to access the certificates required to complete the operation. The NSS server consults SERVAUTH profiles to perform this validation. The profile names consulted by the NSS server are dynamically constructed by the NSS server using the following information:

The system name on which the NSS server is running

The label of the certificate that is used during a certificate operation

The certificate operation that is being performed:

When processing a request to create a signature, the format of the profile that is consulted is **EZB.NSSCERT.*sysname.mappedlabelname*.HOST**. When processing a request to obtain a list of CA certificates, the format of the profile consulted is **EZB.NSSCERT.*sysname.mappedlabelname*.CERTAUTH**.

The NSS server creates a mapped label name using the following algorithm:

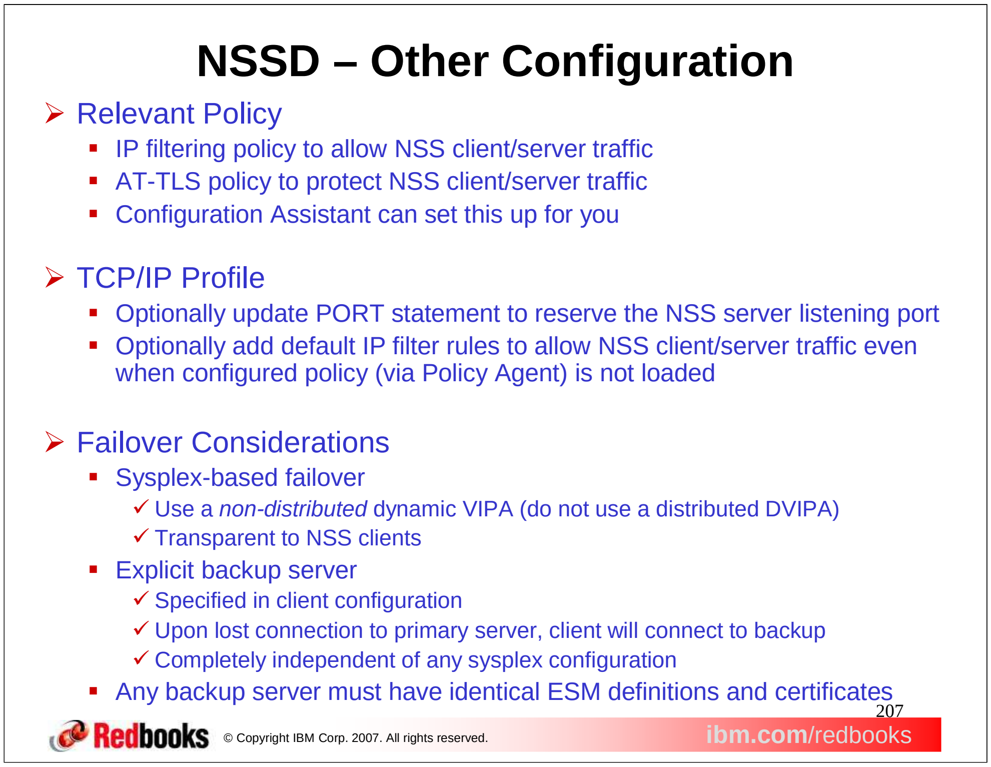
All lowercase alphabetic characters in a certificate’s label are changed to uppercase. This is necessary because the class descriptor table for the SERVAUTH profile permits only uppercase profile names.

The asterisk (*), percent sign (%), and ampersand (&) are replaced by a dollar sign ($). This is necessary because these characters have special meaning when generic profile processing is active.

All embedded blanks are also replaced by a dollar sign ($). This is necessary because blanks are not allowed in SERVAUTH profile names.

Note that the administrator of the NSS server must define profiles using the mapped label names generated by this algorithm. When the certificate’s label name contains lowercase characters, the administrator must change each lowercase character to uppercase. When the certificate’s label name contains the characters *, %, &, or a blank character, the administrator must replace each occurrence with a dollar sign ($) character.

If pass tickets will be used to authenticate clients , then the NSS userid must be permitted to the BPX.DAEMON class and



NSSD – Other Configuration

- Relevant Policy
  - IP filtering policy to allow NSS client/server traffic
  - AT-TLS policy to protect NSS client/server traffic
  - Configuration Assistant can set this up for you
- TCP/IP Profile
  - Optionally update PORT statement to reserve the NSS server listening port
  - Optionally add default IP filter rules to allow NSS client/server traffic even when configured policy (via Policy Agent) is not loaded
- Failover Considerations
  - Sysplex-based failover
    - ✓ Use a *non-distributed* dynamic VIPA (do not use a distributed DVIPA)
    - ✓ Transparent to NSS clients
  - Explicit backup server
    - ✓ Specified in client configuration
    - ✓ Upon lost connection to primary server, client will connect to backup
    - ✓ Completely independent of any sysplex configuration
  - Any backup server must have identical ESM definitions and certificates

The NSS server communicates with NSS clients using the TCP protocol. The NSS server binds to all stacks using INADDR_ANY. IP filters rules must be defined to permit NSS client/server traffic for any IP security stacks that contain an interface to which the NSS client will connect. In addition to the default IP filters, the configured policy which is delivered through the z/OS Policy Agent must also be updated to allow this traffic. Policy-based filters are in effect when a stack initializes with the Policy Agent or when the **ipsec -f reload** command has been issued. IP security filter policy is defined in Policy Agent configuration files. For details about defining IP security policy files, see the Policy Agent and policy applications chapter of *z/OS Communications Server: IP Configuration Reference*. Note that the SourcePortRange value on the IpService statements must include the value specified on the port parameter of the NssConfig statement in the NSS server configuration file.

The NSS server and the IKE daemon require that communications between the NSS server and NSS clients be secured using AT-TLS. You must define AT-TLS rules to secure this communication. Enable AT-TLS processing for a stack by specifying the TTLS parameter on the TCPCONFIG statement in the TCP/IP profile. Specific AT-TLS policy is configured in Policy Agent configuration files. For details about enabling AT-TLS and configuring AT-TLS policy, see Chapter 19 of the IP Configuration Guide. You should define AT-TLS policy such that only cipher suites requiring TLS encryption are exchanged with NSS clients. Failure to restrict the cipher suites to those requiring encryption can result in sensitive information flowing in the clear across an untrusted network. You must define AT-TLS policy for each stack through which the NSS server will communicate with an NSS client.

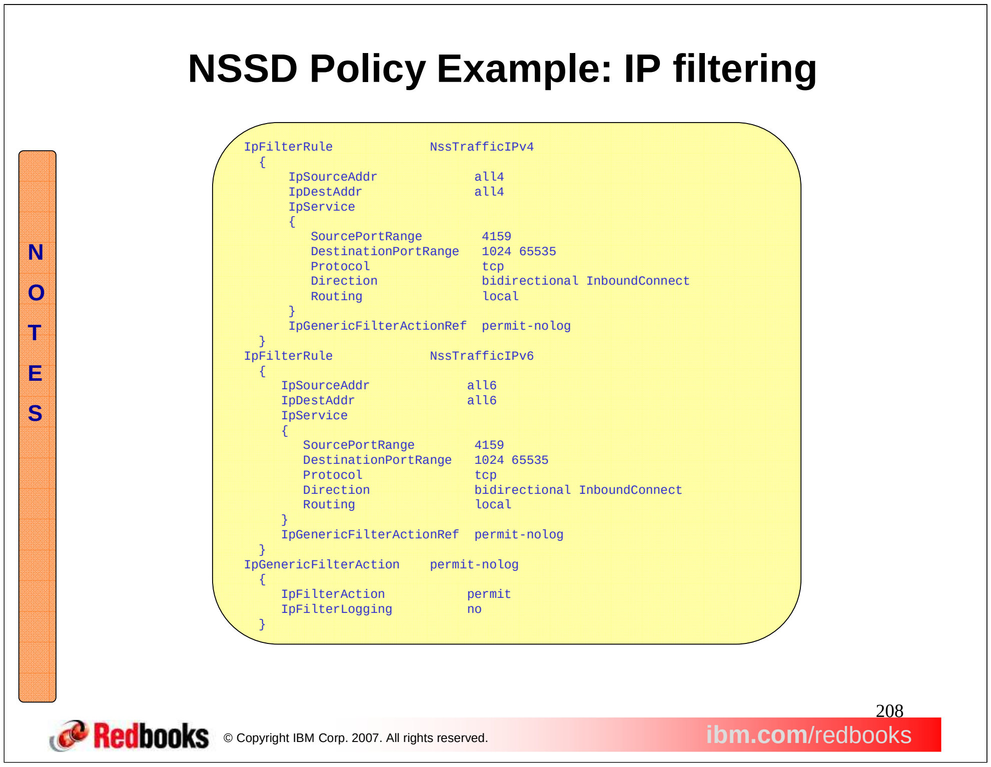
The NSS server acts as the server during an SSL handshake. To act in the server role of an SSL handshake, the NSS server must have access to a private key and certificate verifying its ownership of that private key. For information about creating and managing keys and certificates for servers using AT-TLS, see Appendix B of the IP Configuration Guide. Note that NMI applications use AF_UNIX sockets, so AT-TLS protection does not apply to those connections.

By default the NSS server uses TCP port 4159, but this value is configurable using the Port parameter of the NssConfig statement in the NSS server configuration file. It is also a good idea to update the PORT statement in the TCP/IP profile to reserve the port that the NSS server will use when listening for client connections.

Default IP filter policy is defined in the TCP/IP profile. Updating default IP filter policy to permit communications between the NSS server and NSS clients is optional. Default IP filter policy is in effect only when IP security filter policy cannot be loaded or when the **ipsec -f default** command has been issued. For details about defining default IP filter policy in the TCP/IP profile, see *z/OS Communications Server: IP Configuration Reference*. Note that the SRCport value in the filter rules must include the value specified on the port parameter of the NssConfig statement in the NSS server configuration file.

NSS clients can use the NSS certificate service when negotiating phase 1 security associations. Network monitoring applications can use the NSS network management service to display information about NSS clients. As such, the NSS server should be treated as an application that requires high availability. Take steps to quickly recover from an outage that impacts the NSS server's ability to respond to clients. Recovery configurations for the NSS server include the following:
For recovery of NSS server workload by another NSS server within a sysplex, configure NSS clients to connect to the NSS server on a non-distributed dynamic VIPA. TCP/IP stacks configured as backup for the dynamic VIPA must have the necessary external security manager definitions and certificates to support the NSS clients, and an NSS server must be running on the z/OS system hosting the TCP/IP stack configured as backup. Do not configure NSS clients to connect to a distributed DVIPA address on the NSS server. If a distributed DVIPA is used, the **ipsec** command and IPSec NMI can manage only NSS clients that have been distributed to the system on which the **ipsec** command is being run or the system on which the IPSec NMI is invoked.

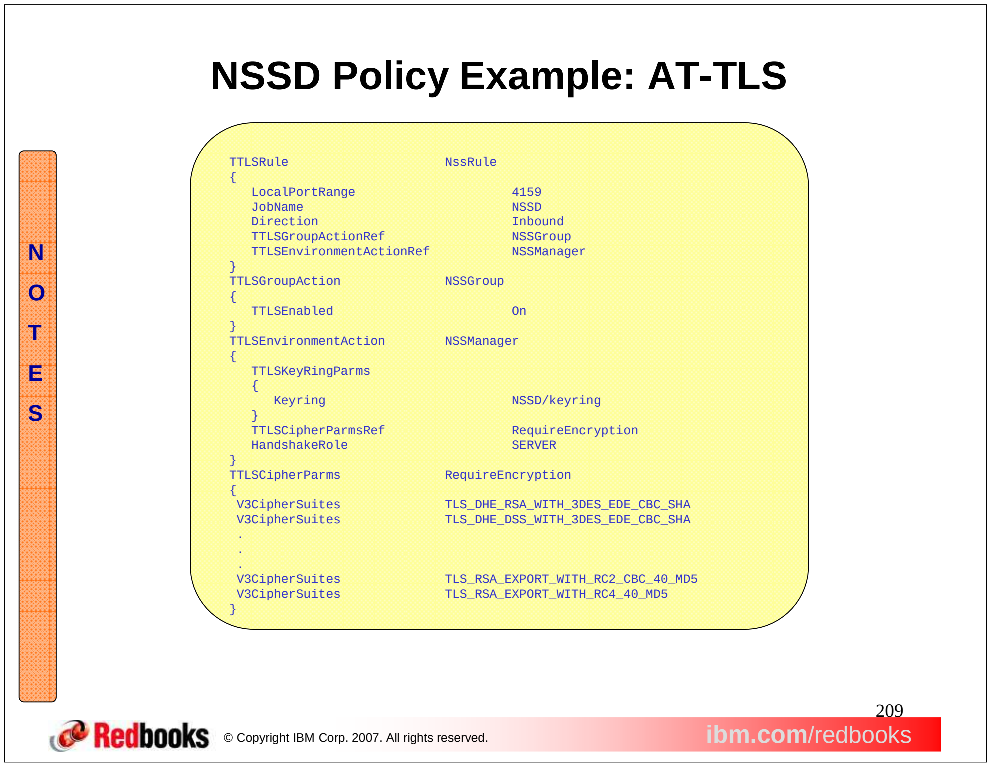
Alternatively, an IKE daemon running as an NSS client can be configured to connect to a backup NSS server with the NetworkSecurityServerBackup parameter on the IkeConfig statement in the IKED.CONF file. When the IKE daemon is unable to connect to the primary NSS server, or when it loses its connection with the primary server, the IKE daemon attempts to connect to the server configured as backup. This recovery configuration can be used regardless of sysplex configurations. The backup server must be configured with all necessary external security manager definitions and certificates to support the NSS clients. For additional details about the IkeConfig statement, see *z/OS Communications Server: IP Configuration Reference*.



NSSD Policy Example: IP filtering

```
IpFilterRule             NssTrafficIPv4
  {
      IpSourceAddr             all4
      IpDestAddr               all4
      IpService
      {
         SourcePortRange        4159
         DestinationPortRange   1024 65535
         Protocol               tcp
         Direction              bidirectional InboundConnect
         Routing                local
      }
      IpGenericFilterActionRef  permit-nolog
  }
IpFilterRule             NssTrafficIPv6
  {
     IpSourceAddr             all6
     IpDestAddr               all6
     IpService
     {
        SourcePortRange        4159
        DestinationPortRange   1024 65535
        Protocol               tcp
        Direction              bidirectional InboundConnect
        Routing                local
     }
     IpGenericFilterActionRef  permit-nolog
  }
IpGenericFilterAction    permit-nolog
  {
     IpFilterAction           permit
     IpFilterLogging          no
  }
```

This is an example of a valid IP filter policy definition for use with NSS assuming that the NSS server will be listening on port 4159.



NSSD Policy Example: AT-TLS

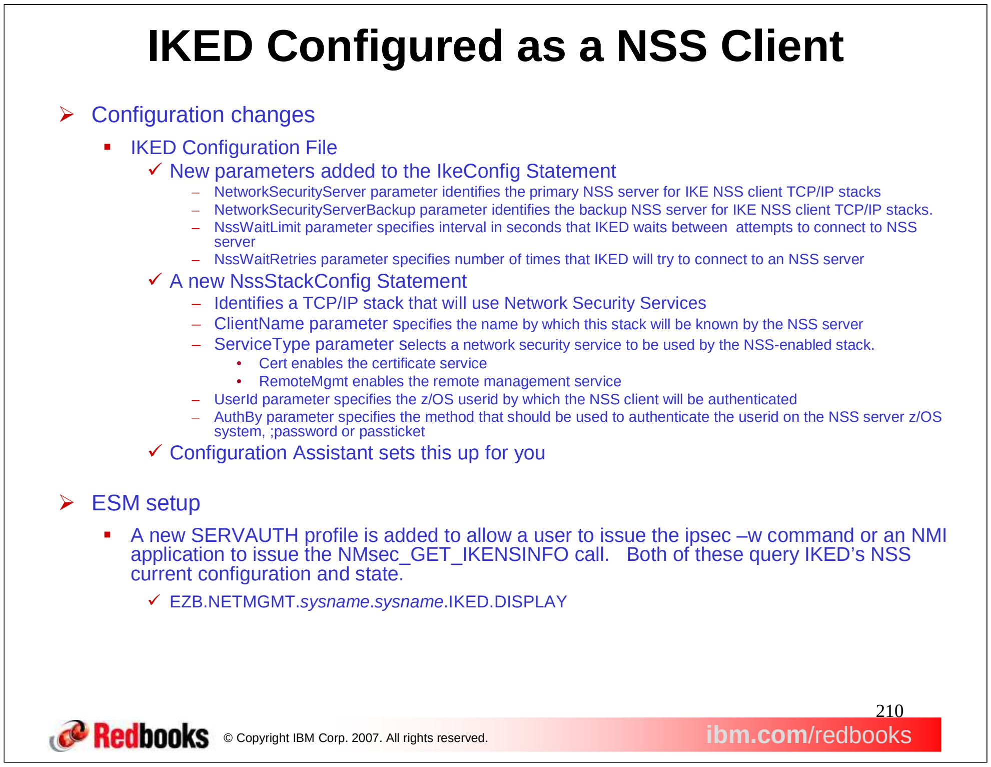
N O T E S

```
TTLSRule                       NssRule
{
   LocalPortRange                       4159
   JobName                              NSSD
   Direction                            Inbound
   TTLSGroupActionRef                   NSSGroup
   TTLSEnvironmentActionRef             NSSManager
}
TTLSGroupAction                NSSGroup
{
   TTLSEnabled                          On
}
TTLSEnvironmentAction          NSSManager
{
   TTLSKeyRingParms
   {
      Keyring                           NSSD/keyring
   }
   TTLSCipherParmsRef                   RequireEncryption
   HandshakeRole                        SERVER
}
TTLSCipherParms                RequireEncryption
{
 V3CipherSuites                TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
 V3CipherSuites                TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
 .
 .
 .
 V3CipherSuites                TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
 V3CipherSuites                TLS_RSA_EXPORT_WITH_RC4_40_MD5
}
```

209

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

This is an example of a valid AT-TLS policy definition for use with NSS assuming that the NSS server will be listening on port 4159.



IKED Configured as a NSS Client

- Configuration changes
  - IKED Configuration File
    - ✓ New parameters added to the IkeConfig Statement
      - NetworkSecurityServer parameter identifies the primary NSS server for IKE NSS client TCP/IP stacks
      - NetworkSecurityServerBackup parameter identifies the backup NSS server for IKE NSS client TCP/IP stacks.
      - NssWaitLimit parameter specifies interval in seconds that IKED waits between attempts to connect to NSS server
      - NssWaitRetries parameter specifies number of times that IKED will try to connect to an NSS server
    - ✓ A new NssStackConfig Statement
      - Identifies a TCP/IP stack that will use Network Security Services
      - ClientName parameter Specifies the name by which this stack will be known by the NSS server
      - ServiceType parameter Selects a network security service to be used by the NSS-enabled stack.
        - Cert enables the certificate service
        - RemoteMgmt enables the remote management service
      - UserId parameter specifies the z/OS userid by which the NSS client will be authenticated
      - AuthBy parameter specifies the method that should be used to authenticate the userid on the NSS server z/OS system, ;password or passticket
    - ✓ Configuration Assistant sets this up for you
- ESM setup
  - A new SERVAUTH profile is added to allow a user to issue the ipsec –w command or an NMI application to issue the NMsec_GET_IKENSINFO call.   Both of these query IKED’s NSS current configuration and state.
    - ✓ EZB.NETMGMT.*sysname*.*sysname*.IKED.DISPLAY

210

ibm.com/redbooks

Four new parameters are added to the IkeConfig statement in the IKE daemon configuration file. All of these are focused on NSS exploitation and apply to all stacks that use NSS through this IKE daemon.

The **NetworkSecurityServer** parameter identifies the primary NSS server for IKE NSS client TCP/IP stacks. A single server is used for all of the TCP/IP stacks configured as NSS clients. Stacks can be configured individually as NSS clients. Stacks with a corresponding NssStackConfig statement are treated as NSS clients; stacks without a corresponding NssStackConfig statement rely solely on local IKE resources. While the NetworkSecurityServer parameter is optional, you must specify at least one of the NetworkSecurityServer and NetworkSecurityServerBackup parameters in order for any of the TCP/IP stacks to use an NSS server. The value of the NetworkSecurityServer parameter can be changed via the MODIFY IKED,REFRESH command. However, existing connections remain in place and new connection attempts will use the new value.

The **NetworkSecurityServerBackup** parameter identifies the backup NSS server for IKE NSS client TCP/IP stacks. Network Security clients switch between the primary and the backup NSS servers whenever their current server becomes unresponsive. If both the primary and the backup become unresponsive, the Network Security client attempts to connect to the primary and the backup in a round-robin fashion until a successful connection is made.

The **NssWaitLimit** parameter specifies the number of seconds that a NSS client waits between connection attempts when trying to establish a connection with a NSS server. The product of the NssWaitLimit value multiplied by the NssWaitRetries value defines the maximum number of seconds that a NSS client attempts to connect to a NSS server before switching to another server. The

The new statement, **NssStackConfig,** should be coded for each TCP/IP stack that will use NSS. Only stacks with a corresponding NssStackConfig statement are eligible for services provided by a NSS server. Stacks that are not configured with an NssStackConfig statement are locally managed. NssStackConfig statements require that a valid NSS server is described in the IkeConfig statement. It is a configuration error to have a NssStackConfig statement without also specifying a NetworkSecurityServer parameter, a NetworkSecurityServerBackup parameter, or both. You can use the MODIFY IKED,REFRESH command to change which TCP/IP stacks are configured as NSS clients, as follows:

•Deleting a NSS client: If it is determined after a refresh that a NssStackConfig statement was removed, then the connection associated with the removed NssStackConfig statement is closed

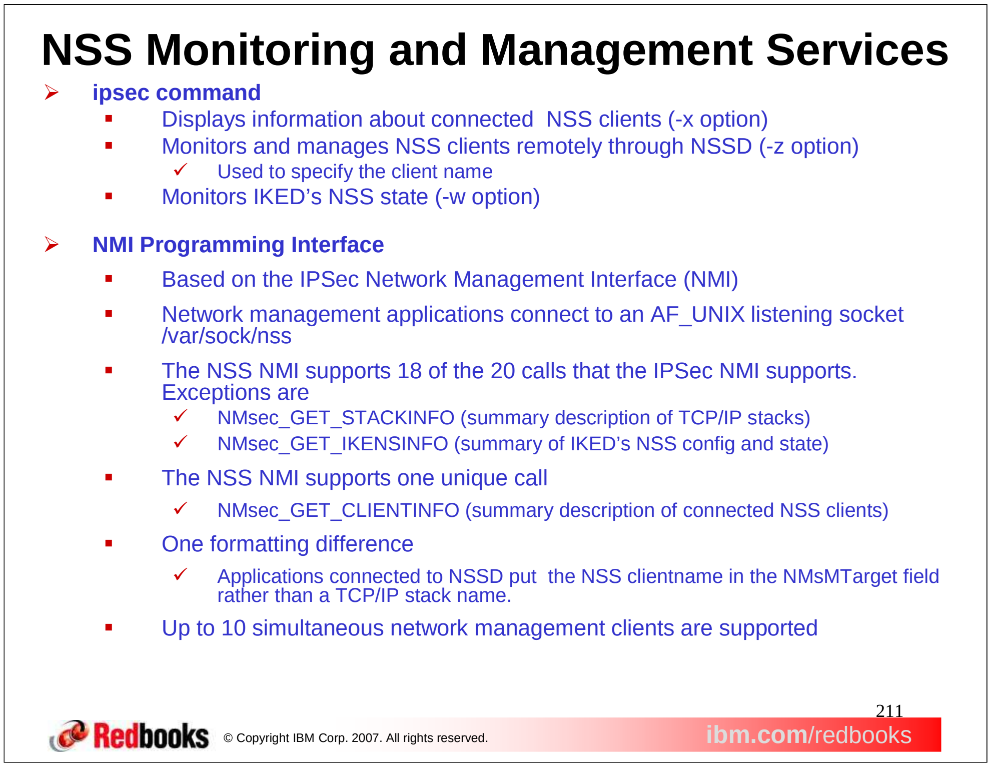
•Adding NSS client: If it is determined after a refresh that a new NssStackConfig statement was added, then the connection for the new stack is opened.

•Changing internal NssStackConfig values: Any change to an internal parameter of the NssStackConfig statement results in a disconnect followed by a reconnect.

The **ClientName** parameter, if not specified, is constructed by the IKE daemon from the z/OS system name and the TCP/IP stack name, as follows: `sysname_stackname.`   Regardless of how name is established, it must match the clientname portion of associated SERVAUTH profiles on the NSS server.

The specified userid, on the **UserId** parameter, must be defined on the z/OS system where the NSS server runs. Furthermore, the userid must be granted read access to any of the associated SERVAUTH profiles that control access to network security services or certificates on the NSS keyring.

sysname in the EZB.NETMGMT.*sysname*.*sysname*.IKED.DISPLAY profile is the name of the z/OS system on which IKED is running.



NSS Monitoring and Management Services

- **ipsec command**
  - Displays information about connected NSS clients (-x option)
  - Monitors and manages NSS clients remotely through NSSD (-z option)
    - ✓ Used to specify the client name
  - Monitors IKED's NSS state (-w option)
- **NMI Programming Interface**
  - Based on the IPSec Network Management Interface (NMI)
  - Network management applications connect to an AF_UNIX listening socket /var/sock/nss
  - The NSS NMI supports 18 of the 20 calls that the IPSec NMI supports. Exceptions are
    - ✓ NMsec_GET_STACKINFO (summary description of TCP/IP stacks)
    - ✓ NMsec_GET_IKENSINFO (summary of IKED's NSS config and state)
  - The NSS NMI supports one unique call
    - ✓ NMsec_GET_CLIENTINFO (summary description of connected NSS clients)
  - One formatting difference
    - ✓ Applications connected to NSSD put the NSS clientname in the NMsMTarget field rather than a TCP/IP stack name.
  - Up to 10 simultaneous network management clients are supported

211

Redbooks

ibm.com/redbooks

The role of the ipsec command expands in z/OS V1R9 to encompass NSS environments. You can use the -x primary option on the ipsec command to display connection information about NSS clients connected to the NSS server. You can Use the -z option on the ipsec command to specify the name of an NSS client rather than a name of a local TCP/IP stack. When the -z option is specified, the ipsec command obtains information about the NSS client from the NSS server.   The -z option is valid only on the system running the NSS server. The NSS client identified by the -z option must be connected to the NSS server. The NSS client must also be enabled to use the NSS network management service.   The new –z option directs the ipsec command to a local NSS server, which will forward the request to the specified NSS client (assuming that client is currently connected to the server).   Almost all of the existing ipsec options work with –z. The only exceptions are -p, which directs the command to a local TCP/IP stack, -x, which requests information about the local NSS server and -w, which requests information about the local IKE daemon. Refer to *IP System Administrator's Commands* for details and examples of each ipsec command option.

NSS provides a NMI programming interface. The NSS server supports a message format that is almost identical to that used by the IKE daemon for local IPSec monitoring and control. Like the local monitoring/control interface, these messages are exchanged over an AF_UNIX socket using a request-response model.   NSSD's AF_UNIX socket is named /var/sock/nss.

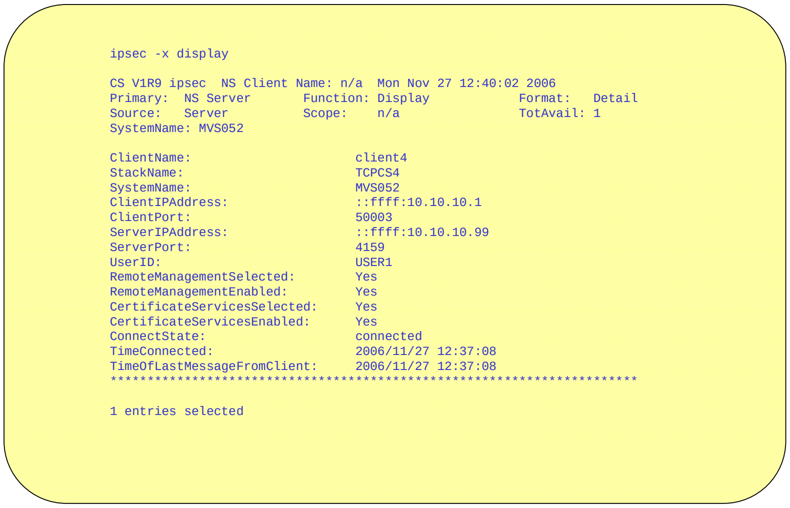
The NSS server supports all of the request messages described for the IKE daemon except for the NMsec_GET_STACKINFO and NMsec_GET_IKENSINFO requests (see “Application interfaces for monitoring IP filtering and IPSec” on page 442). In addition, the NSS server also supports the NMsec_GET_CLIENTINFO request message. The NMsec_GET_CLIENTINFO request obtains a list of NSS clients that are currently connected to the NSS server as well as summary information about each client. This message does not allow a filtering record. If the NMsMTarget field in the message header is blank, then information for all of the currently connected clients is returned. If a client name is specified in the NMsMTarget field, then information for only that client is returned as long as the client is connected. If the specified client is not connected, then no records are returned in the response message. Access to this function is controlled through the EZB.NETMGMT.*sysname*.*sysname*.NSS.DISPLAY resource definition in the SERVAUTH class.

The only difference between the NSS and IPSec NMI message format is that when an NMI message is sent to the NSS server, the NMsMTarget string in the message header identifies the remote NSS client to which the request is directed. Use the *clientname* field of the target NSS client in the NMsMTarget string, padded on the right with blanks. You can obtain the *clientname* values of each client connected to the NSS server by issuing the NMsec_GET_CLIENTINFO request.   The NMsMTarget field can be set to blanks for an NMsec_GET_CLIENTINFO request. If this field is set to blanks for any other request, the request is rejected with an appropriate error code the reply header.

Message layouts are defined in SEZANMAC(EZBNMSEA) and /usr/include/ezbnmsec.h. See the IP Programmer's Guide and Reference for details of each message

ipsec Command –x Option

NOTES



```
ipsec -x display

CS V1R9 ipsec  NS Client Name: n/a  Mon Nov 27 12:40:02 2006
Primary:  NS Server       Function: Display            Format:   Detail
Source:   Server          Scope:    n/a                TotAvail: 1
SystemName: MVS052

ClientName:                      client4
StackName:                       TCPCS4
SystemName:                      MVS052
ClientIPAddress:                 ::ffff:10.10.10.1
ClientPort:                      50003
ServerIPAddress:                 ::ffff:10.10.10.99
ServerPort:                      4159
UserID:                          USER1
RemoteManagementSelected:        Yes
RemoteManagementEnabled:         Yes
CertificateServicesSelected:     Yes
CertificateServicesEnabled:      Yes
ConnectState:                    connected
TimeConnected:                   2006/11/27 12:37:08
TimeOfLastMessageFromClient:     2006/11/27 12:37:08
***********************************************************************

1 entries selected
```

Here are a few noteworthy points regarding the output of this command.

The summary lines at the top are quite similar to that of most other ipsec command options. Nothing too exciting here.

The first several detail lines (ClientName through UserID) display the client identity and address information

The next four lines describe the client configuration as well as the services that are actually enabled (per SERVAUTH profiles):

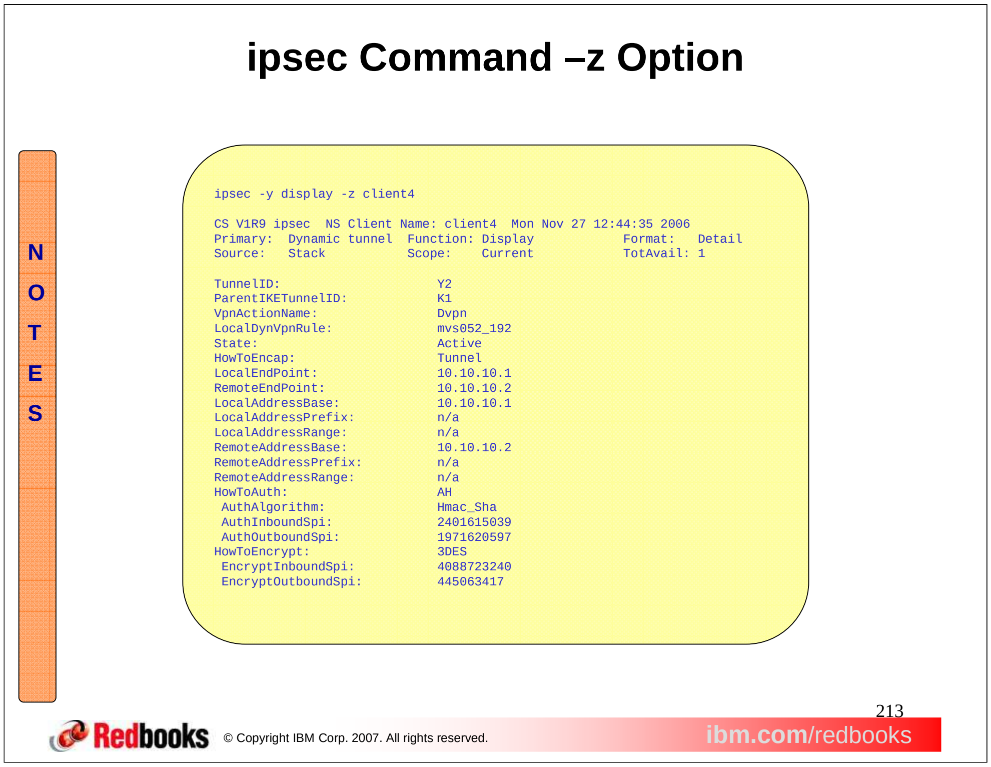
RemoteManagementSelected indicates whether or not the client is configured to use the NSS network management service

RemoteManagementEnabled displays “yes” when the client has selected this service and it is also permitted to the service per the governing SERVAUTH profile. Otherwise, “no” will appear.

CertificateServicesSelected indicates whether or not the client is configured to use the NSS certificate service

CertificateServicesEnabled displays “yes” when the client has selected this service and it is also permitted to the service per the governing SERVAUTH profile. Otherwise, “no” will appear.

The final few lines indicate attributes of the current client connection state

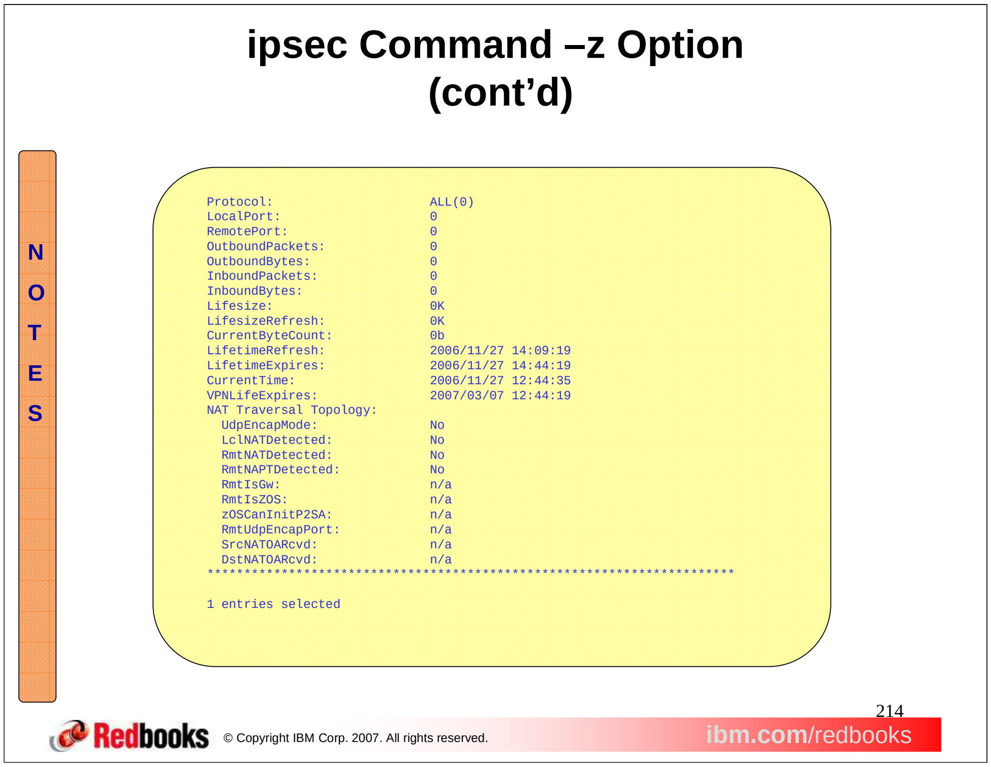


ipsec Command –z Option

N O T E S

```
ipsec -y display -z client4

CS V1R9 ipsec  NS Client Name: client4  Mon Nov 27 12:44:35 2006
Primary:  Dynamic tunnel  Function: Display            Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 1

TunnelID:                     Y2
ParentIKETunnelID:            K1
VpnActionName:                Dvpn
LocalDynVpnRule:              mvs052_192
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                10.10.10.1
RemoteEndPoint:               10.10.10.2
LocalAddressBase:             10.10.10.1
LocalAddressPrefix:           n/a
LocalAddressRange:            n/a
RemoteAddressBase:            10.10.10.2
RemoteAddressPrefix:          n/a
RemoteAddressRange:           n/a
HowToAuth:                    AH
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              2401615039
 AuthOutboundSpi:             1971620597
HowToEncrypt:                 3DES
 EncryptInboundSpi:           4088723240
 EncryptOutboundSpi:          445063417
```

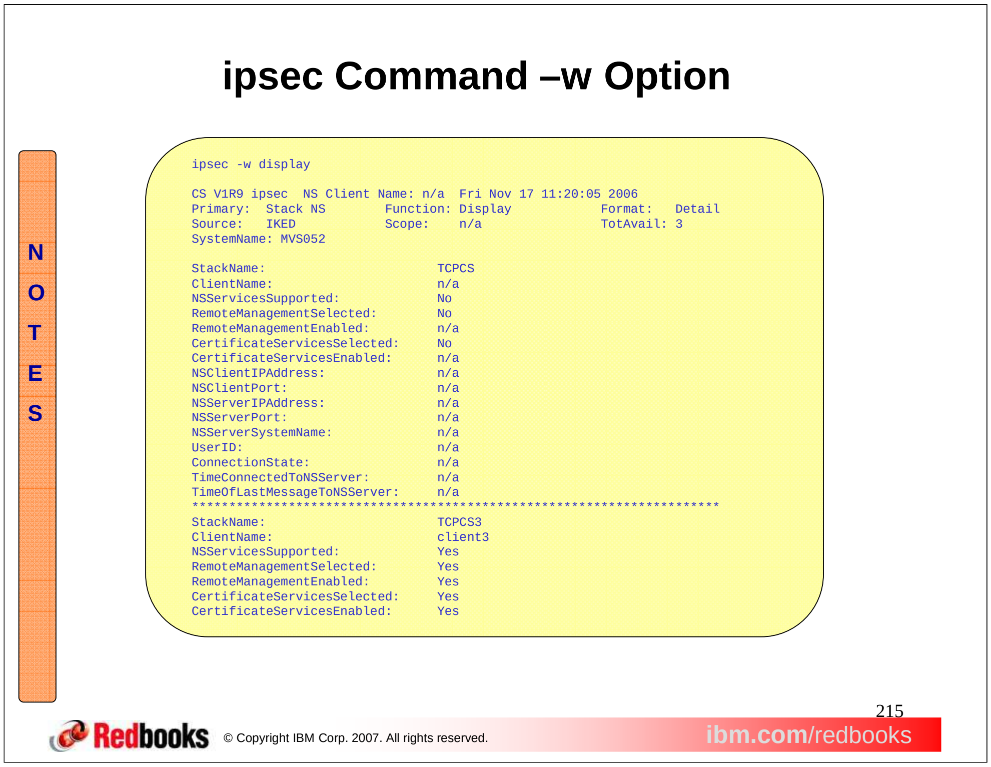
This slide shows an example using the -z option to display phase 2 security association information about the NSS client client4, where the name client4 was obtained from the previous ipsec -x display command. As you can see, the output looks exactly as it would if the same command were issued locally against the TCP/IP stack using the –p option. The only difference is in the summary header information that describes the target as an NSS client rather than simply a TCP/IP stack.



ipsec Command –z Option (cont’d)

```
Protocol:                      ALL(0)
LocalPort:                     0
RemotePort:                    0
OutboundPackets:               0
OutboundBytes:                 0
InboundPackets:                0
InboundBytes:                  0
Lifesize:                      0K
LifesizeRefresh:               0K
CurrentByteCount:              0b
LifetimeRefresh:               2006/11/27 14:09:19
LifetimeExpires:               2006/11/27 14:44:19
CurrentTime:                   2006/11/27 12:44:35
VPNLifeExpires:                2007/03/07 12:44:19
NAT Traversal Topology:
  UdpEncapMode:                No
  LclNATDetected:              No
  RmtNATDetected:              No
  RmtNAPTDetected:             No
  RmtIsGw:                     n/a
  RmtIsZOS:                    n/a
  zOSCanInitP2SA:              n/a
  RmtUdpEncapPort:             n/a
  SrcNATOARcvd:                n/a
  DstNATOARcvd:                n/a
***********************************************************************

1 entries selected
```

This slide contains the remainder of the ipsec –z output. This is the information that is normally shown when displaying phase 2 security associations.



ipsec Command –w Option

```
ipsec -w display

CS V1R9 ipsec  NS Client Name: n/a  Fri Nov 17 11:20:05 2006
Primary:  Stack NS        Function: Display            Format:   Detail
Source:   IKED            Scope:    n/a                TotAvail: 3
SystemName: MVS052

StackName:                       TCPCS
ClientName:                      n/a
NSServicesSupported:             No
RemoteManagementSelected:        No
RemoteManagementEnabled:         n/a
CertificateServicesSelected:     No
CertificateServicesEnabled:      n/a
NSClientIPAddress:               n/a
NSClientPort:                    n/a
NSServerIPAddress:               n/a
NSServerPort:                    n/a
NSServerSystemName:              n/a
UserID:                          n/a
ConnectionState:                 n/a
TimeConnectedToNSServer:         n/a
TimeOfLastMessageToNSServer:     n/a
***********************************************************************
StackName:                       TCPCS3
ClientName:                      client3
NSServicesSupported:             Yes
RemoteManagementSelected:        Yes
RemoteManagementEnabled:         Yes
CertificateServicesSelected:     Yes
CertificateServicesEnabled:      Yes
```

You can use the -w primary option on the ipsec command to query a local IKE daemon to determine which active stacks are configured as NSS clients, as well as their current status.

This slide illustrates the output of an ipsec –w command for a z/OS system that has three active TCP/IP stacks. Two of these (TCPCS3 and TCPCS4) are enabled for NSS, while the first one (TCPCS) is not. Here are a few noteworthy points regarding the output of this command:

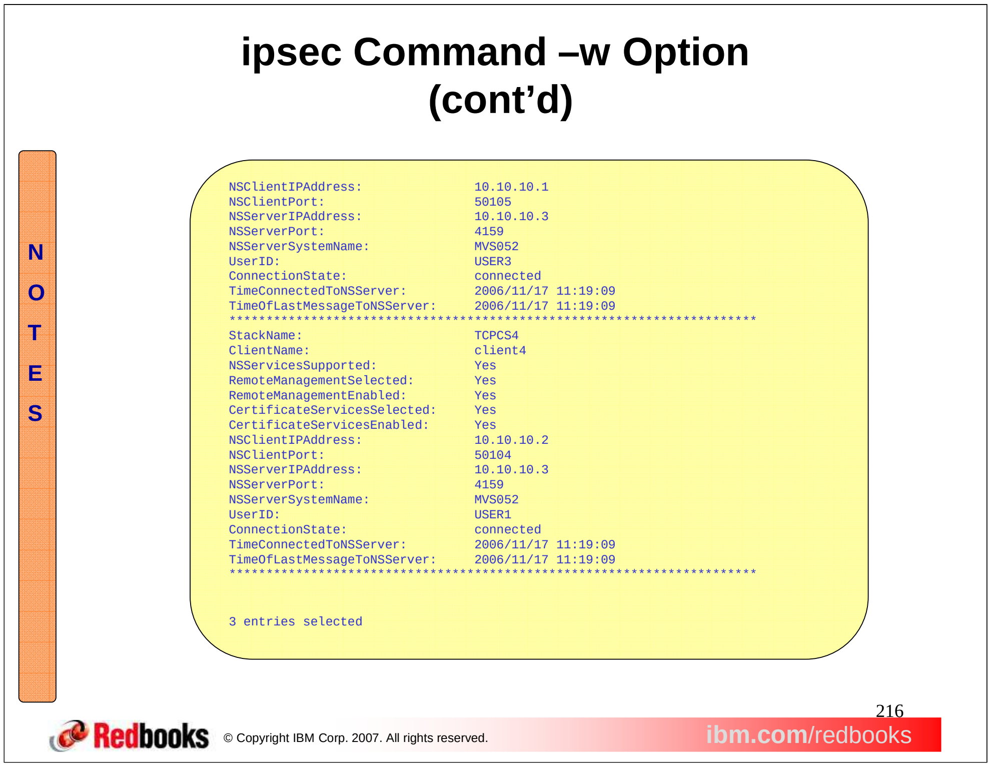
•The summary lines at the top are quite similar to that of most other ipsec command options. Nothing too exciting here.

•The first two detail lines for each stack indicate the stack identity

•The next five lines describe the client configuration as well as the services that are actually enabled (per SERVAUTH profiles at the server):

> •NSServicesSupported indicates whether or not the IKE daemon itself is configured to use NSS.
>
> •RemoteManagementSelected indicates whether or not the client is configured to use the NSS network management service
>
> •RemoteManagementEnabled displays “yes” when the client has selected this service and it is also permitted to the service per the governing SERVAUTH profile. Otherwise, “no” will appear.
>
> •CertificateServicesSelected indicates whether or not the client is configured to use the NSS certificate service
>
> •CertificateServicesEnabled displays “yes” when the client has selected this service and it is also permitted to the service per the governing SERVAUTH profile. Otherwise, “no” will appear.
>
> •The remaining lines describe the client and server addresses and indicate attributes of the current client connection state

Note that output for each stack is separated by a line of asterisks

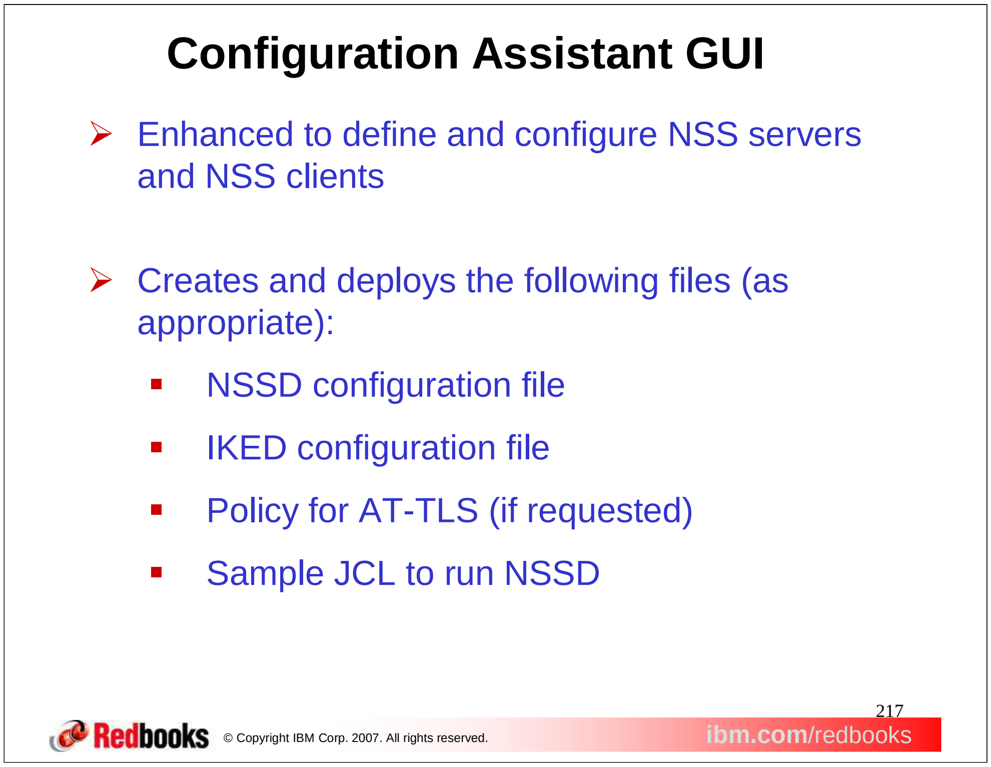


ipsec Command –w Option (cont'd)

N O T E S

```
NSClientIPAddress:              10.10.10.1
NSClientPort:                   50105
NSServerIPAddress:              10.10.10.3
NSServerPort:                   4159
NSServerSystemName:             MVS052
UserID:                         USER3
ConnectionState:                connected
TimeConnectedToNSServer:        2006/11/17 11:19:09
TimeOfLastMessageToNSServer:    2006/11/17 11:19:09
**********************************************************************
StackName:                      TCPCS4
ClientName:                     client4
NSServicesSupported:            Yes
RemoteManagementSelected:       Yes
RemoteManagementEnabled:        Yes
CertificateServicesSelected:    Yes
CertificateServicesEnabled:     Yes
NSClientIPAddress:              10.10.10.2
NSClientPort:                   50104
NSServerIPAddress:              10.10.10.3
NSServerPort:                   4159
NSServerSystemName:             MVS052
UserID:                         USER1
ConnectionState:                connected
TimeConnectedToNSServer:        2006/11/17 11:19:09
TimeOfLastMessageToNSServer:    2006/11/17 11:19:09
**********************************************************************


3 entries selected
```

This slide contains the reminder of the ipsec –w output.



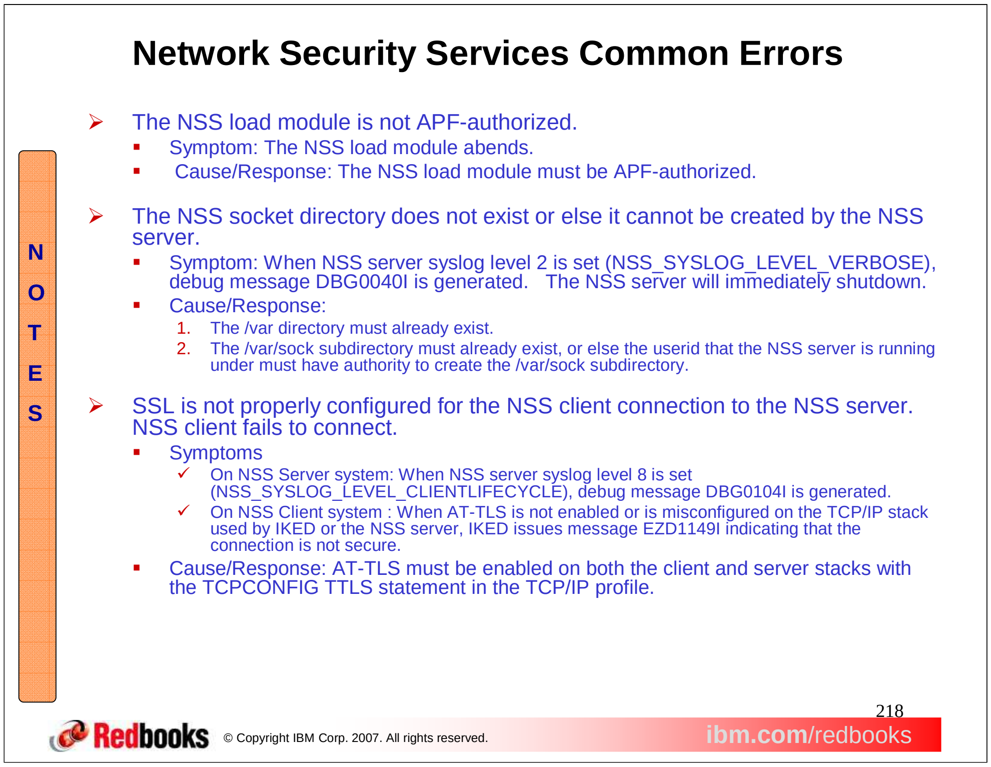
Configuration Assistant GUI

- Enhanced to define and configure NSS servers and NSS clients
- Creates and deploys the following files (as appropriate):
  - NSSD configuration file
  - IKED configuration file
  - Policy for AT-TLS (if requested)
  - Sample JCL to run NSSD

A new perspective has been added to the current list of perspectives available for the user to configure on the main panel of the Configuration Assistant GUI.

From the NSS perspective the user will be able to create images and have them be an NSS server, NSS client, or both. Stacks can also be created under an image that is either an NSS server or client. Currently the only technology to take advantage of the NSS function is IPSec. From the IPSec perspective the user can also set their NSS client image and stack settings. The design of the panels have been made in such a way to encourage the user to set as many defaults at the image level as they can and then have all of the stacks take those settings as their defaults. Each stack can override the image level defaults if they need to. The user can use either the NSS or IPSec perspective to setup NSS.

As with most other perspectives, key configuration files and excerpts of other files can be generated and deployed to target machines. This includes a sample RACF job that includes the RACF commands that are required to get all the stacks and images setup to use the NSS services.



N O T E S

Network Security Services Common Errors

- The NSS load module is not APF-authorized.
  - Symptom: The NSS load module abends.
  - Cause/Response: The NSS load module must be APF-authorized.
- The NSS socket directory does not exist or else it cannot be created by the NSS server.
  - Symptom: When NSS server syslog level 2 is set (NSS_SYSLOG_LEVEL_VERBOSE), debug message DBG0040I is generated. The NSS server will immediately shutdown.
  - Cause/Response:
    1. The /var directory must already exist.
    2. The /var/sock subdirectory must already exist, or else the userid that the NSS server is running under must have authority to create the /var/sock subdirectory.
- SSL is not properly configured for the NSS client connection to the NSS server. NSS client fails to connect.
  - Symptoms
    - ✓ On NSS Server system: When NSS server syslog level 8 is set (NSS_SYSLOG_LEVEL_CLIENTLIFECYCLE), debug message DBG0104I is generated.
    - ✓ On NSS Client system : When AT-TLS is not enabled or is misconfigured on the TCP/IP stack used by IKED or the NSS server, IKED issues message EZD1149I indicating that the connection is not secure.
  - Cause/Response: AT-TLS must be enabled on both the client and server stacks with the TCPCONFIG TTLS statement in the TCP/IP profile.

218

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

When the NSS load module is not APF-authorized, an abend occurs. The following message will be logged to the console:

IEF450I NSSD STEP1 - ABEND=S000 U4087 REASON=00000000

To APF-authorize a data set, add an APF ADD statement for the data set to a PROG*xx* member of parmlib that is used for IPL. To immediately APF-authorize the data set, use the SETPROG APF z/OS command.

You will get the following message if the /var directory does not exist or the NSSD userid does not have authority to create the /var/sock subdirectory when the NSS server syslog level 2 is set:

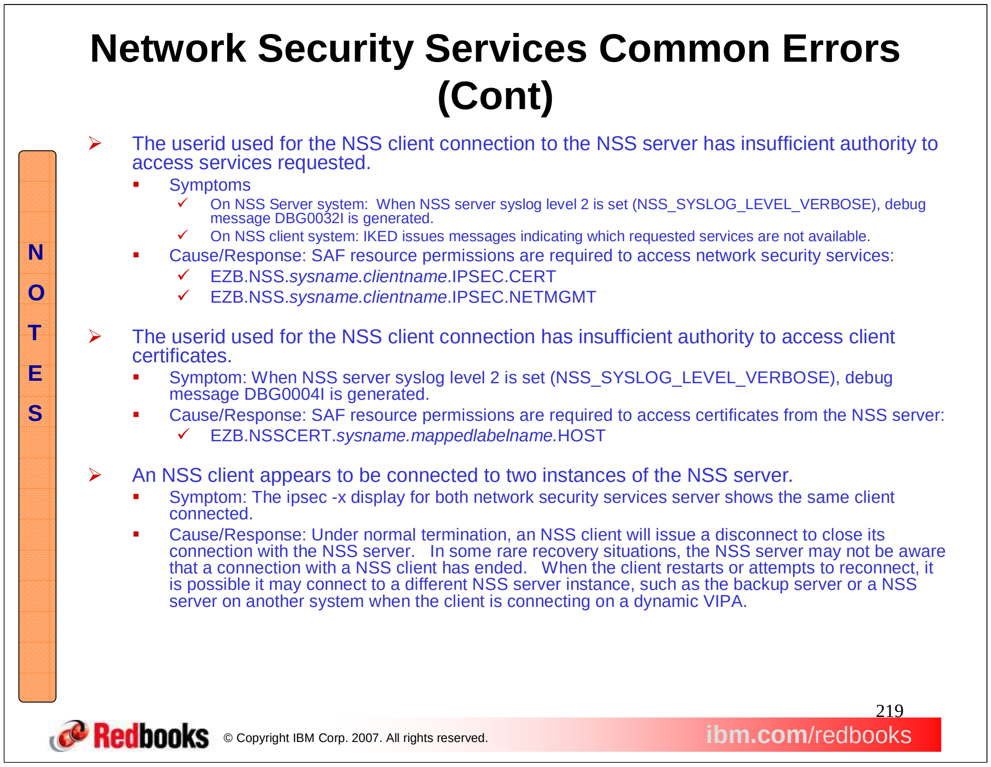
DBG0040I NSS_VERBOSE Cannot create socket directory /var/sock - rc -1 errno 135 EDC5135I Not a directory.

Write access to the /var directory is controlled through standard unix file permissions, so the userid under which NSSD runs needs to have write permissions according to those flags.

When SSL is not properly configured, you may get the following message on the NSS server system when the NSS server syslog level 8 is set:

DBG0104I NSS_LIFECYCLE NSS connID 1 - the connection is not secure - the connection will be closed

IKED, acting as the NSS client, will issue message EZD1149I indicating that the connection is not secure. AT-TLS policies must be defined for both the client and the server to secure the connection. Refer to "AT-TLS policy" in chapter 18 "Providing network security services" of the IP Configuration Guide. If AT-TLS is enabled and the definitions are configured on the client and server stacks but these errors still occur then refer to the IP Diagnosis Guide chapter 30 "Diagnosing Application Transparent Transport Layer Security (AT-TLS).“



Network Security Services Common Errors (Cont)

- The userid used for the NSS client connection to the NSS server has insufficient authority to access services requested.
  - Symptoms
    - ✓ On NSS Server system: When NSS server syslog level 2 is set (NSS_SYSLOG_LEVEL_VERBOSE), debug message DBG0032I is generated.
    - ✓ On NSS client system: IKED issues messages indicating which requested services are not available.
  - Cause/Response: SAF resource permissions are required to access network security services:
    - ✓ EZB.NSS.*sysname.clientname*.IPSEC.CERT
    - ✓ EZB.NSS.*sysname.clientname*.IPSEC.NETMGMT
- The userid used for the NSS client connection has insufficient authority to access client certificates.
  - Symptom: When NSS server syslog level 2 is set (NSS_SYSLOG_LEVEL_VERBOSE), debug message DBG0004I is generated.
  - Cause/Response: SAF resource permissions are required to access certificates from the NSS server:
    - ✓ EZB.NSSCERT.*sysname.mappedlabelname*.HOST
- An NSS client appears to be connected to two instances of the NSS server.
  - Symptom: The ipsec -x display for both network security services server shows the same client connected.
  - Cause/Response: Under normal termination, an NSS client will issue a disconnect to close its connection with the NSS server. In some rare recovery situations, the NSS server may not be aware that a connection with a NSS client has ended. When the client restarts or attempts to reconnect, it is possible it may connect to a different NSS server instance, such as the backup server or a NSS server on another system when the client is connecting on a dynamic VIPA.

219

When the SAF resources, EZB.NSS.*sysname.clientname*.IPSEC.CERT or EZB.NSS.*sysname.clientname*.IPSEC.NETMGMT, are not defined on the NSS server system or the userid of the NSS client trying to request the service has not been permitted read access to the resource then you will get a message similar to the the following, on the NSS server system, when the NSS server syslog level 2 is set:

DBG0032I NSS_VERBOSE ServauthCheck(USER2 ,EZB.NSS.MVS093.CLIENT2.IPSEC.CERT) rc 4 (DENY) racfRC 4 racfRsn 0

On the NSS client system, IKED will issue a message similar to the following messages:

EZD1145I The network security certificate service is not available for stack TCPCS2

EZD1147I The network security remote management service is not available for stack TCPCS2

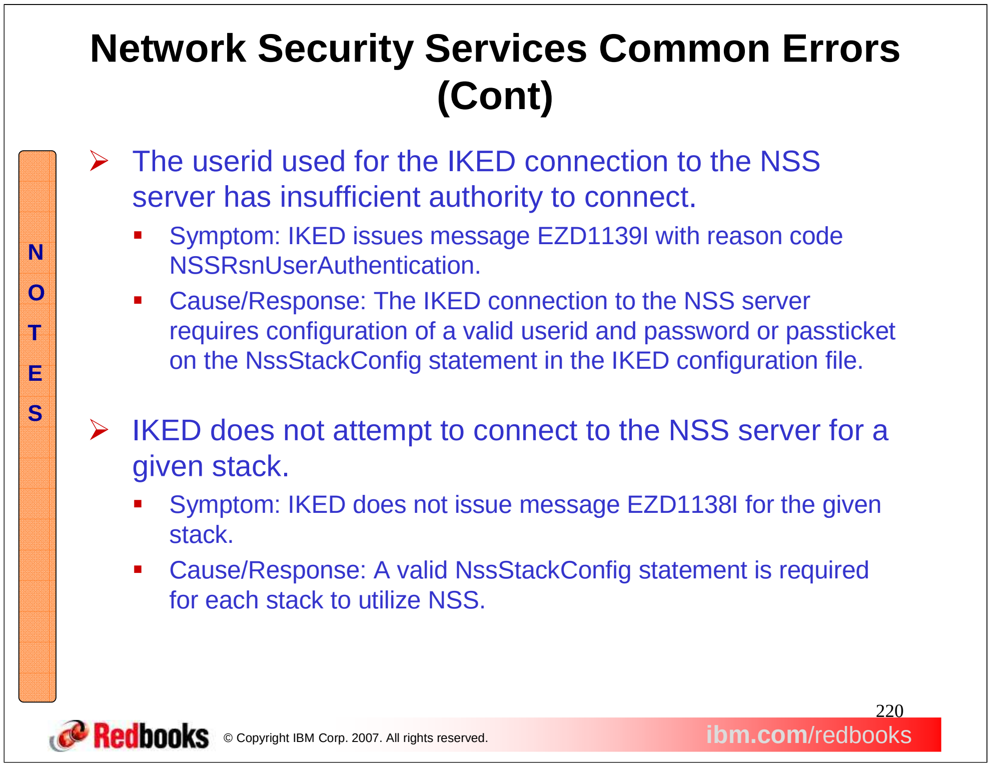
These resources must be defined on the NSS server system and the userid configured on the NssStackConfig statement in the IKED configuration file must be permitted read access to them. Refer to "Steps for authorizing resources for NSS" in chapter 18 "Providing network security services" of the IP Configuration Guide.

When the SAF resource, EZB.NSSCERT.*sysname.mappedlabelname*.HOST, is not defined or the userid of the NSS Client trying to access the certificate is not permitted read access to the resource profile then a message similar to the the following will be issued when the NSS server syslog level 2 is set:

DBG0004I NSS_CERTINFO Client MVS093_TCPCS3 connected as userid USER1 is not authorized to profile EZB.NSSCERT.VIC012.NSCLIENT3.HOST associated with matching certificate (NSCLIENT3) for request 00000000000000150000000000000000

This resource must be defined on the NSS server system and the client userid must be permitted read access to it.

If a NSS client appears to be connected to two instances of an NSS server then issue the ipsec -w display on the system running the affected NSS client to determine to which NSS server the client is actually connected. Optionally, use the netstat drop command to close out the old connection on the other NSS server.



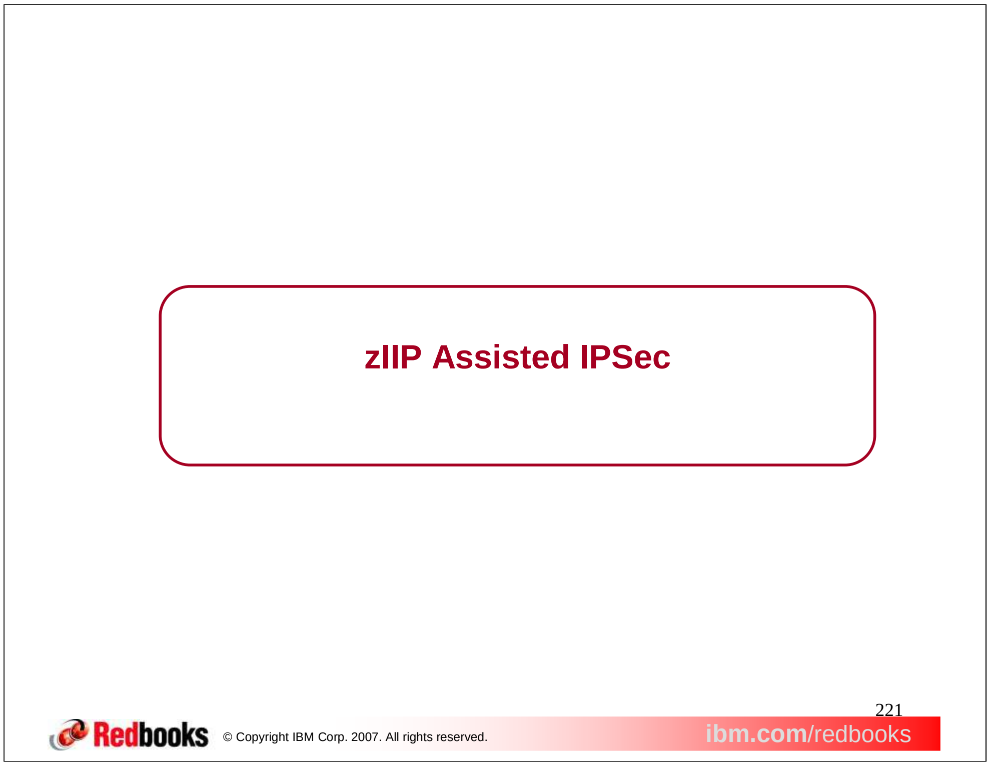
Network Security Services Common Errors (Cont)

- The userid used for the IKED connection to the NSS server has insufficient authority to connect.
  - Symptom: IKED issues message EZD1139I with reason code NSSRsnUserAuthentication.
  - Cause/Response: The IKED connection to the NSS server requires configuration of a valid userid and password or passticket on the NssStackConfig statement in the IKED configuration file.
- IKED does not attempt to connect to the NSS server for a given stack.
  - Symptom: IKED does not issue message EZD1138I for the given stack.
  - Cause/Response: A valid NssStackConfig statement is required for each stack to utilize NSS.

IKED, acting as the NSS client, issues the following message when the userid associated with the NSS client can not be authenticated.

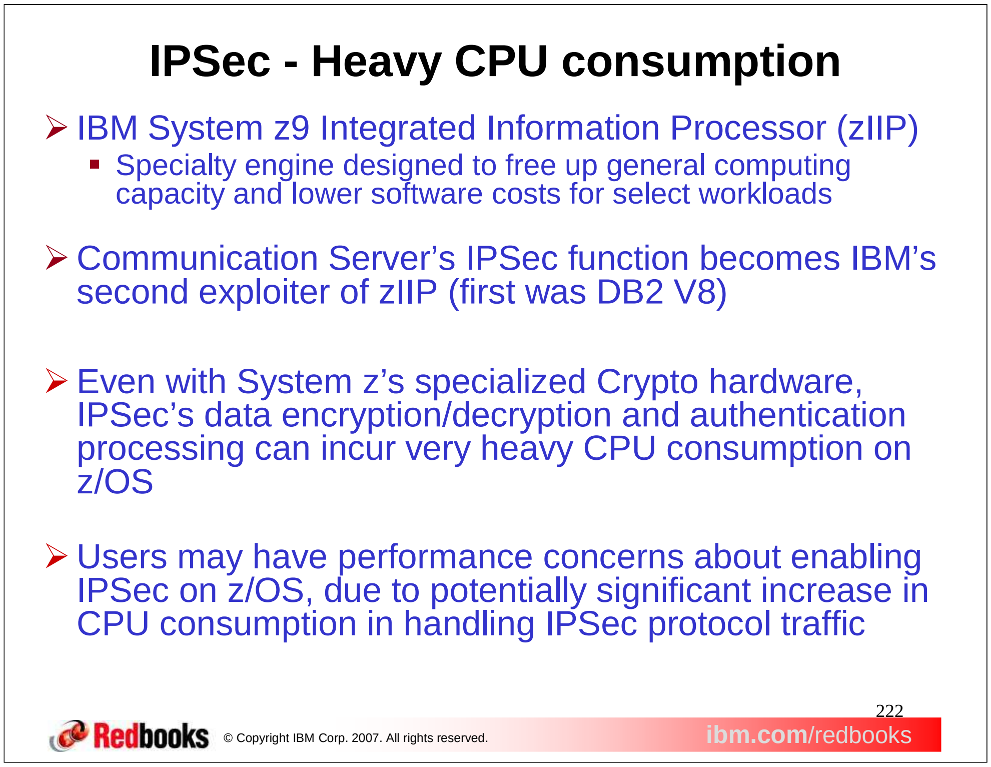
EZD1139I Request type NSS_ConnectClientReqToSrv with correlator ID 00000000000000040000000000000000 for stack TCPCS2 failed - return code EACCES reason code NSSRsnUserAuthentication

A valid NssStackConfig statement must be configured for each stack that will act as a NSS client. Refer to chapter 8 "IKE daemon" in the IP Configuration Reference for information about configuring the NssStackConfig statement.



zIIP Assisted IPSec

This section describes how CommServer makes use of the z9 Integrated Information Processor (zIIP) for IPSec protocol traffic.

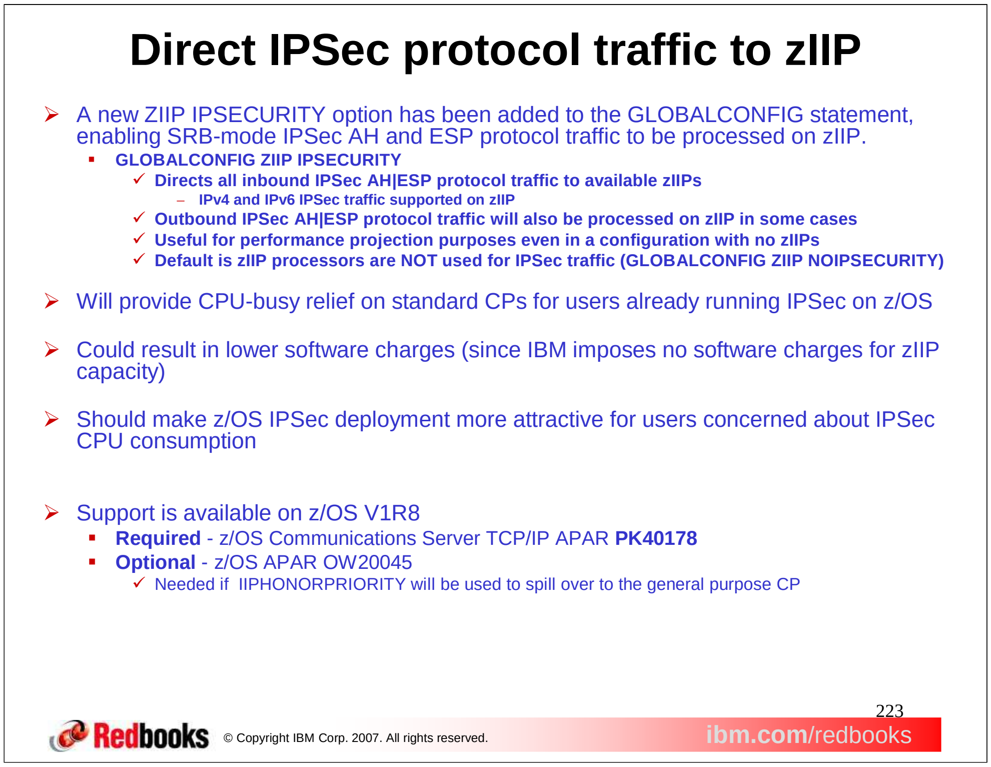


IPSec - Heavy CPU consumption

- IBM System z9 Integrated Information Processor (zIIP)
  - Specialty engine designed to free up general computing capacity and lower software costs for select workloads
- Communication Server's IPSec function becomes IBM's second exploiter of zIIP (first was DB2 V8)
- Even with System z's specialized Crypto hardware, IPSec's data encryption/decryption and authentication processing can incur very heavy CPU consumption on z/OS
- Users may have performance concerns about enabling IPSec on z/OS, due to potentially significant increase in CPU consumption in handling IPSec protocol traffic

The z9 Integrated Information Processor (zIIP) was announced in 1Q2006. At that time, IBM DB2 V8 was the only zIIP exploiter.

IPSec CPU consumption for certain types of network traffic can be very intensive. For example, securing bulk data workloads (like FTP or TSM) via IPSec can be especially CPU intensive, since IPSec CPU processing cost is relative to the amount of data being moved. The extra cycles consumed by IPSec could be problematic for users already running their z/OS LPARs at high utilization.



Direct IPSec protocol traffic to zIIP

- A new ZIIP IPSECURITY option has been added to the GLOBALCONFIG statement, enabling SRB-mode IPSec AH and ESP protocol traffic to be processed on zIIP.
  - **GLOBALCONFIG ZIIP IPSECURITY**
    - ✓ **Directs all inbound IPSec AH|ESP protocol traffic to available zIIPs**
      - **IPv4 and IPv6 IPSec traffic supported on zIIP**
    - ✓ **Outbound IPSec AH|ESP protocol traffic will also be processed on zIIP in some cases**
    - ✓ **Useful for performance projection purposes even in a configuration with no zIIPs**
    - ✓ **Default is zIIP processors are NOT used for IPSec traffic (GLOBALCONFIG ZIIP NOIPSECURITY)**
- Will provide CPU-busy relief on standard CPs for users already running IPSec on z/OS
- Could result in lower software charges (since IBM imposes no software charges for zIIP capacity)
- Should make z/OS IPSec deployment more attractive for users concerned about IPSec CPU consumption
- Support is available on z/OS V1R8
  - **Required** - z/OS Communications Server TCP/IP APAR **PK40178**
  - **Optional** - z/OS APAR OW20045
    - ✓ Needed if IIPHONORPRIORITY will be used to spill over to the general purpose CP

223

Redbooks ibm.com/redbooks

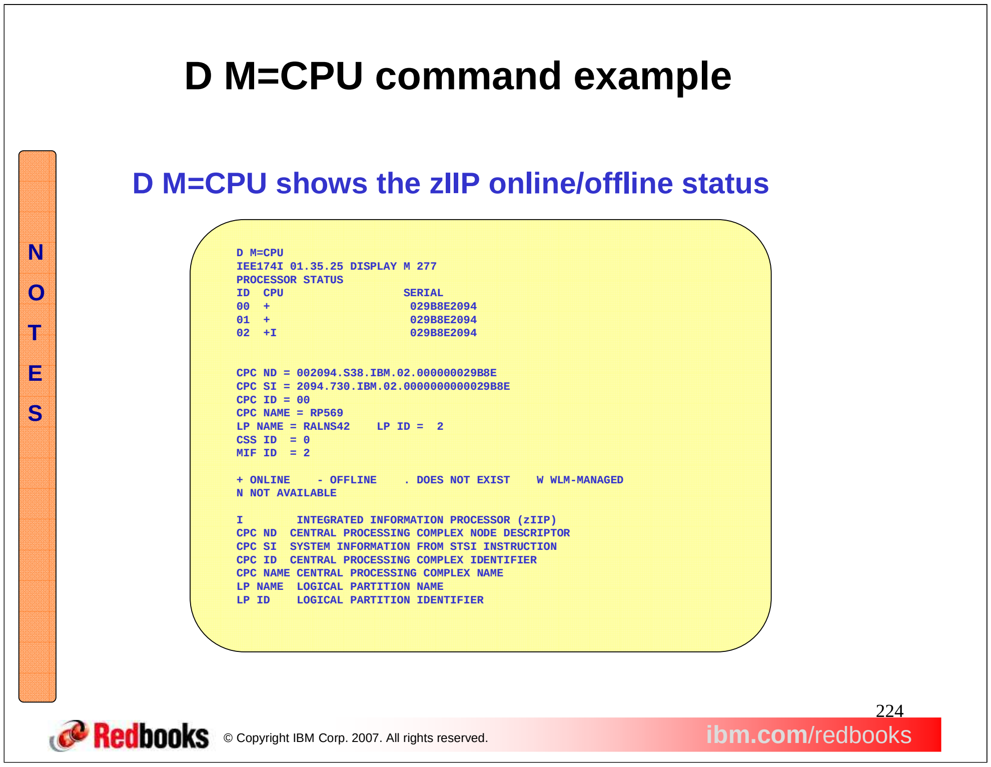
The zIIP IPSECURITY feature helps position IBM System z9 as a cost-effective server in environments requiring end-to-end security for IP network traffic. By directing IPSec's Authentication Header (AH) and Encapsulating Security Payload (ESP) protocol traffic to zIIP, your standard CPs will run less busy, and this could result in reduced software charges (since IBM imposes no software charges for zIIP capacity). Users who have decided against IPSec deployment on z/OS (due to CPU consumption issues) may find the zIIP IPSECURITY feature now makes such deployment feasible.

Configuring GLOBALCONFIG ZIIP IPSECURITY causes *inbound* ESP and AH Protocol traffic to be processed in Enclave SRBs, and targeted to available zIIPs. *Outbound* ESP and AH protocol traffic may also be processed on available zIIPs when either the application invoking the send() function is already running on a zIIP, or when the data to be transmitted is in response to normal TCP flow control (for example, data transmitted in response to a received TCP acknowledgement or window update).

Users with no zIIPs can also use GLOBALCONFIG ZIIP IPSECURITY in conjunction with the MVS PROJECTCPU function, to obtain RMF projection data on the percentage of workload that is eligible to be run on zIIP.

The default setting is to leave IPSec processing on standard CPs, so if you do want to direct your IPSec processing to zIIP, you need to code GLOBALCONFIG ZIIP IPSECURITY.

Various Netstat options are available for viewing zIIP IPSec behavior and configuration. If in doubt about zIIP online|offline status, use the MVS D M=CPU command. If viewing a dump and you're interested in zIIP IPSec configuration, you can use the TCPIPCS IPSEC or TCPIPCS PROFILE commands.



D M=CPU command example

N
O
T
E
S

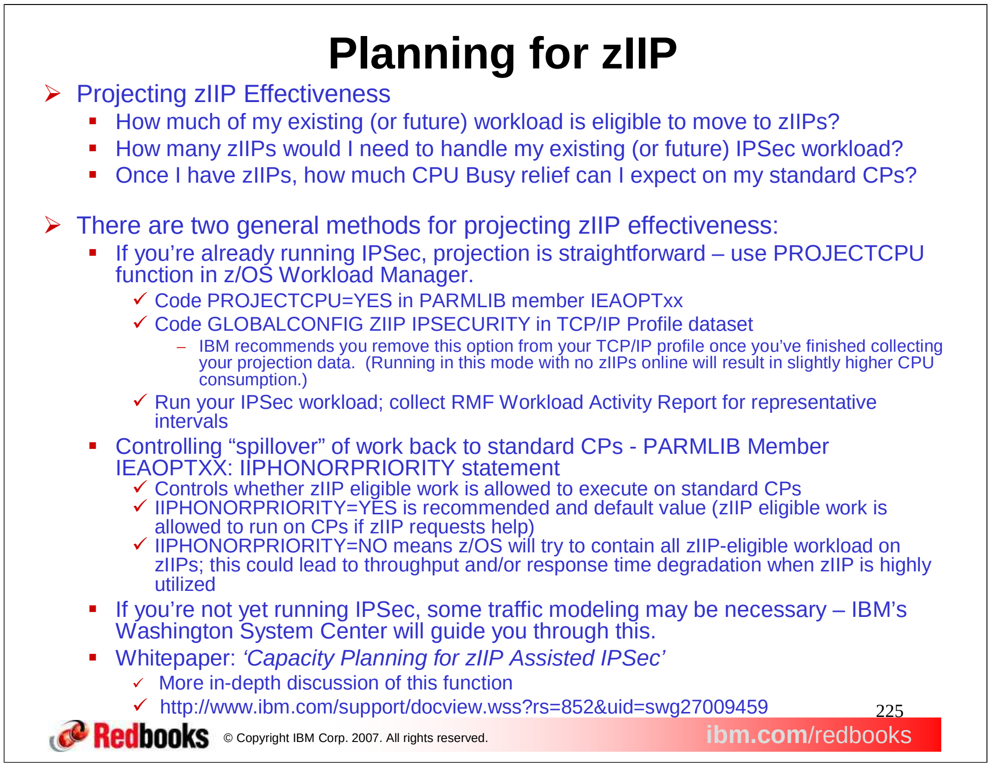
D M=CPU shows the zIIP online/offline status

```
D M=CPU
IEE174I 01.35.25 DISPLAY M 277
PROCESSOR STATUS
ID  CPU                  SERIAL
00  +                     029B8E2094
01  +                     029B8E2094
02  +I                    029B8E2094


CPC ND = 002094.S38.IBM.02.000000029B8E
CPC SI = 2094.730.IBM.02.0000000000029B8E
CPC ID = 00
CPC NAME = RP569
LP NAME = RALNS42    LP ID =  2
CSS ID  = 0
MIF ID  = 2

+ ONLINE    - OFFLINE    . DOES NOT EXIST    W WLM-MANAGED
N NOT AVAILABLE

I        INTEGRATED INFORMATION PROCESSOR (zIIP)
CPC ND  CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI  SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID  CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME  LOGICAL PARTITION NAME
LP ID    LOGICAL PARTITION IDENTIFIER
```

224

ibm.com/redbooks

Once you have a zIIP configured to your LPAR, you can use the MVS D M=CPU command to display zIIP Status. In this example, we have two standard CPs and one zIIP online. The zIIP is identified by the “I” character next to the Online|Offline status indicator.



Planning for zIIP

- ➢ Projecting zIIP Effectiveness
  - ▪ How much of my existing (or future) workload is eligible to move to zIIPs?
  - ▪ How many zIIPs would I need to handle my existing (or future) IPSec workload?
  - ▪ Once I have zIIPs, how much CPU Busy relief can I expect on my standard CPs?
- ➢ There are two general methods for projecting zIIP effectiveness:
  - ▪ If you're already running IPSec, projection is straightforward – use PROJECTCPU function in z/OS Workload Manager.
    - ✓ Code PROJECTCPU=YES in PARMLIB member IEAOPTxx
    - ✓ Code GLOBALCONFIG ZIIP IPSECURITY in TCP/IP Profile dataset
      - – IBM recommends you remove this option from your TCP/IP profile once you've finished collecting your projection data. (Running in this mode with no zIIPs online will result in slightly higher CPU consumption.)
    - ✓ Run your IPSec workload; collect RMF Workload Activity Report for representative intervals
  - ▪ Controlling "spillover" of work back to standard CPs - PARMLIB Member IEAOPTXX: IIPHONORPRIORITY statement
    - ✓ Controls whether zIIP eligible work is allowed to execute on standard CPs
    - ✓ IIPHONORPRIORITY=YES is recommended and default value (zIIP eligible work is allowed to run on CPs if zIIP requests help)
    - ✓ IIPHONORPRIORITY=NO means z/OS will try to contain all zIIP-eligible workload on zIIPs; this could lead to throughput and/or response time degradation when zIIP is highly utilized
  - ▪ If you're not yet running IPSec, some traffic modeling may be necessary – IBM's Washington System Center will guide you through this.
  - ▪ Whitepaper: *'Capacity Planning for zIIP Assisted IPSec'*
    - ✓ More in-depth discussion of this function
    - ✓ http://www.ibm.com/support/docview.wss?rs=852&uid=swg27009459

225

Redbooks

ibm.com/redbooks

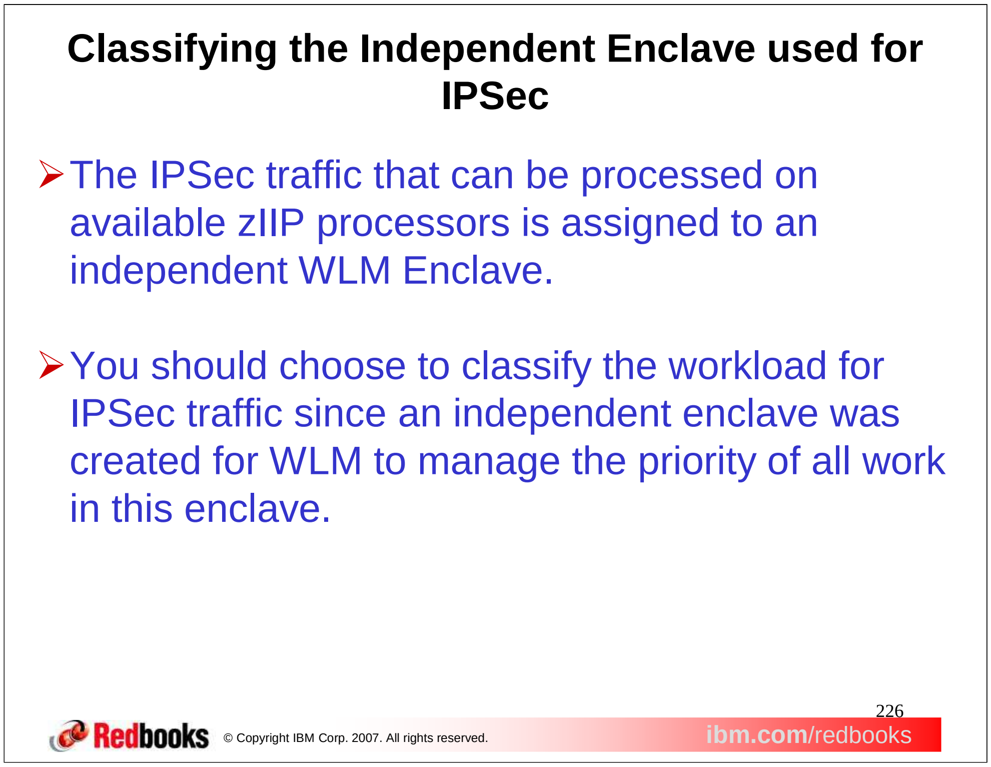
If you're running IPSec, zIIP may significantly reduce the CPU utilization of your standard CPs. In planning for zIIP, you'll need to determine (a) how much of your workload is eligible to move to zIIP, (b) how many zIIPs would be required to fully handle that load, then (c) once you do have zIIPs, how much CPU busy relief you can expect on your standard CPs.

If you're already running your representative IPSec workload, performing zIIP Projection analysis is pretty simple. Function exists within z/OS that will allow users already running IPSec (but not currently using zIIPs) to accurately project the amount of their existing workload that is eligible to move the zIIPs. This function builds upon the PROJECTCPU service present in z/OS. PROJECTCPU gives a very precise accounting of workload that is zIIP-eligible. Using PROJECTCPU for zIIP capacity planning purposes is therefore very accurate and simple, since no extra analysis of network traffic is required.

You'll want to specify GLOBALCONFIG ZIIP IPSECURITY only if (a) you already have zIIP(s) online to your LPAR or (b) you're executing performance runs to obtain RMF data to project zIIP effectiveness. Specifying IIPHONORPRIORITY=YES allows zIIP eligible workload to run on standard CPs, if zIIP work is not completed in a reasonable time period. This is the default and recommended value. Specifying IIPHONORPRIORITY=NO disallows any zIIP eligible work from running on CPs (unless no zIIPs are online, or zIIP work is holding system locks or other resources impeding non zIIP work). When the NO value is set and zIIPs are present in the configuration, zIIP eligible work will be contained on the zIIPs. During periods of very high zIIP utilization, throughput and response time may suffer. It may be reasonable to tradeoff throughput/response time in some environments, where minimizing utilization of the standard CPs is paramount.

If you're not yet running IPSec, some complex traffic modeling may be necessary to derive accurate estimates of zIIP effectiveness in your future IPSec configuration. System z sales personnel will engage the Washington System Center to perform this modeling, when necessary.

We've produced a whitepaper that covers zIIP IPSec projection modeling in depth, and also presents some of the early zIIP IPSec performance data collected in IBM labs. It can be found on ibm.com.



Classifying the Independent Enclave used for IPSec

- The IPSec traffic that can be processed on available zIIP processors is assigned to an independent WLM Enclave.
- You should choose to classify the workload for IPSec traffic since an independent enclave was created for WLM to manage the priority of all work in this enclave.

226

ibm.com/redbooks

The WLM independent Enclave is an entity that encapsulates this IPSec workload as execution units which are separately classified and managed in a WLM Service Class.

Here is a more detailed description of classifying the independent Enclave used for IPSec workload. This is performed by the following WLM Service definitions using the WLM ISPF panels:

1) Create a workload for the IPSec traffic that will be operating on the independent Enclave. From the primary WLM ISPF panel select option 2 "Workloads".

2) Create a service class that will contain an appropriate performance goals for the IPSec independent Enclave. From the primary WLM ISPF panel select option 4 "Service Classes". From this panel you will define your new service class and associate it with the workload you previously defined. When you define a BASE GOAL information for your single defined period you will choose a goal type of "Execution velocity". After this is selected then you will need to define a Velocity and Importance for the service class being defined. It is important to set an appropriate value depending on other traffic that may be competing for zIIP or General CPU resources (General CPs become a factor when you have defined the IIPHONORPRIORITY parameter located in the IEAOPTxx member of SYS1.PARMLIB to a value of YES).

3) Create a WLM "subsystem type" for TCP/IP . The subsystem type name must be specified as **TCP** and can be defined by using the WLM ISPF application. From the primary WLM ISPF panel select option 6 "Classification Rules". From this panel "Subsystem Type Selection List for Rules" you will move your cursor to the field "Subsystem-Type" and press the enter key. You will then be prompted for the type of operation you wish to perform. Since you want to create an new subsystem type you will select option 1 "Create". From this new screen "Create Rules for the Subsystem Type" you should specify the "Subsystem Type" of **TCP** and a desired description of this new subsystem type.

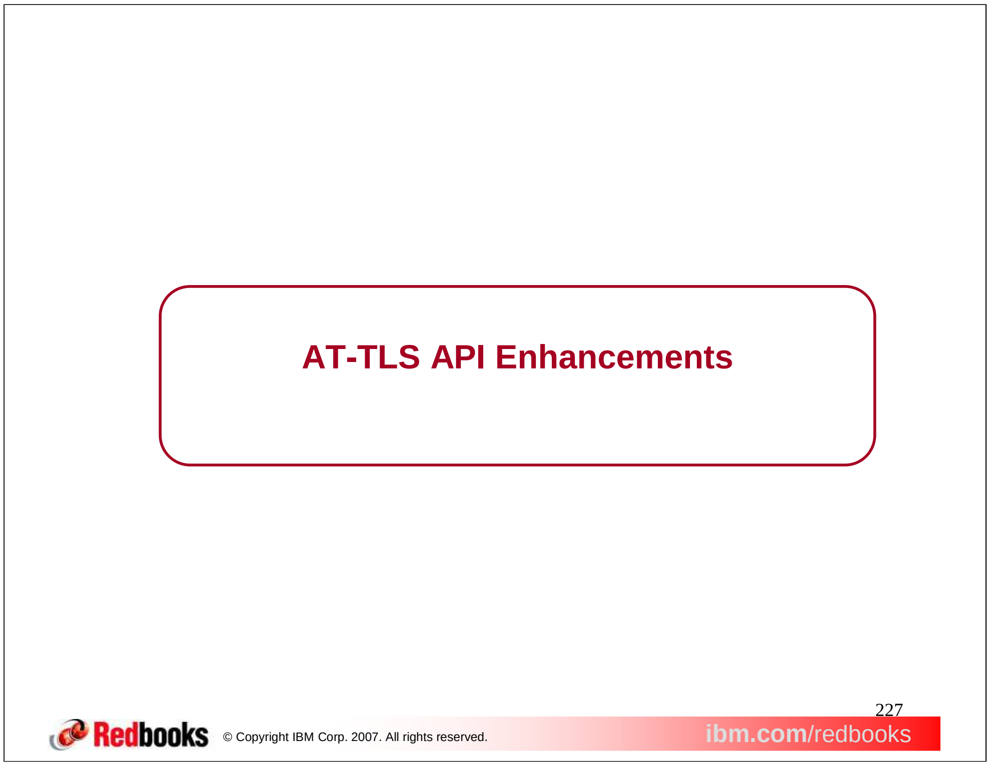
4) Create a classification rule for the created subsystem type of **TCP** by using the WLM ISPF application screen "Create Rules for the Subsystem Type". (Reach this screen using option 6 from the primary screen or you may already be in this screen after the creation of the new subsystem type). At this point, define a classification rule for the subsystem type. This rule determines what work is associated with a service class for this subsystem type. The following work qualifiers can be used for the new independent Enclave for IPSec work:

 * Transaction Name qualifier should be set to a value of **TCPENC01**

 * Subsystem Instance (SI) can be set to the TCP/IP stack's jobname. This option can be used to identify a specific stack to this WLM rule.

For more information on the additional assist for IPSec Protocol traffic that is available via the z9 Integrated Information Processor (zIIP) refer to the z/OS CS V1R9 IP Configuration guide.

For a more detailed description of defining Workload Manager (WLM) service definitions (workloads, service classifications, classification rules, subsystem type, etc.) and WLM in general refer to *z/OS MVS Planning: Workload Management* and the Redbook *System Programmer’s Guide to: Workload Manager*

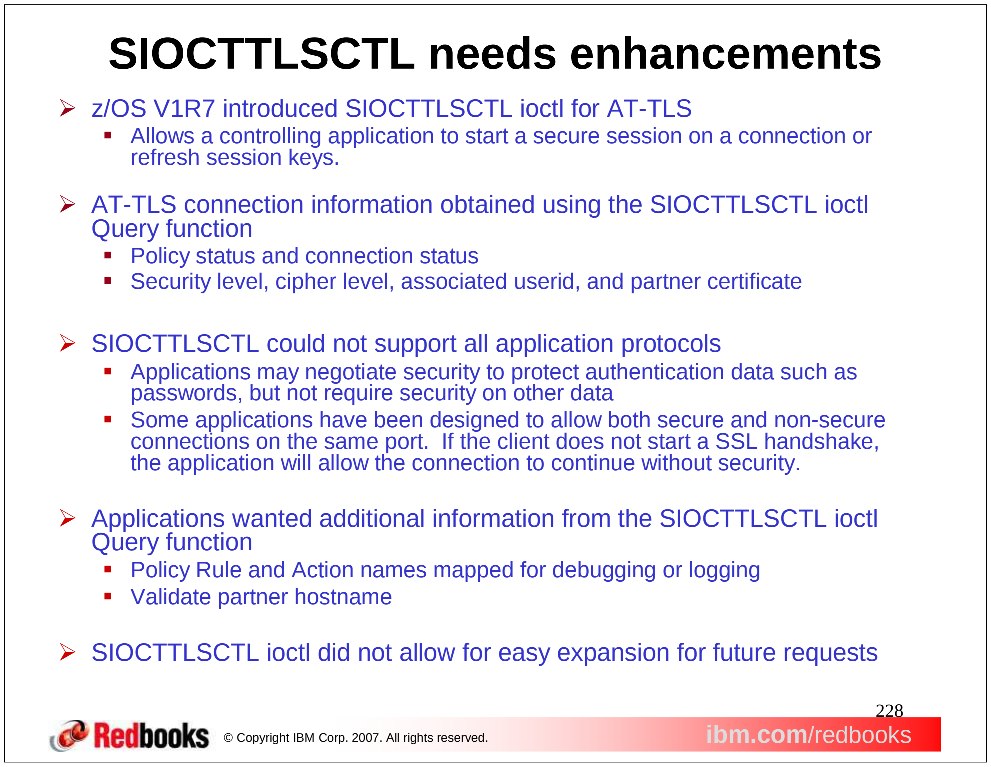


AT-TLS API Enhancements

227

ibm.com/redbooks

This section covers enhancements to the AT-TLS API using the existing SIOCTTLSCTL ioctl.



SIOCTTLSCTL needs enhancements

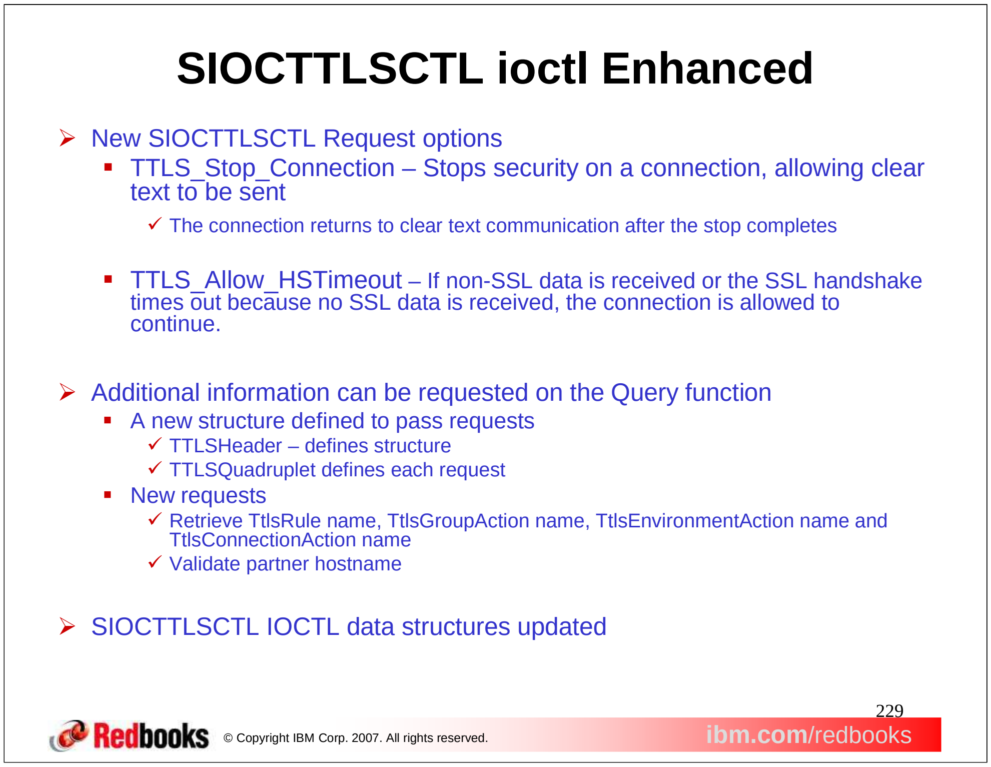
- z/OS V1R7 introduced SIOCTTLSCTL ioctl for AT-TLS
  - Allows a controlling application to start a secure session on a connection or refresh session keys.
- AT-TLS connection information obtained using the SIOCTTLSCTL ioctl Query function
  - Policy status and connection status
  - Security level, cipher level, associated userid, and partner certificate
- SIOCTTLSCTL could not support all application protocols
  - Applications may negotiate security to protect authentication data such as passwords, but not require security on other data
  - Some applications have been designed to allow both secure and non-secure connections on the same port. If the client does not start a SSL handshake, the application will allow the connection to continue without security.
- Applications wanted additional information from the SIOCTTLSCTL ioctl Query function
  - Policy Rule and Action names mapped for debugging or logging
  - Validate partner hostname
- SIOCTTLSCTL ioctl did not allow for easy expansion for future requests

z/OS 1.7 introduced Application Transparent Transport Layer Security (AT-TLS) and the SIOCTTLSCTL ioctl. This allowed applications to control AT-TLS security on a connection. The application starts security on the connection. The application can also reset the cipher being used to generate new session keys for the connection or reset the session associated with the connection to force a full SSL handshake. This type of application is called a controlling application. The AT-TLS policy must be defined with ApplicationControlled On.

The SIOCTTLSCTL ioctl currently can be used to obtain information about the connection. The state of the connection (secure, not secure, or handshake in progress) and the policy status (unknown, client, server or server with client authentication) can be obtained. For secure connections, the security level(SSLv2, SSLv3 or TLSv1) and the negotiated cipher can be obtained. For connections which the certificate has been received, the certificate and associated userid can be obtained.

Many applications use a secure connection for sensitive data during the connection. After this data exchange, security is no longer needed for the connection. The application will stop security on the connection, reducing the CPU overhead of security. Some applications also support both secure and non-secure connections on the same port. These applications detect which type of client has connected and act accordingly. These type of applications could not use the SIOCTTLSCTL ioctl to implement security.

Additional information is available about the connection using netstat. Applications can use the policy rule and action names for debugging purposes. Some applications need to validate the hostname received in the partner certificate against the hostname they have connected to as described in RFC 2818. The SIOCTTLSCTL ioctl did not allow for additional functions to be easily defined.



SIOCTTLSCTL ioctl Enhanced

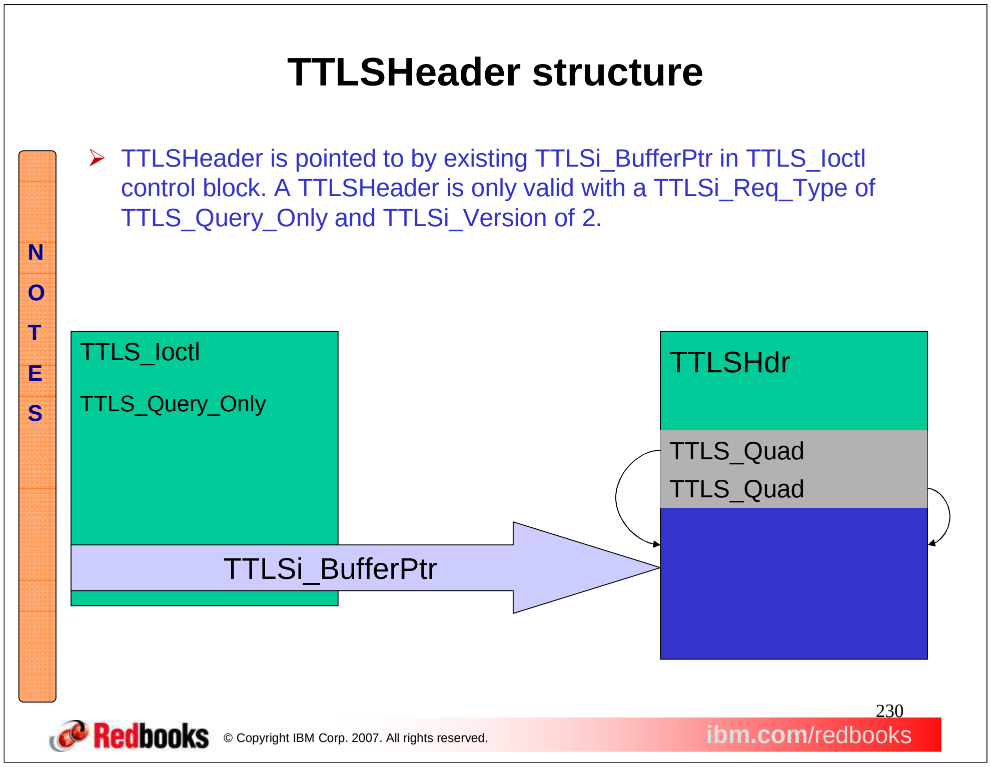
- New SIOCTTLSCTL Request options
  - TTLS_Stop_Connection – Stops security on a connection, allowing clear text to be sent
    - ✓ The connection returns to clear text communication after the stop completes
  - TTLS_Allow_HSTimeout – If non-SSL data is received or the SSL handshake times out because no SSL data is received, the connection is allowed to continue.
- Additional information can be requested on the Query function
  - A new structure defined to pass requests
    - ✓ TTLSHeader – defines structure
    - ✓ TTLSQuadruplet defines each request
  - New requests
    - ✓ Retrieve TtlsRule name, TtlsGroupAction name, TtlsEnvironmentAction name and TtlsConnectionAction name
    - ✓ Validate partner hostname
- SIOCTTLSCTL IOCTL data structures updated

Two new options now are defined for the SIOCTTLSCTL ioctl. TTLS_Stop_Connection allows the application to stop security on a connection. The SSL security on the connection will be stopped and future data will be sent as clear text. The TTLS_Stop_Connection request behaves differently for blocking and non-blocking sockets. For blocking sockets, the ioctl will return once the stop completes. For non-blocking sockets, the ioctl will return immediately with EInProgress. The application can use a select for write to determine when the stop is complete. Future data on the socket will be sent in clear text. SSLv2 does not support any type of close notification, so stop is not supported on SSLv2 connections. All application data needs to be read before the stop is issued.

TTLS_Allow_HSTimeout will allow the SSL handshake to timeout if no SSL data is received from the client or if clear text data is received. This option is only valid with TTLS_Init_Connection since it only applies to a SSL handshake on a clear text connection. The AT-TLS policy must have a non-zero HandshakeTimeout value. This is required so that the handshake will not hang indefinitely. TTLS_Allow_HSTimeout can only be used when the application is acting as the server in the SSL handshake. SSL handshakes always start with the client sending a SSL hello.

A new structure is created, the TTLSHeader. The TTLSHeader contains control information about the number of requests contained in the buffer. Each request is represented by a TTLSQuadruplet, which defines the request. The TTLSHeader is pointed to be the existing TTLSi_BufferPtr. TTLSK_TTLSRule_Name, TTLSK_TTLSGroupAction_Name, TTLSK_TTLSEnvironmentAction_Name, and TTLSK_TTLSConnectionAction_Name are the **TTLSQ_Key** values used on the ioctl to **r**etrieve policy rule and action names. The policy rule and action names will return up to a 48 character buffer with a null character terminated. TTLSK_Certificate is used to retrieve the partners certificate. It is equivalent to a TTLSi_Return_Certificate request. TTLSK_Host_Status is used to validate the hostname from the partner certificate against a hostname supplied in the TTLSHeader buffer. A hostname must be passed, pointed to by the TTLSQuadruplet. The TTLSQ_Rcode will be set upon return.

The SIOCTTLSCTL IOCTL data structures has been updated for all supported languages. Assembler (ezbztlsp.macro), C (ezbtlsc.h), Cobol (ezbztlsb.sample), PL/1 (ezbztls1.sample), and Rexx sockets (STOPCONNECTION and INITCONNHSTIMEOUT requests defined). Rexx sockets has a unique constant defined, INITCONNHSTIMEOUT, which combines the TTLS_Init_Connection and TTLS_Allow_HSTimeout options. Rexx sockets have constants for the AT-TLS Query functions. QueryRuleName, QueryGroupActionName, QueryEnvironmentActionName, and QueryConnectionActionName are used to retrieve policy rule and action names QueryHost accepts a hostname as a parameter to validate against the partner certificate. Rexx sockets do not have access to the partner certificate directly.



TTLSHeader structure

- TTLSHeader is pointed to by existing TTLSi_BufferPtr in TTLS_Ioctl control block. A TTLSHeader is only valid with a TTLSi_Req_Type of TTLS_Query_Only and TTLSi_Version of 2.

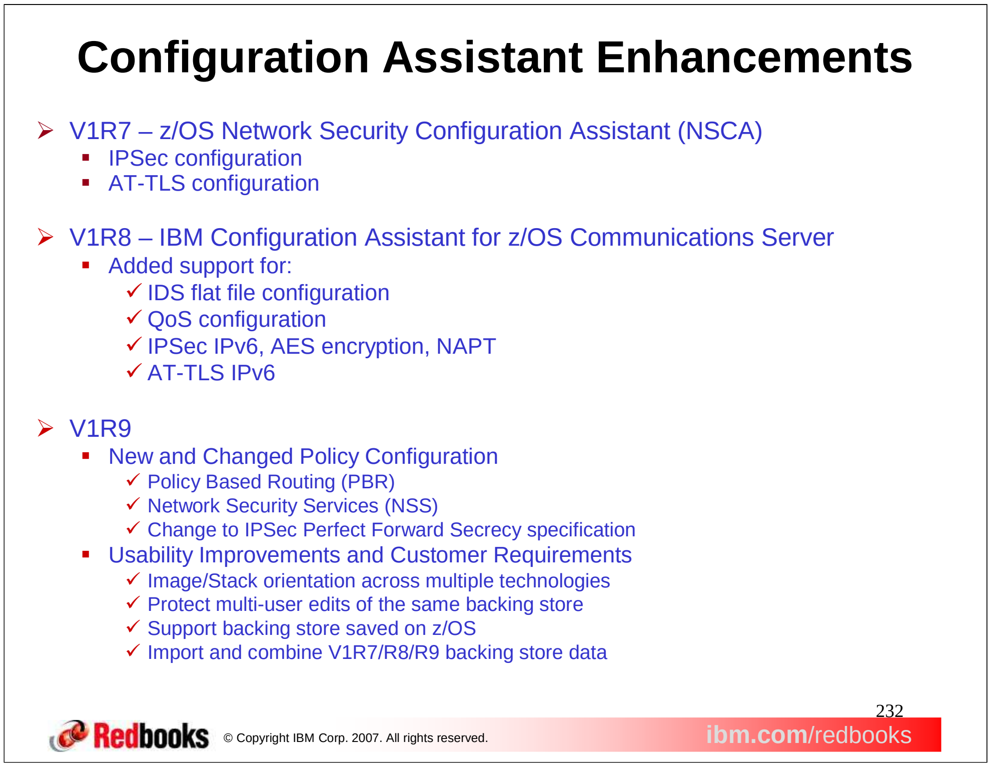
The TTLSHeader is pointed to by the existing TTLSi_BufferPtr. The version in the SIOCTTLSCTL request must be set to 2. The TTLSHeader is only supported when the TTLSi_Req_Type is set to TTLS_Query_Only(0). The TTLSHeader must be first in storage and contains the number of requests in the buffer. The TTLS_Quadruplets immediately follow the TTLSHeader. Each TTLS_Quadruplet can point into the buffer for the request. For example, the hostname to be compared against a partner certificate would be after the TTLSQuadruplet and pointed to be the TTLSQuadruplet. Upon return, the TTLSQuadruplet will be updated to point to the returned information, if any.

ibm.com

Configuration Assistant for z/OS Communications Server

International Technical Support Organization

This presentation describes the new functions in z/OS V1R9 Communications Server Configuration Assistant.



Configuration Assistant Enhancements

- V1R7 – z/OS Network Security Configuration Assistant (NSCA)
  - IPSec configuration
  - AT-TLS configuration
- V1R8 – IBM Configuration Assistant for z/OS Communications Server
  - Added support for:
    - ✓ IDS flat file configuration
    - ✓ QoS configuration
    - ✓ IPSec IPv6, AES encryption, NAPT
    - ✓ AT-TLS IPv6
- V1R9
  - New and Changed Policy Configuration
    - ✓ Policy Based Routing (PBR)
    - ✓ Network Security Services (NSS)
    - ✓ Change to IPSec Perfect Forward Secrecy specification
  - Usability Improvements and Customer Requirements
    - ✓ Image/Stack orientation across multiple technologies
    - ✓ Protect multi-user edits of the same backing store
    - ✓ Support backing store saved on z/OS
    - ✓ Import and combine V1R7/R8/R9 backing store data

232

Redbooks

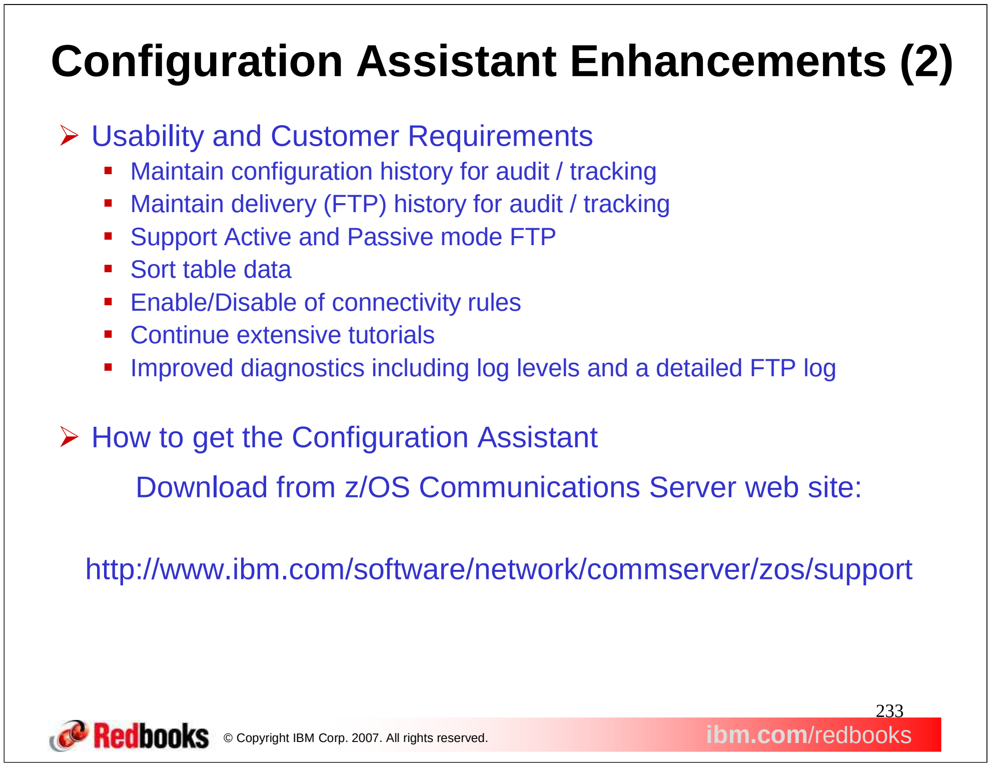
ibm.com/redbooks

The GUI was initially available for V1R7 and was named the Network Security Configuration Assistant (NSCA). For V1R8, the GUI was renamed to the Configuration Assistant since it was enhanced to configure non-security related features.

Communications Server functions can be very complicated and time-consuming to configure by manually creating configuration files. The goal of the Configuration Assistant is to enable administrators to be able to configure these functions as easily as possible without having to understand the syntax of the configuration files.

The Configuration Assistant supports configuration of many functions. This slide and the following is a list of all the enhancements made to the V1R9 Configuration Assistant.

Updates to the Configuration Assistant provide the solution for easier configuration of PBR and NSS. With the addition of these new technologies and to allow for expansion in the future, the Configuration Assistant was restructured to handle configuration of multiple technologies. Customers expressed concern about storing configuration information on the workstation as well as protecting against multiple administrators making configuration changes at the same time. These problems have been resolved by allowing the backing store files to be stored on z/OS via FTP and providing a locking mechanism to protect against multiple users making changes at the same time. The V1R8 Configuration Assistant allows for configuring one technology at a time. This prevents the ability to check for errors across technologies. The V1R9 Configuration Assistant solves this problem by allowing multiple technologies to be configured at the same time. Also it provides for the import of multiple backing store files from previous releases.



Configuration Assistant Enhancements (2)

- Usability and Customer Requirements
  - Maintain configuration history for audit / tracking
  - Maintain delivery (FTP) history for audit / tracking
  - Support Active and Passive mode FTP
  - Sort table data
  - Enable/Disable of connectivity rules
  - Continue extensive tutorials
  - Improved diagnostics including log levels and a detailed FTP log
- How to get the Configuration Assistant

Download from z/OS Communications Server web site:

http://www.ibm.com/software/network/commserver/zos/support

233

Redbooks ibm.com/redbooks

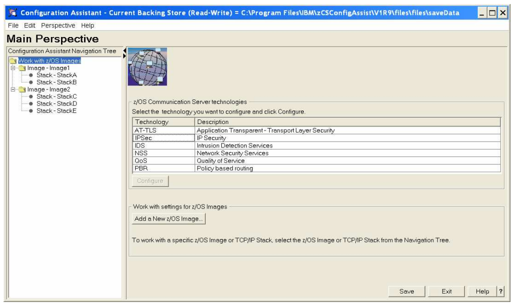
This slide is a continuation of the previous slide listing all the enhancements in the V1R9 Configuration Assistant.

Customers have asked for a way to record changes made to configurations. The solution was to add the ability to enter comments to be stored in the backing store files whenever the files are saved. Also comments can now be added to the policy files when the files are delivered using FTP. To resolve problems with firewalls preventing the FTP of configuration files, the Configuration Assistant now supports both Active and Passive mode FTP. Most tables in the Configuration Assistant can now be sorted allowing for easier navigation within large tables. When customers wanted to make a quick configuration change temporarily and then revert back to the original configuration, they needed to remove configuration rules and then re-key them when reverting to the original configuration. In the V1R9 Configuration Assistant this is resolved by allowing rules to be disabled. When the customer wants to revert to the original configuration, this is easily done by re-enabling the rules. Customers often comment about how much the tutorials have helped them. The Configuration Assistant includes new tutorials for PBR and NSS. To allow for faster resolution of customer problems, the V1R9 Configuration Assistant provides more detailed logging including a separate log for FTP connections.

The Configuration Assistant is available from the Download section of the Communications Server support web site. Separate versions are available for V1R9, V1R8 and V1R7. Support is provided on a ‘best effort’ basis from the Communications Server newsgroup at news://news.software.ibm.com/ibm.software.commserver.os390.ip

Image/Stack Orientation

- The new look is centered around the images and stacks to be configured



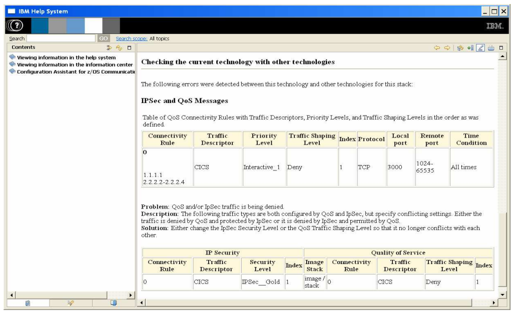
Configuration Assistant - Current Backing Store (Read-Write) = C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData
File Edit Perspective Help
Main Perspective
Configuration Assistant Navigation Tree
Work with z/OS Images
Image - Image1
Stack - StackA
Stack - StackB
Image - Image2
Stack - StackC
Stack - StackD
Stack - StackE
z/OS Communication Server technologies
Configure
Work with settings for z/OS Images
Add a New z/OS Image...
Save
Exit
Help
?

Select the technology you want to configure and click Configure.

Technology	Description
AT-TLS	Application Transparent - Transport Layer Security
IPSec	IP Security
IDS	Intrusion Detection Services
NSS	Network Security Services
QoS	Quality of Service
PBR	Policy based routing

To work with a specific z/OS Image or TCP/IP Stack, select the z/OS Image or TCP/IP Stack from the Navigation Tree.

- Allows multiple technologies to be configured in one session enabling Health Check across technologies.



IBM Help System
IBM.
Search
GO
Search scope: All topics
Contents
Viewing information in the help system
Viewing information in the information center
Configuration Assistant for z/OS Communicatic

Checking the current technology with other technologies

The following errors were detected between this technology and other technologies for this stack:

IPSec and QoS Messages

Table of QoS Connectivity Rules with Traffic Descriptors, Priority Levels, and Traffic Shaping Levels in the order as was defined.

Connectivity Rule	Traffic Descriptor	Priority Level	Traffic Shaping Level	Index	Protocol	Local port	Remote port	Time Condition
0 1.1.1.1 2.2.2.2-2.2.2.4	CICS	Interactive_1	Deny	1	TCP	3000	1024-65535	All times

Problem: QoS and/or IpSec traffic is being denied.
Description: The following traffic types are both configured by QoS and IpSec, but specify conflicting settings. Either the traffic is denied by QoS and protected by IpSec or it is denied by IpSec and permitted by QoS.
Solution: Either change the IpSec Security Level or the QoS Traffic Shaping Level so that it no longer conflicts with each other.

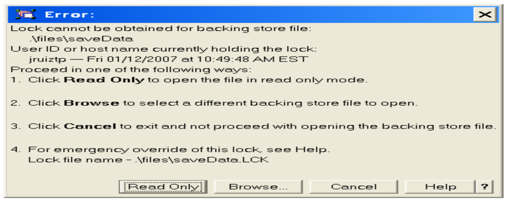
IP Security				Quality of Service				
Connectivity Rule	Traffic Descriptor	Security Level	Index	Image Stack	Connectivity Rule	Traffic Descriptor	Traffic Shaping Level	Index
0	CICS	IPSec__Gold	1	image / stack	0	CICS	Deny	1

The user's images and stacks can be defined once and are then available when configuring any of the supported technologies (AT-TLS, IPSec, IDS, NSS, QoS, and PBR).

Since the Configuration Assistant allows for multiple technologies to be configured in a single session, the Health Check feature is able to check for errors and inconsistencies between the configuration of multiple technologies. For example, in this screen shot Health Check is warning that the user has configured QoS to Deny traffic that IPSec is configured to protect.

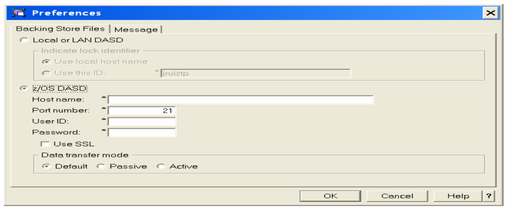
Backing Store Files

- Protect multi-user edits
  - Locking mechanism to help prevent multiple users from modifying the same backing store file at the same time.
- Support backing store files saved on z/OS
  - Allow backing store files to be stored on z/OS DASD via FTP directly from the Configuration Assistant
- Import files
  - Multiple V1R7 and V1R8 backing store files can be imported into a single V1R9 backing store file.

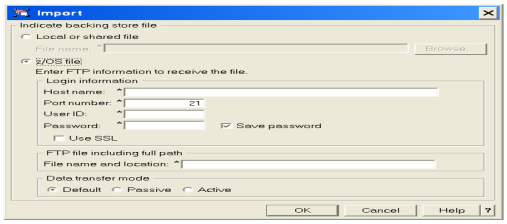


Lock cannot be obtained for backing store file:
.\files\saveData
User ID or host name currently holding the lock:
jruiztp — Fri 01/12/2007 at 10:49:48 AM EST
Proceed in one of the following ways:

1. Click **Read Only** to open the file in read only mode.
2. Click **Browse** to select a different backing store file to open.
3. Click **Cancel** to exit and not proceed with opening the backing store file.
4. For emergency override of this lock, see Help.
   Lock file name - .\files\saveData.LCK



○ Local or LAN DASD
Indicate lock identifier
● Use local host name
○ Use this ID: * jruiztp
● z/OS DASD
Host name: *
Port number: * 21
User ID: *
Password: *
□ Use SSL
Data transfer mode
● Default ○ Passive ○ Active



Indicate backing store file
○ Local or shared file
File name: *
● z/OS file
Enter FTP information to receive the file.
Login information
Host name: *
Port number: * 21
User ID: *
Password: * ☑ Save password
□ Use SSL
FTP file including full path
File name and location: *
Data transfer mode
● Default ○ Passive ○ Active

235

Redbooks ibm.com/redbooks

The Configuration Assistant saves the user's configuration data in a file called the backing store file.

When a backing store file is opened in the Configuration Assistant, a lock file is created which contains a lock ID value, and the date and time at which the lock file was created. When you specify "Local or LAN DASD" as your preference, you may create a lock ID value of your own or use the host name of the workstation which is running the Configuration Assistant. When you specify "z/OS DASD" as your preference, the lock ID value is the user id that is used to establish the FTP connection. If needed, the locking mechanism can be circumvented by manually deleting the lock file.

This file locking function has also been added to the V1R7 and V1R8 GUIs.

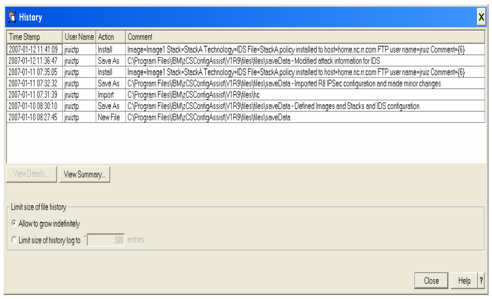
Users can manage different sets of configuration information by keeping them in different backing store files. In V1R9, backing store files can now be stored on z/OS DASD using FTP as well on the local file system. This allows for easier sharing of backing store files between multiple users.

Also added support for both Active and Passive mode FTP.

Multiple backing store files can be imported in to a single V1R9 backing store file. This can be done for backing store files stored on the local file system or on a z/OS DASD. This is especially useful to run Health Check across multiple technologies.

Maintain History for Backing Store File

- ➢ Maintain configuration history for audit / tracking
  - ▪ Automatic backing store history kept when backing store file is created and saved, when a file is imported, and when flat file configuration was transferred via FTP



History

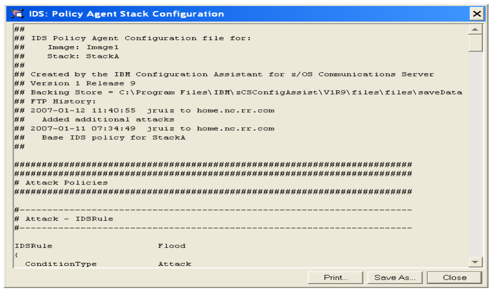
Time Stamp	User Name	Action	Comment
2007-01-12 11:41:09	jruiztp	Install	Image=Image1 Stack=StackA Technology=IDS File=StackA.policy installed to host=home.nc.rr.com FTP user name=jruiz Comment={6}
2007-01-12 11:36:47	jruiztp	Save As	C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData - Modified attack information for IDS
2007-01-11 07:35:05	jruiztp	Install	Image=Image1 Stack=StackA Technology=IDS File=StackA.policy installed to host=home.nc.rr.com FTP user name=jruiz Comment={6}
2007-01-11 07:32:32	jruiztp	Save As	C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData - Imported R8 IPSec configuration and made minor changes
2007-01-11 07:31:39	jruiztp	Import	C:\Program Files\IBM\zCSConfigAssist\V1R9\files\hc
2007-01-10 08:30:10	jruiztp	Save As	C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData - Defined Images and Stacks and IDS configuration
2007-01-10 08:27:45	jruiztp	New File	C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData

Limit size of file history

● Allow to grow indefinitely

○ Limit size of history log to * 300 entries

- ➢ Maintain delivery (FTP) history for audit / tracking
  - ▪ FTP history information is also stored as comments in the prolog of the configuration flat files.



IDS: Policy Agent Stack Configuration

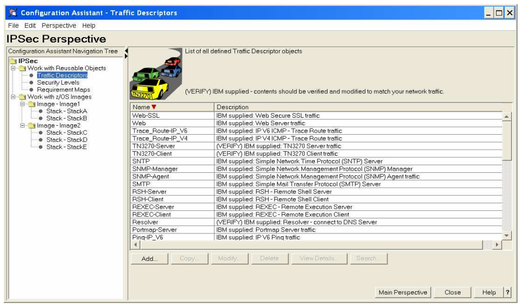
```
##
## IDS Policy Agent Configuration file for:
##     Image: Image1
##     Stack: StackA
##
## Created by the IBM Configuration Assistant for z/OS Communications Server
## Version 1 Release 9
## Backing Store = C:\Program Files\IBM\zCSConfigAssist\V1R9\files\files\saveData
## FTP History:
## 2007-01-12 11:40:55  jruiz to home.nc.rr.com
##    Added additional attacks
## 2007-01-11 07:34:49  jruiz to home.nc.rr.com
##    Base IDS policy for StackA
##

###########################################################################
###########################################################################
# Attack Policies
###########################################################################

#--------------------------------------------------------------------------
# Attack - IDSRule
#--------------------------------------------------------------------------

IDSRule                    Flood
{
  ConditionType            Attack
```

A common customer requirement was to allow a user to add comments about the changes made to a configuration. Users can now add comments whenever they save a backing store file or FTP the flat file configuration. Here’s an example showing user comments during save and FTP as well as automatic history entries when the file was created and another backing store file was imported.

The FTP history information is saved as comments in the configuration flat file and corresponds to the history information of the backing store file.

Other Enhancements

- Table Sorting
  - Column sort added to most tables e.g. traffic descriptors

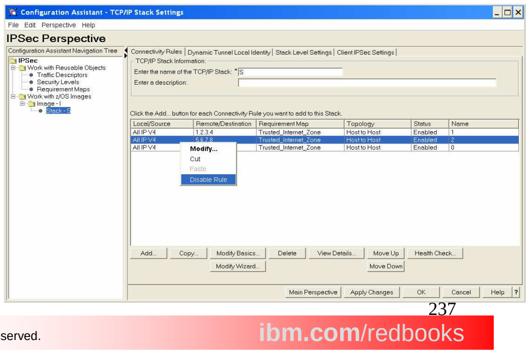


List of all defined Traffic Descriptor objects

(VERIFY) IBM supplied - contents should be verified and modified to match your network traffic.

Name ▼	Description
Web-SSL	IBM supplied: Web Secure SSL traffic
Web	IBM supplied: Web Server traffic
Trace_Route-IP_V6	IBM supplied: IP V6 ICMP - Trace Route traffic
Trace_Route-IP_V4	IBM supplied: IP V4 ICMP - Trace Route traffic
TN3270-Server	(VERIFY) IBM supplied: TN3270 Server traffic
TN3270-Client	(VERIFY) IBM supplied: TN3270 Client traffic
SNTP	IBM supplied: Simple Network Time Protocol (SNTP) Server
SNMP-Manager	IBM supplied: Simple Network Management Protocol (SNMP) Manager
SNMP-Agent	IBM supplied: Simple Network Management Protocol (SNMP) Agent traffic
SMTP	IBM supplied: Simple Mail Transfer Protocol (SMTP) Server
RSH-Server	IBM supplied: RSH - Remote Shell Server
RSH-Client	IBM supplied: RSH - Remote Shell Client
REXEC-Server	IBM supplied: REXEC - Remote Execution Server
REXEC-Client	IBM supplied: REXEC - Remote Execution Client
Resolver	(VERIFY) IBM supplied: Resolver - connect to DNS Server
Portmap-Server	IBM supplied: Portmap Server traffic
Ping-IP_V6	IBM supplied: IP V6 Ping traffic

- Enable/Disable Rules
  - Context menu to enable/disable connectivity rules



TCP/IP Stack Information:

Enter the name of the TCP/IP Stack: * S

Enter a description:

Click the Add... button for each Connectivity Rule you want to add to this Stack.

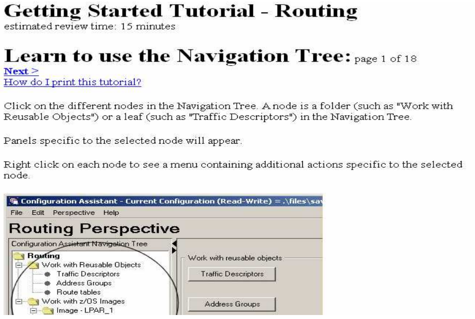
Local/Source	Remote/Destination	Requirement Map	Topology	Status	Name
All IP V4	1.2.3.4	Trusted_Internet_Zone	Host to Host	Enabled	1
All IP V4	5.6.7.8	Trusted_Internet_Zone	Host to Host	Enabled	2
All IP V4		Trusted_Internet_Zone	Host to Host	Enabled	0

Column sort allows you to sort most tables within the Configuration Assistant by clicking on the column header you want to sort by. The sorting toggles between alphabetic ascending, alphabetic descending and the default sort.

This solves the customer requirement to allow for disabling rules temporarily. Individual rules can be disabled without removing them from the configuration. When needed, the rules can be enabled to return to the original configuration.

Built-in Tutorials & Diagnosis

- Picture based tutorials built-in
- Separate Getting Started Tutorials for each technology



Getting Started Tutorial - Routing
estimated review time: 15 minutes

Learn to use the Navigation Tree: page 1 of 18
Next >
How do I print this tutorial?

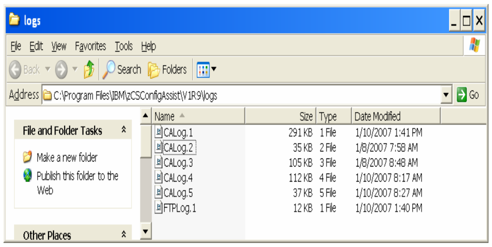
Click on the different nodes in the Navigation Tree. A node is a folder (such as "Work with Reusable Objects") or a leaf (such as "Traffic Descriptors") in the Navigation Tree.

Panels specific to the selected node will appear.

Right click on each node to see a menu containing additional actions specific to the selected node.

Configuration Assistant - Current Configuration (Read-Write) = .\files\sa
File
Edit
Perspective
Help
Routing Perspective
Configuration Assistant Navigation Tree
Routing
Work with Reusable Objects
Traffic Descriptors
Address Groups
Route tables
Work with z/OS Images
Image - LPAR_1
Work with reusable objects
Traffic Descriptors
Address Groups

- Traces are always running in the GUI
  - Last 5 trace files kept in logs directory where GUI is installed
  - Separate trace files kept for each FTP session



Address C:\Program Files\IBM\zCSConfigAssist\V1R9\logs

Name	Size	Type	Date Modified
CALog.1	291 KB	1 File	1/10/2007 1:41 PM
CALog.2	35 KB	2 File	1/8/2007 7:58 AM
CALog.3	105 KB	3 File	1/8/2007 8:48 AM
CALog.4	112 KB	4 File	1/10/2007 8:17 AM
CALog.5	37 KB	5 File	1/10/2007 8:27 AM
FTPLog.1	12 KB	1 File	1/10/2007 1:40 PM

The V1R7 and V1R8 GUIs include extensive tutorials. In V1R9, the Configuration Assistant adds tutorials for learning to configure Policy Based Routing and Network Security Services.

Trace information is always available in the CALog.n files where n increments between 1 and 5. The traces are stored in the logs directory where the GUI is installed, the default being C:\Program Files\IBM\zCSConfigAssisV1R9\logs. Sort by “Date Modified” to determine the log file you need. Detailed trace information about FTP sessions is stored separately in FTPLog.n files where n increments between 1 and 5.

The ca.properties file located in the .\files directory can be modified to specify various log settings. This would typically be used to provide additional information for debugging specific problems.

ibm.com

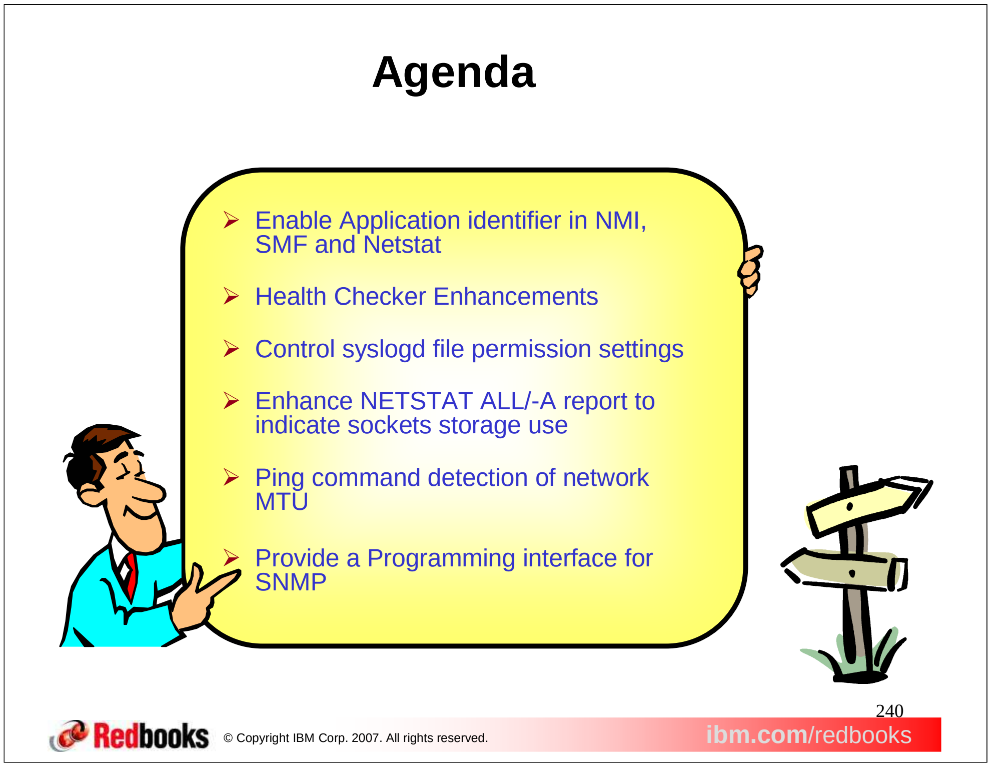
e-business

Systems Management

Redbooks

International Technical Support Organization

This presentation describes the changes made in z/OS V1R9 Communications Server in the area of systems management.

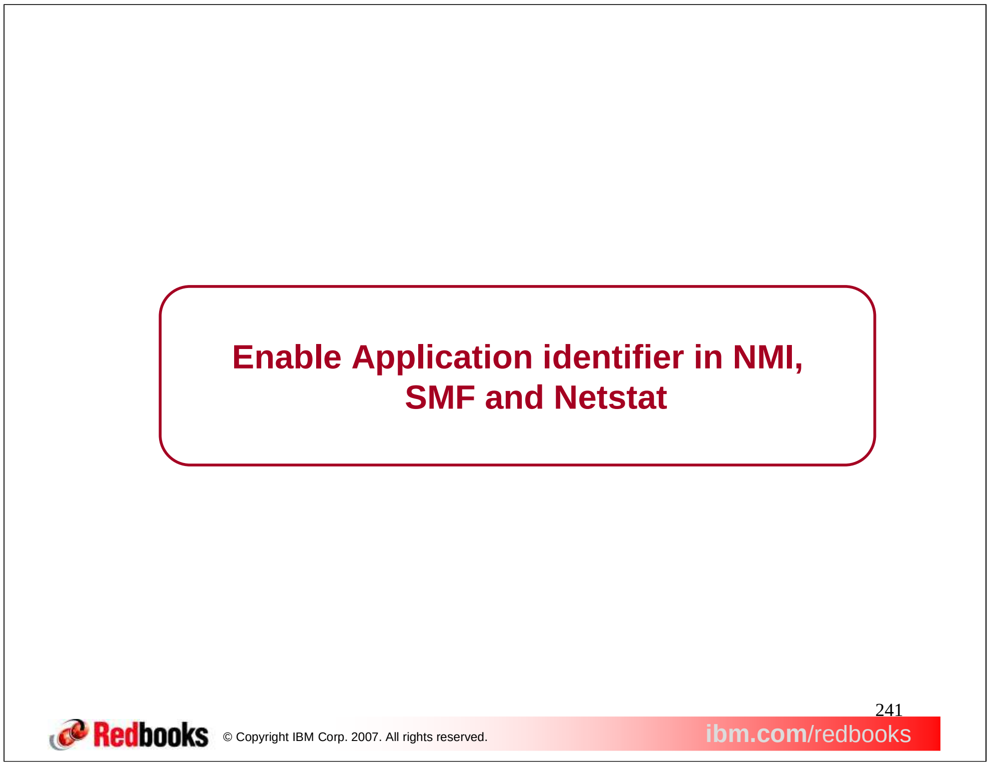


Agenda

- Enable Application identifier in NMI, SMF and Netstat
- Health Checker Enhancements
- Control syslogd file permission settings
- Enhance NETSTAT ALL/-A report to indicate sockets storage use
- Ping command detection of network MTU
- Provide a Programming interface for SNMP

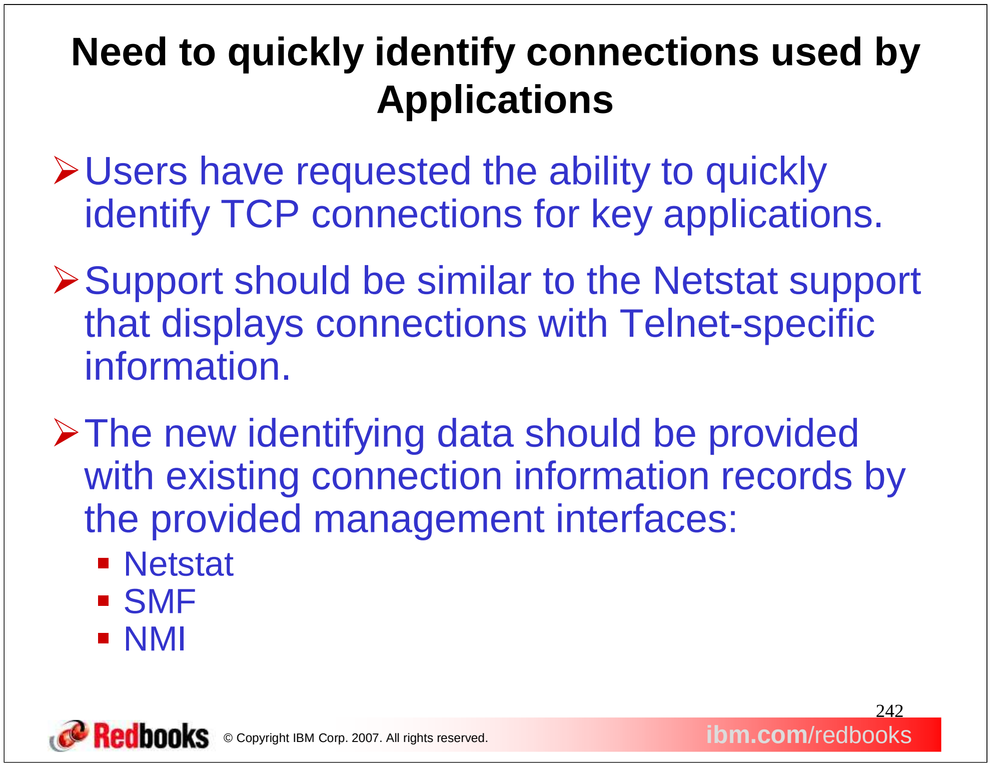
z/OS V1R9 Communications Server has several enhancements in the area of systems management. We will be discussing all of these in detail.

We’ll talk about the new function to allow applications to store identifying data on TCP connections.



Enable Application identifier in NMI, SMF and Netstat

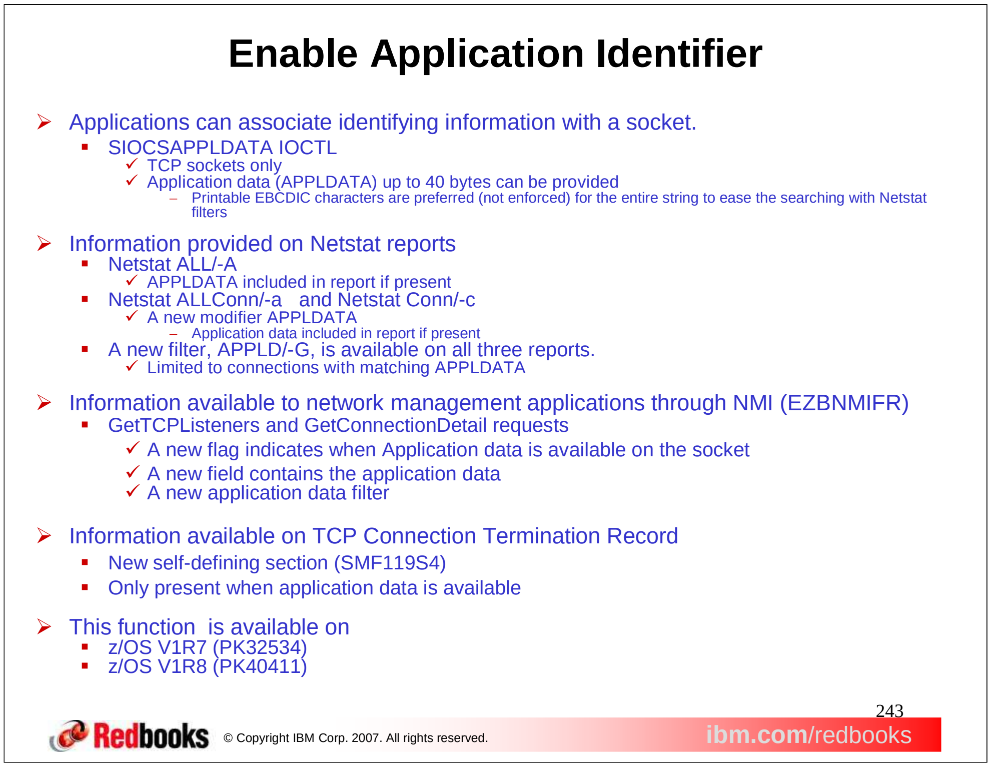
This section discusses the logistics of enabling and exploiting application identifiers or application data against socket connection information returned by NMI, SMF and Netstat.



Need to quickly identify connections used by Applications

- Users have requested the ability to quickly identify TCP connections for key applications.
- Support should be similar to the Netstat support that displays connections with Telnet-specific information.
- The new identifying data should be provided with existing connection information records by the provided management interfaces:
  - Netstat
  - SMF
  - NMI

Often an end user on the telephone with a problem does not know TCP/IP resource information like IP addresses and port numbers. Instead they know high level information about the application they are using.



Enable Application Identifier

- ➢ Applications can associate identifying information with a socket.
  - SIOCSAPPLDATA IOCTL
    - ✓ TCP sockets only
    - ✓ Application data (APPLDATA) up to 40 bytes can be provided
      - – Printable EBCDIC characters are preferred (not enforced) for the entire string to ease the searching with Netstat filters
- ➢ Information provided on Netstat reports
  - Netstat ALL/-A
    - ✓ APPLDATA included in report if present
  - Netstat ALLConn/-a and Netstat Conn/-c
    - ✓ A new modifier APPLDATA
      - – Application data included in report if present
  - A new filter, APPLD/-G, is available on all three reports.
    - ✓ Limited to connections with matching APPLDATA
- ➢ Information available to network management applications through NMI (EZBNMIFR)
  - GetTCPListeners and GetConnectionDetail requests
    - ✓ A new flag indicates when Application data is available on the socket
    - ✓ A new field contains the application data
    - ✓ A new application data filter
- ➢ Information available on TCP Connection Termination Record
  - New self-defining section (SMF119S4)
  - Only present when application data is available
- ➢ This function is available on
  - z/OS V1R7 (PK32534)
  - z/OS V1R8 (PK40411)

243

Redbooks ibm.com/redbooks

To help real time problem determination, as well as capacity planning and accounting applications, TCP/IP applications can now associate identifying information with the socket resources they use.

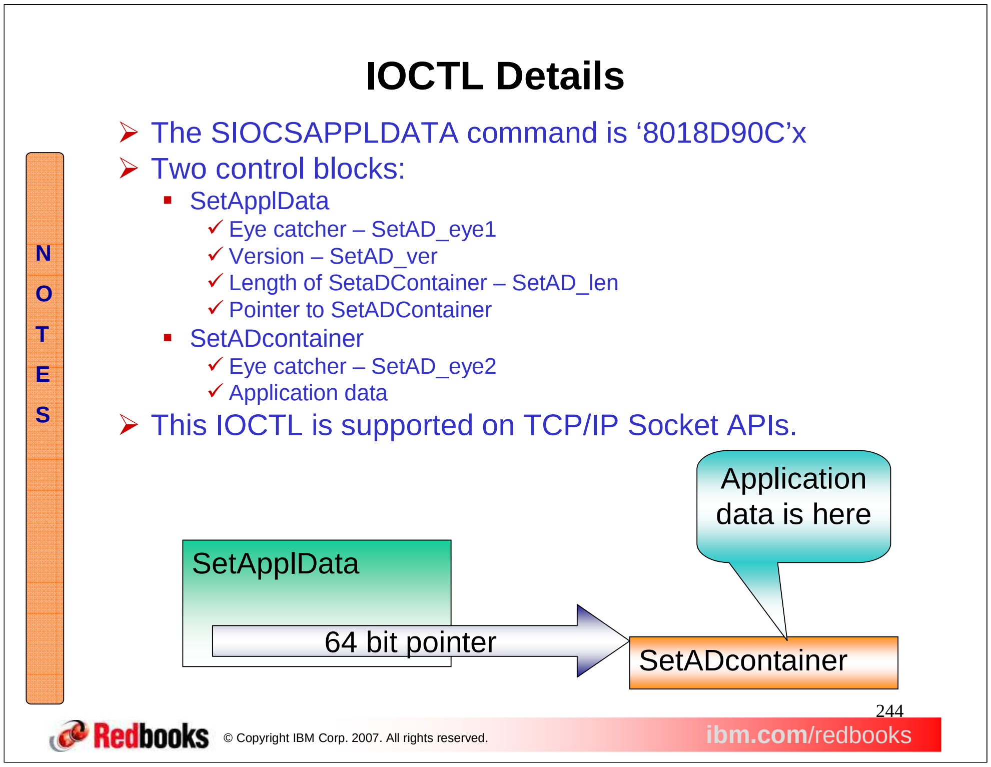
A new IOCTL command is provided to allow applications to associate up to 40 bytes of identifying information with a TCP socket. Application supplied data exists for the life of the socket. It can be replaced by the application at any time. It can be removed by setting it to all nulls or all blanks. Information is inherited from listener to new connections when they are placed on the backlog. It is up to each exploiting application to document the content, format and meaning of the information provided. The application should uniquely identify itself at the beginning of string. Strings beginning with three character IBM product identifiers are reserved for IBM use. IBM product identifiers begin with a letter in the range A to I. Printable EBCDIC characters are preferred, but are not enforced, for the entire string to ease the searching with Netstat filters. If an application chooses to use non-printable characters, they will be displayed by Netstat as '.' (dot). Users entering APPLD filters will need to enter a wildcard character (? or *) to match the non-printable characters stored in the string.

Customers can easily locate and display connections used by the applications since the unique application data is provided on Netstat reports. Application data is always displayed on the TSO Netstat ALL and z/OS Unix netstat –A report if it is present. Application data is only displayed on the MVS and TSO Netstat ALLConn and COnn reports and the z/OS Unix netstat -a and netstat –c reports when explicitly requested with either the APPLDATA modifier or the APPLD/-G filter. The filter is also supported on the Netstat ALL/-A commands. The filter supports wildcard searches. Wildcards supported are question mark (?), which matches exactly one arbitrary character, and an asterisk (*), which matches zero or more arbitrary characters. The filter provided results in a case insensitive search.

The application data is also available to network management applications through the Network Management Interface. EZBNMIFR is a callable polling-type interface that returns the status of connections, listeners, and endpoints at a given point in time. The caller can specify filters that limit the returned data to a specific set of information. GetTCPListeners requests information about all listening TCP sockets. GetConnectionDetail requests information about all connected TCP sockets. This support adds a new flag, NWMTCPLApplDataSet, that indicates when ApplData is available on a socket and a new field, NWMConnApplData, containing the ApplData for these two requests. These requests may be filtered in several ways to reduce the number of sockets included in the results. This support adds a new filter that only matches sockets that have ApplData available and that matches the supplied filter. The application data filter can have wildcard characters. Use a question mark (?) as a wildcard for a single character and an asterisk (*) as a wildcard for zero or more characters. The filter is case insensitive. The macro, EZBNMRHA, and header, EZBNMRHC, are supplied to assist in writing applications that use the NMI.

This solution also allows customers or vendor tools to identify these connections in TCP/IP SMF records written at connection termination. Application data is added to the SMF Type 119 TCP Connection Termination Record as a new self-defining section. This record is produced when a TCP connection is terminated. The macro, EZASMF77, and header, EZASMF, are supplied to assist in writing programs that process SMF records. Application data is available when the SIOCSAPPLDATA IOCTL is used to supply ApplData on the parent listening socket or on the connected socket. Application data can be deleted from a socket by using the SIOCSAPPLDATA IOCTL to supply ApplData that is all blanks or all nulls (x'00').

This support was rolled back to z/OS V1R7 and V1R8 at the request of other IBM applications that are interested in exploiting it.



N O T E S

IOCTL Details

- The SIOCSAPPLDATA command is ‘8018D90C’x
- Two control blocks:
  - SetApplData
    - ✓ Eye catcher – SetAD_eye1
    - ✓ Version – SetAD_ver
    - ✓ Length of SetaDContainer – SetAD_len
    - ✓ Pointer to SetADContainer
  - SetADcontainer
    - ✓ Eye catcher – SetAD_eye2
    - ✓ Application data
- This IOCTL is supported on TCP/IP Socket APIs.

Application data is here

SetApplData

64 bit pointer

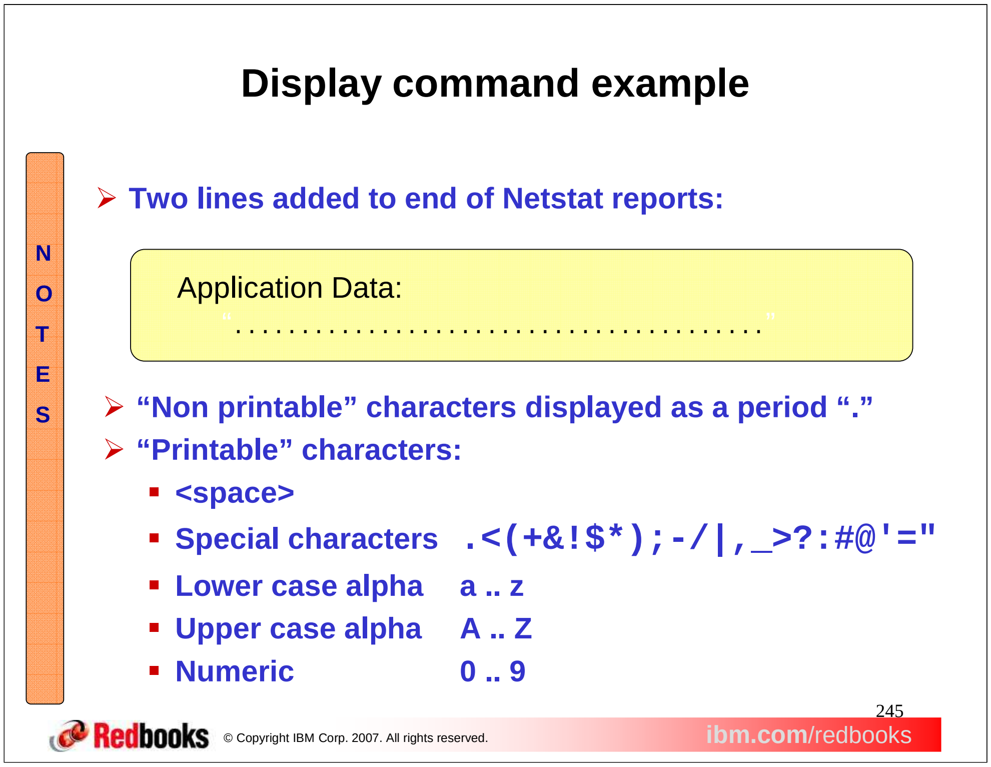
SetADcontainer

244

Redbooks ibm.com/redbooks

As with any IOCTL socket command, a unique socket command variable following by a structure to map the IOCTL request data. This slide illustrates the structure elements and their relationship. Note that the structure containing the application data can optionally reside above the 2G bar.

The SIOCSAPPLDATA IOCTL data structures are in assembler and C/C++. The assembler macro, EZBYAPPL, is located in the SEZANMAC dataset. The C/C++ header, EZBYAPLC, is located in SEZANMAC and in /usr/lpp/tcpip/include/ezbyaplc.h



Display command example

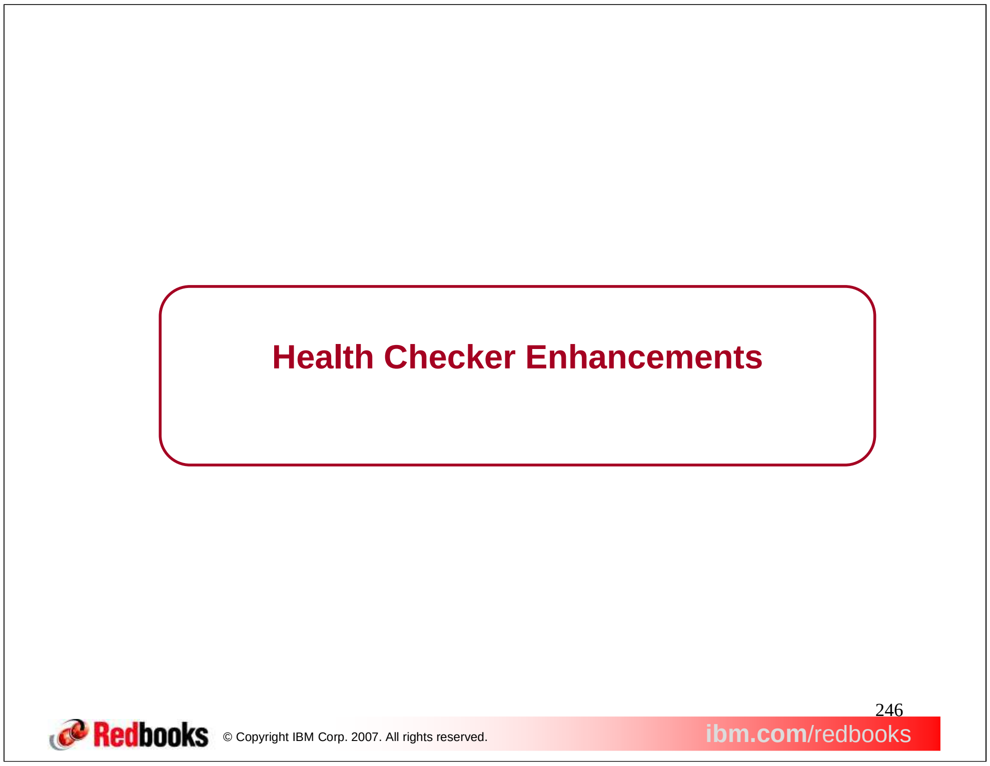
N O T E S

- ➢ Two lines added to end of Netstat reports:

```
Application Data:
    “........................................”
```

- ➢ “Non printable” characters displayed as a period “.”
- ➢ “Printable” characters:
  - <space>
  - Special characters `.<(+&!$*);-/|,_>?:#@'="`
  - Lower case alpha a .. z
  - Upper case alpha A .. Z
  - Numeric 0 .. 9

Applications may place non-printable characters in the string. Netstat will display them as ‘.’. Only printable characters may be entered in Netstat filters. Non-printable characters must be skipped over with wild card characters in the filter.

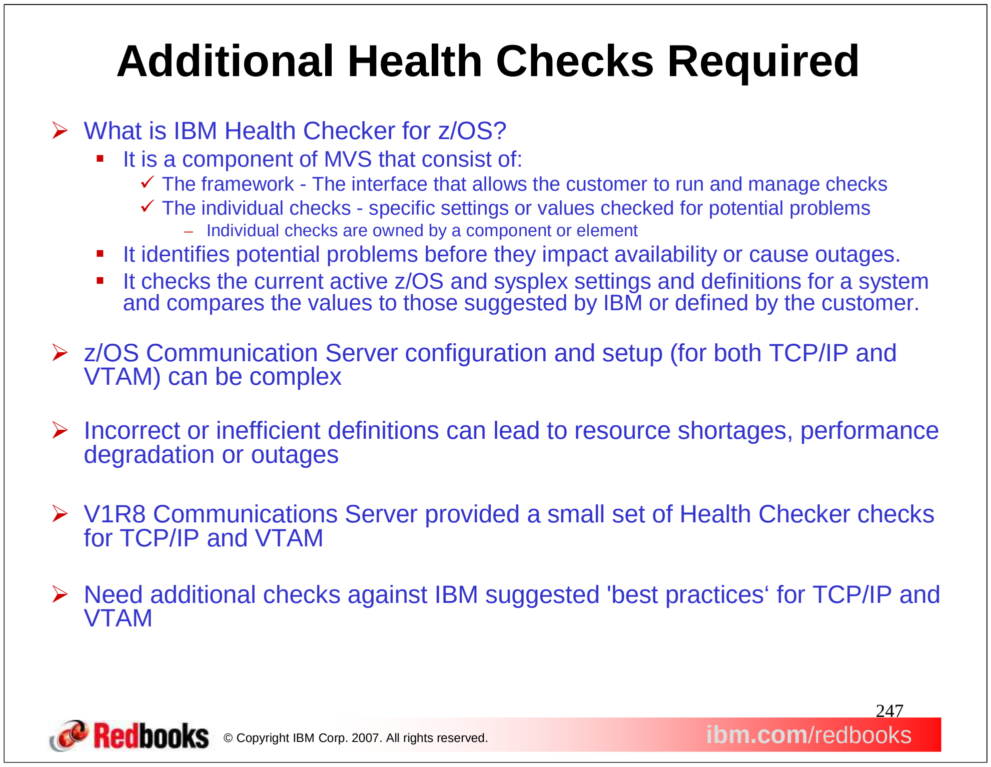


Health Checker Enhancements

246

ibm.com/redbooks

This section describes the z/OS Communications Server Health Checker Enhancements for V1R9.



Additional Health Checks Required

- What is IBM Health Checker for z/OS?
  - It is a component of MVS that consist of:
    - ✓ The framework - The interface that allows the customer to run and manage checks
    - ✓ The individual checks - specific settings or values checked for potential problems
      - Individual checks are owned by a component or element
  - It identifies potential problems before they impact availability or cause outages.
  - It checks the current active z/OS and sysplex settings and definitions for a system and compares the values to those suggested by IBM or defined by the customer.
- z/OS Communication Server configuration and setup (for both TCP/IP and VTAM) can be complex
- Incorrect or inefficient definitions can lead to resource shortages, performance degradation or outages
- V1R8 Communications Server provided a small set of Health Checker checks for TCP/IP and VTAM
- Need additional checks against IBM suggested 'best practices' for TCP/IP and VTAM

Health Checker is an MVS component that provides a framework for running and managing checks on the MVS system. There is a set of checks that are provided by the z/OS components and elements.

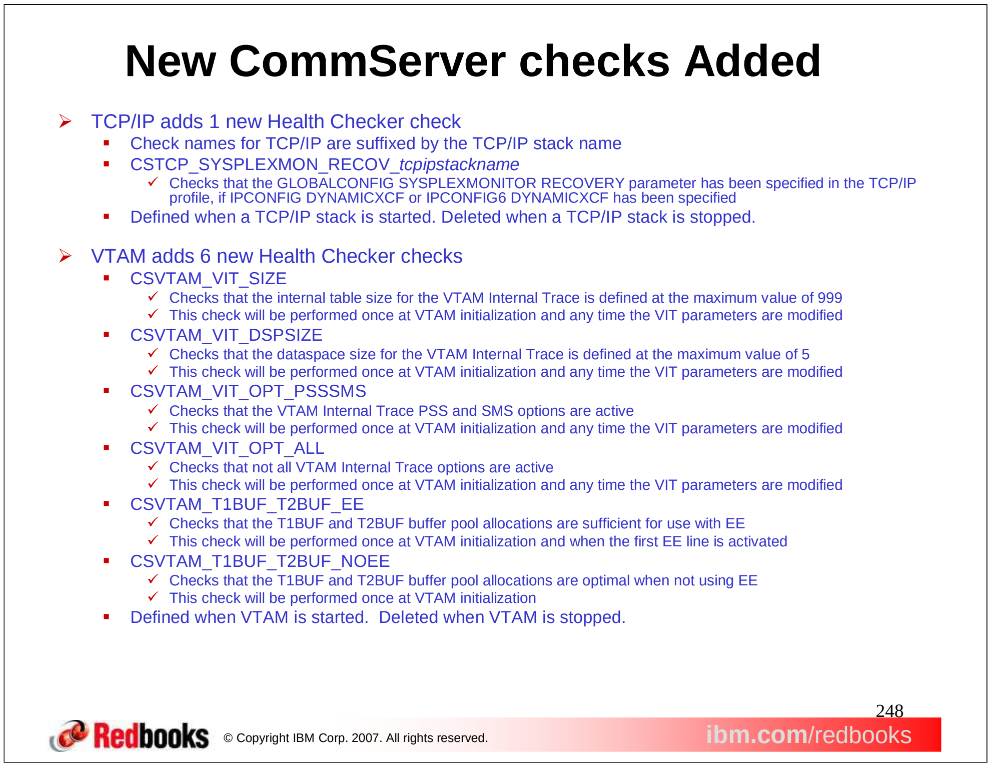
Configuration can be complicated.   Many outages or performance bottlenecks are caused by configuration problems. Sometimes, default values are best guesses. Best practices may not become known until exposure in many environments.

Customers can also create their own checks. The checks provided by IBM will compare current active z/OS and sysplex settings and definitions to those suggested by IBM as 'best practice' values. The intent is to be able to identify potential problems in the z/OS and sysplex configuration before they can impact system performance or availability.

The checks produce output in the form of detailed messages that identify the potential problem and suggest actions to be taken. These messages can be viewed using SDSF, the HZSPRINT utility, or via a log stream. In addition, if a potential problem is found, an operator's console message is issued. Use the information in the check message to resolve possible configuration problems.

There are several steps required to set up Health Check to run checks. You must allocate the HZSPDATA data set to save the check data between restarts. You must set up the HZSPRINT utility. You can define log streams for the check output if you want to maintain an historical record of your check output. You will need to create security definitions, including Multilevel security definitions, if necessary. You can create an HZSPRMxx member from the HZSPRM00 parmlib member if you want to make changes to check values and parameters, or if you want to deactivate a check. You can then start the Health Checker proc. Step-by-step instructions for setting up Health Checker can be found in the IBM Health Checker for z/OS User's Guide.

z/OS Communications Server added some Health Checker checks in release V1R8. Some additional checks will be added for V1R9.



New CommServer checks Added

- TCP/IP adds 1 new Health Checker check
  - Check names for TCP/IP are suffixed by the TCP/IP stack name
  - CSTCP_SYSPLEXMON_RECOV_*tcpipstackname*
    - ✓ Checks that the GLOBALCONFIG SYSPLEXMONITOR RECOVERY parameter has been specified in the TCP/IP profile, if IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF has been specified
  - Defined when a TCP/IP stack is started. Deleted when a TCP/IP stack is stopped.
- VTAM adds 6 new Health Checker checks
  - CSVTAM_VIT_SIZE
    - ✓ Checks that the internal table size for the VTAM Internal Trace is defined at the maximum value of 999
    - ✓ This check will be performed once at VTAM initialization and any time the VIT parameters are modified
  - CSVTAM_VIT_DSPSIZE
    - ✓ Checks that the dataspace size for the VTAM Internal Trace is defined at the maximum value of 5
    - ✓ This check will be performed once at VTAM initialization and any time the VIT parameters are modified
  - CSVTAM_VIT_OPT_PSSSMS
    - ✓ Checks that the VTAM Internal Trace PSS and SMS options are active
    - ✓ This check will be performed once at VTAM initialization and any time the VIT parameters are modified
  - CSVTAM_VIT_OPT_ALL
    - ✓ Checks that not all VTAM Internal Trace options are active
    - ✓ This check will be performed once at VTAM initialization and any time the VIT parameters are modified
  - CSVTAM_T1BUF_T2BUF_EE
    - ✓ Checks that the T1BUF and T2BUF buffer pool allocations are sufficient for use with EE
    - ✓ This check will be performed once at VTAM initialization and when the first EE line is activated
  - CSVTAM_T1BUF_T2BUF_NOEE
    - ✓ Checks that the T1BUF and T2BUF buffer pool allocations are optimal when not using EE
    - ✓ This check will be performed once at VTAM initialization
  - Defined when VTAM is started. Deleted when VTAM is stopped.

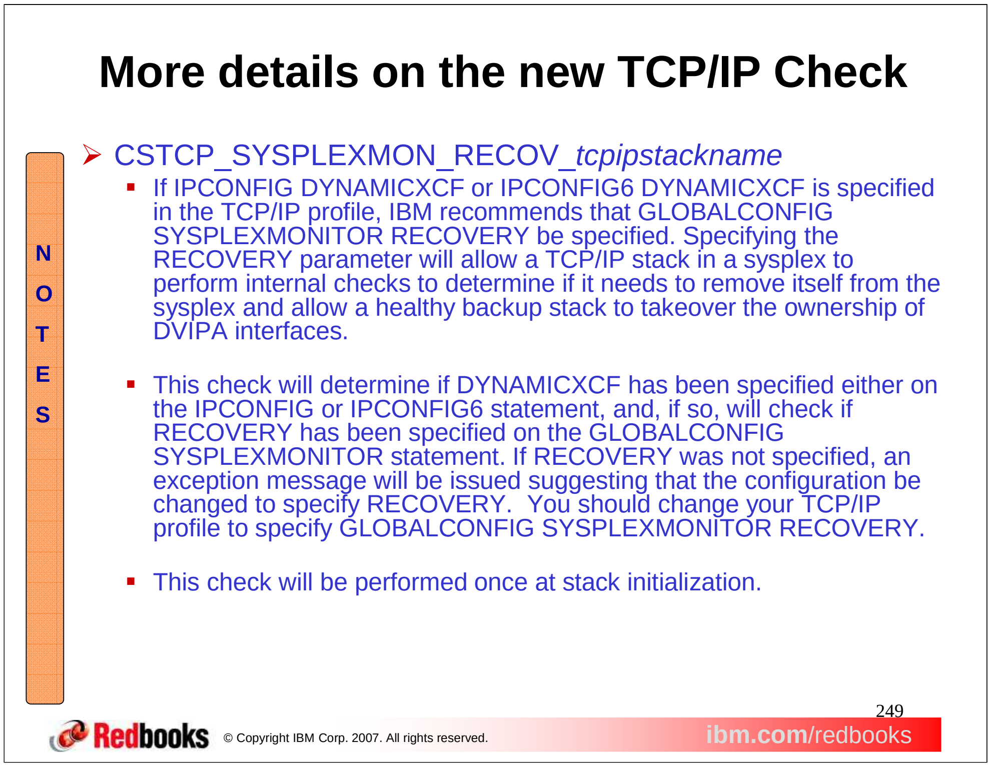
248

Redbooks ibm.com/redbooks

New Communications Server Health Checker checks have been added in z/OS V1R9.

TCP/IP adds one new Health Checker check. As with the V1R8 TCP/IP checks, the check name will be suffixed by the TCP/IP stack name, so as to provide a unique check name for each started stack on a system. All TCP/IP checks are defined when the TCP/IP stack is started and deleted when the TCP/IP stack is stopped. The new V1R9 TCP/IP check is CSTCP_SYSPLEXMON_RECOV_tcpipstackname. It will check if the SYSPLEXMONITOR RECOVERY parameter has been specified on the GLOBALCONFIG statement, if the DYNAMICXCF parameter has been specified on an IPCONFIG or IPCONFIG6 statement.

VTAM adds six new Health Checker checks. As with the V1R8 VTAM check, these checks are defined when VTAM is started and deleted when VTAM is stopped. This slide lists the name of each new check and a brief description of what check is performed.



More details on the new TCP/IP Check

NOTES

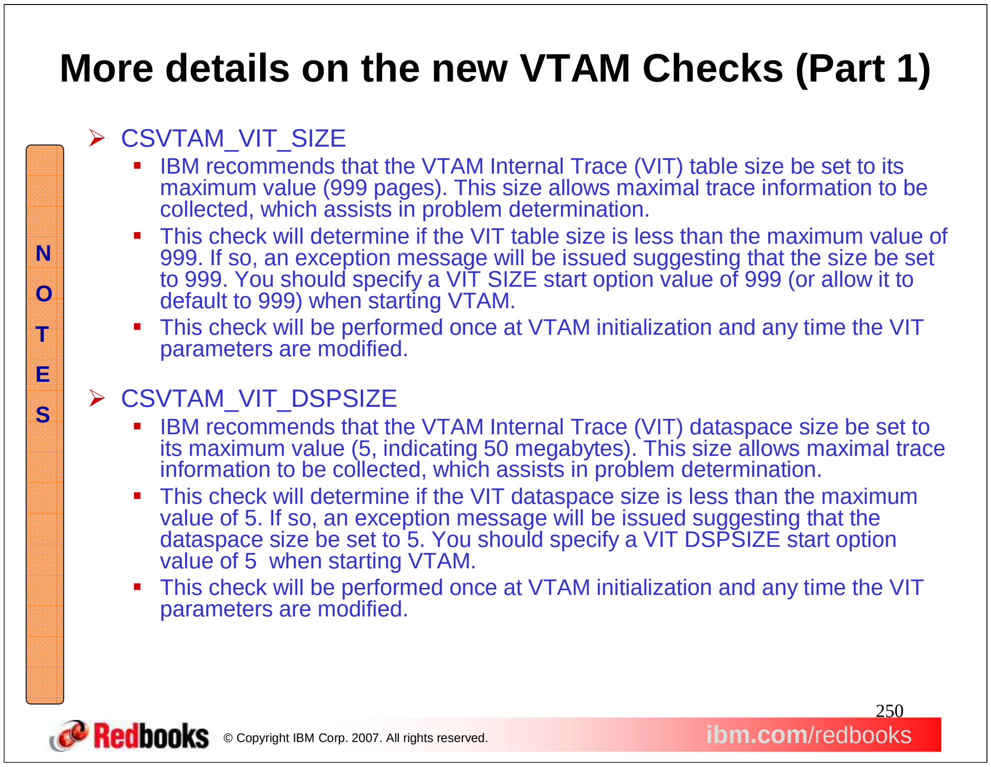
- CSTCP_SYSPLEXMON_RECOV_*tcpipstackname*
  - If IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF is specified in the TCP/IP profile, IBM recommends that GLOBALCONFIG SYSPLEXMONITOR RECOVERY be specified. Specifying the RECOVERY parameter will allow a TCP/IP stack in a sysplex to perform internal checks to determine if it needs to remove itself from the sysplex and allow a healthy backup stack to takeover the ownership of DVIPA interfaces.
  - This check will determine if DYNAMICXCF has been specified either on the IPCONFIG or IPCONFIG6 statement, and, if so, will check if RECOVERY has been specified on the GLOBALCONFIG SYSPLEXMONITOR statement. If RECOVERY was not specified, an exception message will be issued suggesting that the configuration be changed to specify RECOVERY. You should change your TCP/IP profile to specify GLOBALCONFIG SYSPLEXMONITOR RECOVERY.
  - This check will be performed once at stack initialization.

249

Redbooks

ibm.com/redbooks

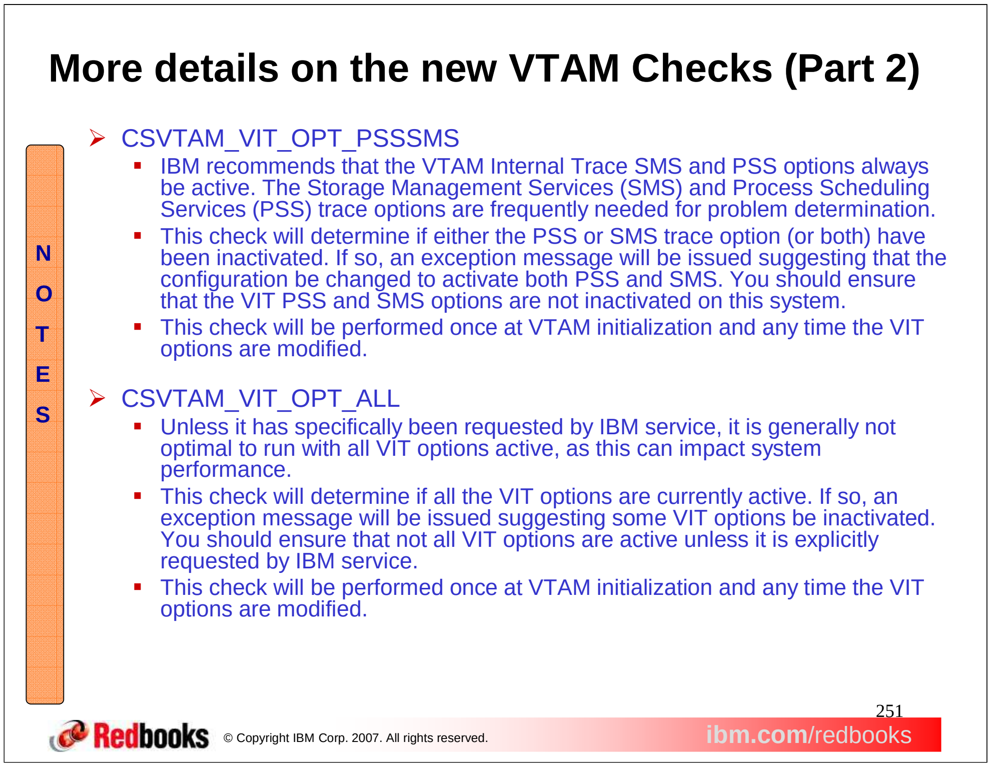
This slide contains additional details about the new TCP/IP check. It describes why IBM suggests that RECOVERY be specified when DYNAMICXCF is specified. It also indicates under what conditions the check will issue an exception message.



More details on the new VTAM Checks (Part 1)

- CSVTAM_VIT_SIZE
  - IBM recommends that the VTAM Internal Trace (VIT) table size be set to its maximum value (999 pages). This size allows maximal trace information to be collected, which assists in problem determination.
  - This check will determine if the VIT table size is less than the maximum value of 999. If so, an exception message will be issued suggesting that the size be set to 999. You should specify a VIT SIZE start option value of 999 (or allow it to default to 999) when starting VTAM.
  - This check will be performed once at VTAM initialization and any time the VIT parameters are modified.
- CSVTAM_VIT_DSPSIZE
  - IBM recommends that the VTAM Internal Trace (VIT) dataspace size be set to its maximum value (5, indicating 50 megabytes). This size allows maximal trace information to be collected, which assists in problem determination.
  - This check will determine if the VIT dataspace size is less than the maximum value of 5. If so, an exception message will be issued suggesting that the dataspace size be set to 5. You should specify a VIT DSPSIZE start option value of 5 when starting VTAM.
  - This check will be performed once at VTAM initialization and any time the VIT parameters are modified.

This panel provides more details about two of the new V1R9 VTAM checks, describing why each check is needed and under what conditions an exception message is generated.



More details on the new VTAM Checks (Part 2)

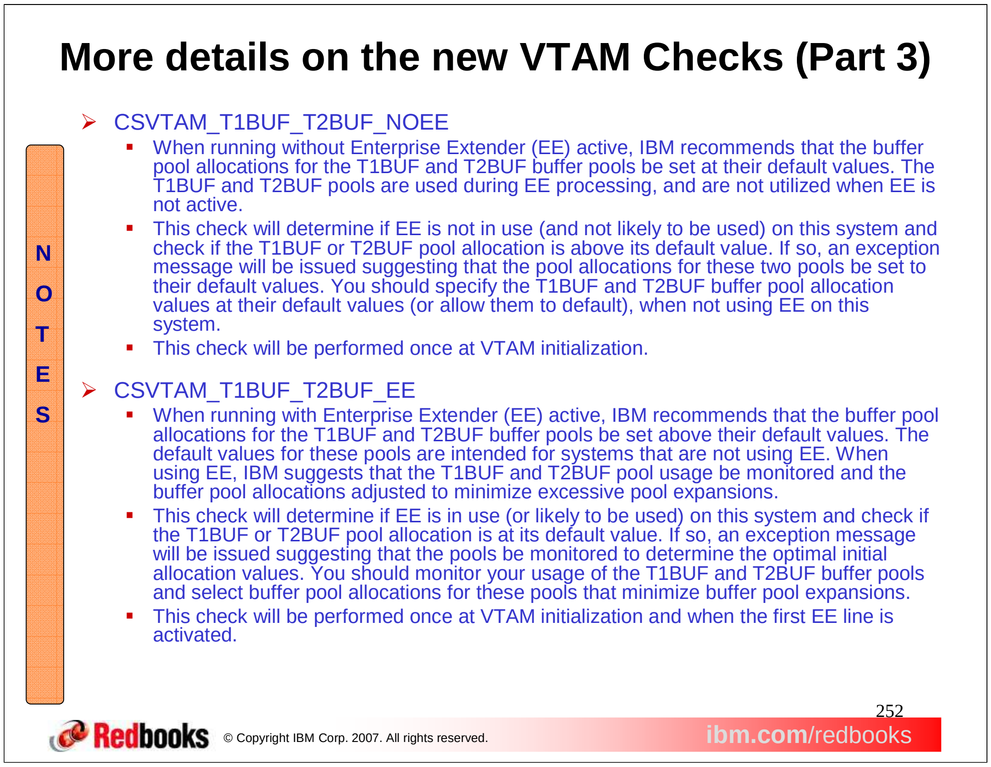
NOTES

- CSVTAM_VIT_OPT_PSSSMS
  - IBM recommends that the VTAM Internal Trace SMS and PSS options always be active. The Storage Management Services (SMS) and Process Scheduling Services (PSS) trace options are frequently needed for problem determination.
  - This check will determine if either the PSS or SMS trace option (or both) have been inactivated. If so, an exception message will be issued suggesting that the configuration be changed to activate both PSS and SMS. You should ensure that the VIT PSS and SMS options are not inactivated on this system.
  - This check will be performed once at VTAM initialization and any time the VIT options are modified.
- CSVTAM_VIT_OPT_ALL
  - Unless it has specifically been requested by IBM service, it is generally not optimal to run with all VIT options active, as this can impact system performance.
  - This check will determine if all the VIT options are currently active. If so, an exception message will be issued suggesting some VIT options be inactivated. You should ensure that not all VIT options are active unless it is explicitly requested by IBM service.
  - This check will be performed once at VTAM initialization and any time the VIT options are modified.

251

Redbooks ibm.com/redbooks

This slide provides more details about the next 2 of the new V1R9 VTAM checks, describing why each check is needed and under what conditions an exception message is generated.



More details on the new VTAM Checks (Part 3)

N O T E S

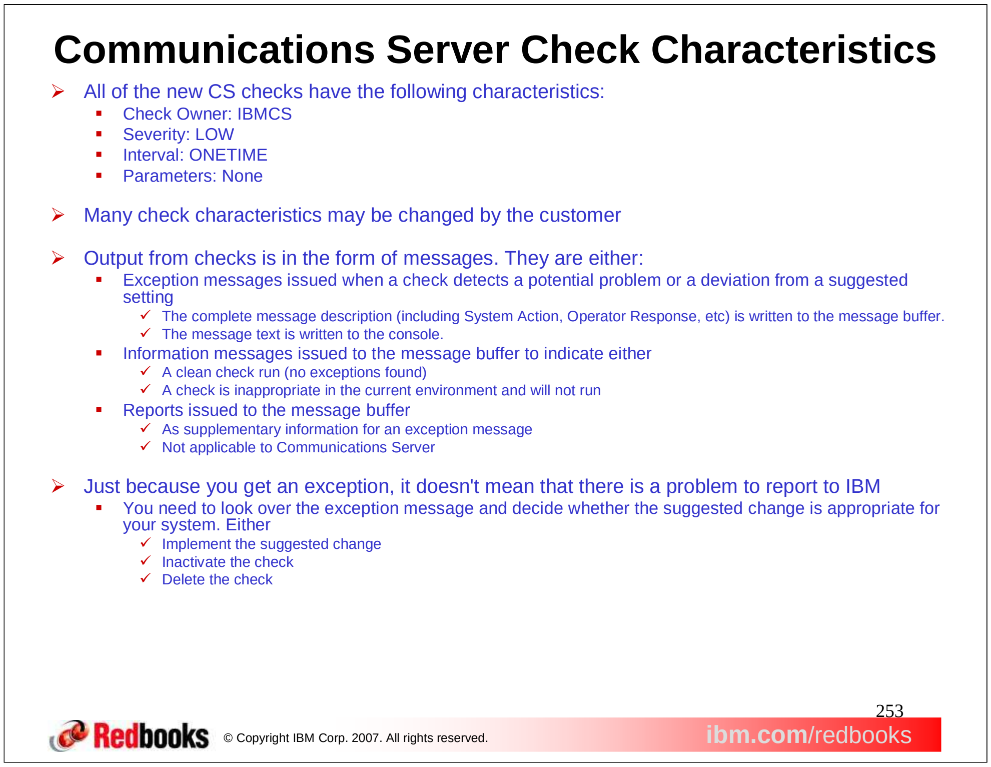
- CSVTAM_T1BUF_T2BUF_NOEE
  - When running without Enterprise Extender (EE) active, IBM recommends that the buffer pool allocations for the T1BUF and T2BUF buffer pools be set at their default values. The T1BUF and T2BUF pools are used during EE processing, and are not utilized when EE is not active.
  - This check will determine if EE is not in use (and not likely to be used) on this system and check if the T1BUF or T2BUF pool allocation is above its default value. If so, an exception message will be issued suggesting that the pool allocations for these two pools be set to their default values. You should specify the T1BUF and T2BUF buffer pool allocation values at their default values (or allow them to default), when not using EE on this system.
  - This check will be performed once at VTAM initialization.
- CSVTAM_T1BUF_T2BUF_EE
  - When running with Enterprise Extender (EE) active, IBM recommends that the buffer pool allocations for the T1BUF and T2BUF buffer pools be set above their default values. The default values for these pools are intended for systems that are not using EE. When using EE, IBM suggests that the T1BUF and T2BUF pool usage be monitored and the buffer pool allocations adjusted to minimize excessive pool expansions.
  - This check will determine if EE is in use (or likely to be used) on this system and check if the T1BUF or T2BUF pool allocation is at its default value. If so, an exception message will be issued suggesting that the pools be monitored to determine the optimal initial allocation values. You should monitor your usage of the T1BUF and T2BUF buffer pools and select buffer pool allocations for these pools that minimize buffer pool expansions.
  - This check will be performed once at VTAM initialization and when the first EE line is activated.

252

Redbooks

ibm.com/redbooks

This slide provides more details about the final 2 of the new V1R9 VTAM checks, describing why each check is needed and under what conditions an exception message is generated.



Communications Server Check Characteristics

- All of the new CS checks have the following characteristics:
  - Check Owner: IBMCS
  - Severity: LOW
  - Interval: ONETIME
  - Parameters: None
- Many check characteristics may be changed by the customer
- Output from checks is in the form of messages. They are either:
  - Exception messages issued when a check detects a potential problem or a deviation from a suggested setting
    - ✓ The complete message description (including System Action, Operator Response, etc) is written to the message buffer.
    - ✓ The message text is written to the console.
  - Information messages issued to the message buffer to indicate either
    - ✓ A clean check run (no exceptions found)
    - ✓ A check is inappropriate in the current environment and will not run
  - Reports issued to the message buffer
    - ✓ As supplementary information for an exception message
    - ✓ Not applicable to Communications Server
- Just because you get an exception, it doesn't mean that there is a problem to report to IBM
  - You need to look over the exception message and decide whether the suggested change is appropriate for your system. Either
    - ✓ Implement the suggested change
    - ✓ Inactivate the check
    - ✓ Delete the check

Redbooks

ibm.com/redbooks

The check characteristics listed on this slide are common to all the new V1R9 Communications Server checks. Check Owner is the name of the z/OS component that owns the check. Check Owner plus Check Name uniquely identifies a check. For z/OS Communication Server checks, the Check Owner is IBMCS. Severity indicates the severity level of the check. Health Checker allows 3 levels of severity:

•LOW - When a low-severity check detects an exception, an informational WTO is issued.

•MED - When a medium-severity check detects an exception, an eventual action WTO is issued.

•HI - When a high-severity check detects an exception, a critical eventual action WTO is issued.

Interval indicates the frequency of the check. ONETIME indicates the check will run once and will not be rescheduled. Otherwise, a time interval in hours and minutes may be specified. A check may have one or more parameters specifying values that are used in the check. Note that there are no parameters associated with any of the new checks.

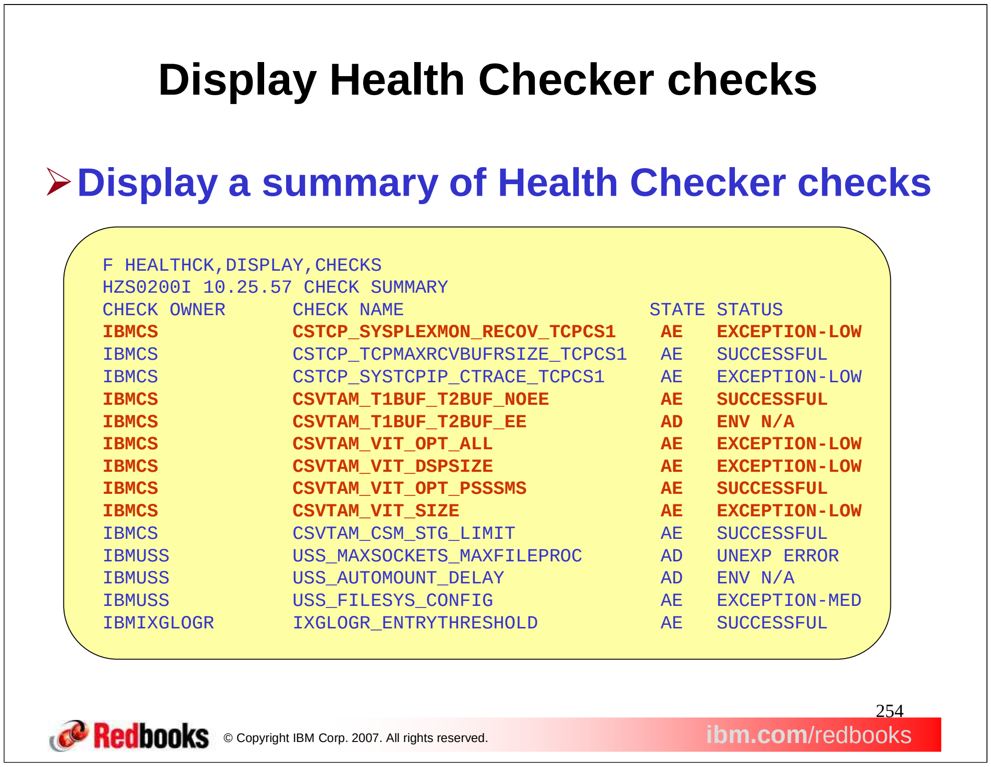
Many of the check characteristics (Severity, Interval, Parameter values) may be changed by the customer. Dynamic, temporary changes may be made either using the SDSF CK command or through the MODIFY *hzsproc* command. Persistent changes may be made through entries in the HZSPRMxx parmlib member. See *IBM Health Checker for z/OS User's Guide* for details on modifying check characteristics.

Output from checks is in the form of messages. They can be either exception messages, information messages are reports. Note that no current Communications Server checks issue reports.

Complete output messages in the message buffer can be viewed using the HZSPRINT utility, the SDSF CK command or a log stream. You may need to set up authorization through your Security Access Facility (e.g. RACF) to view the Health Checker message output. See *IBM Health Checker for z/OS User's Guide* for a complete description of how to display check output messages

An exception message merely indicates the potential for an availability or performance problem resulting from the configuration parameters being checked. You shouldn't call IBM service when you receive an exception message. Instead, you should investigate the configuration problem reported in the message and determine whether the problem is applicable to this system. If so, you should implement the change suggested in the check exception message. If you implement the suggestion, an exception message should not be issued when this check is run again. Issue F *hzsproc*,RUN,CHECK=(*checkowner*,*checkname*) to verify that no exception message is issued.

If you believe the check is not applicable, you can inactivate or delete the check. To inactivate the check issue F *hzsproc*,DEACTIVATE,CHECK(*checkowner*,*checkname*) or specify UPDATE INACTIVATE CHECK(*checkowner*,*checkname*) in the HZSPRMxx parmlib member. You can re-activate the check by issuing F *hzsproc*,ACTIVATE,CHECK(*checkowner*,*checkname*) or removing the UPDATE INACTIVATE from the HZSPRMxx parmlib member. To delete the check specify DELETE CHECK(*checkowner*,*checkname*) in the HZSPRMxx parmlib member. You can un-delete the check by removing the DELETE from the HZSPRMxx parmlib member.



Display Health Checker checks

➢ Display a summary of Health Checker checks

```
F HEALTHCK,DISPLAY,CHECKS
HZS0200I 10.25.57 CHECK SUMMARY
CHECK OWNER      CHECK NAME                      STATE STATUS
IBMCS            CSTCP_SYSPLEXMON_RECOV_TCPCS1    AE    EXCEPTION-LOW
IBMCS            CSTCP_TCPMAXRCVBUFRSIZE_TCPCS1   AE    SUCCESSFUL
IBMCS            CSTCP_SYSTCPIP_CTRACE_TCPCS1     AE    EXCEPTION-LOW
IBMCS            CSVTAM_T1BUF_T2BUF_NOEE          AE    SUCCESSFUL
IBMCS            CSVTAM_T1BUF_T2BUF_EE            AD    ENV N/A
IBMCS            CSVTAM_VIT_OPT_ALL               AE    EXCEPTION-LOW
IBMCS            CSVTAM_VIT_DSPSIZE               AE    EXCEPTION-LOW
IBMCS            CSVTAM_VIT_OPT_PSSSMS            AE    SUCCESSFUL
IBMCS            CSVTAM_VIT_SIZE                  AE    EXCEPTION-LOW
IBMCS            CSVTAM_CSM_STG_LIMIT             AE    SUCCESSFUL
IBMUSS           USS_MAXSOCKETS_MAXFILEPROC       AD    UNEXP ERROR
IBMUSS           USS_AUTOMOUNT_DELAY              AD    ENV N/A
IBMUSS           USS_FILESYS_CONFIG               AE    EXCEPTION-MED
IBMIXGLOGR       IXGLOGR_ENTRYTHRESHOLD           AE    SUCCESSFUL
```

This panel shows how to get a summary display of Health Checker checks and their status. This chart displays only a partial list of checks. Highlighted in red are the new V1R9 Communications Server checks. The letters in the state column can be

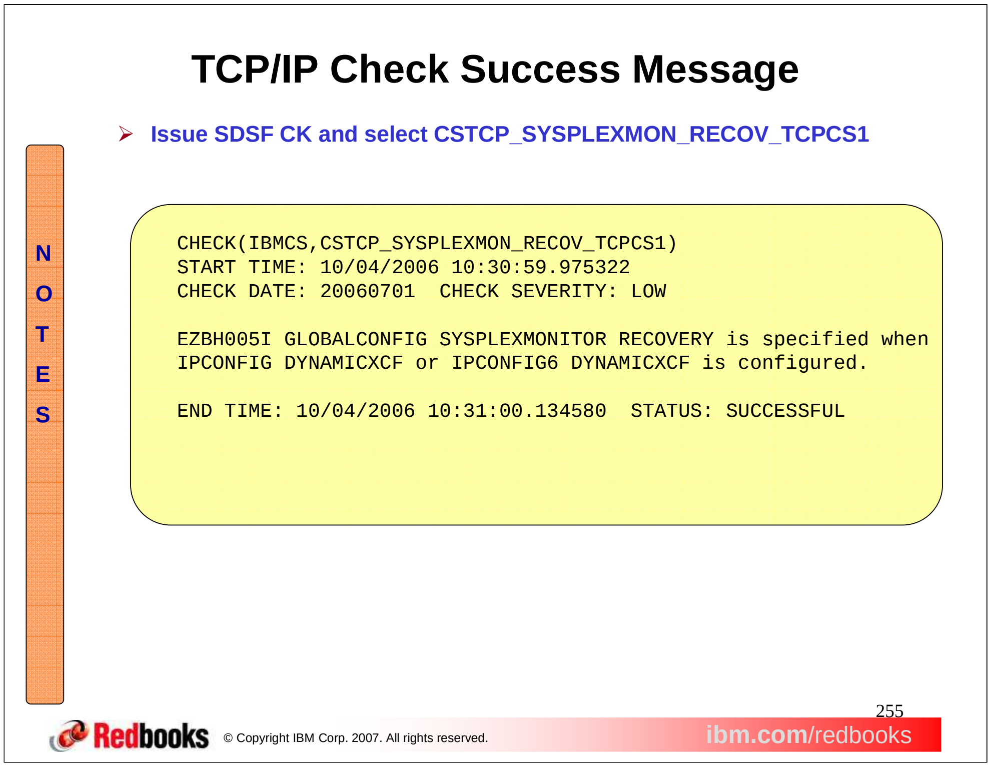
A – Active

I – Inactive

E – Enabled

D - Disabled

The status field of the display shows the status of the check. That is, whether the check was successful or generated an exception message. If an exception message was generated, it indicates if the exception severity level was low, medium, or high. The status field can also indicate if a check was not run because it was not applicable in the current environment or due to an unexpected error during check processing.



TCP/IP Check Success Message

- Issue SDSF CK and select CSTCP_SYSPLEXMON_RECOV_TCPCS1

```
CHECK(IBMCS,CSTCP_SYSPLEXMON_RECOV_TCPCS1)
START TIME: 10/04/2006 10:30:59.975322
CHECK DATE: 20060701  CHECK SEVERITY: LOW

EZBH005I GLOBALCONFIG SYSPLEXMONITOR RECOVERY is specified when
IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF is configured.

END TIME: 10/04/2006 10:31:00.134580  STATUS: SUCCESSFUL
```

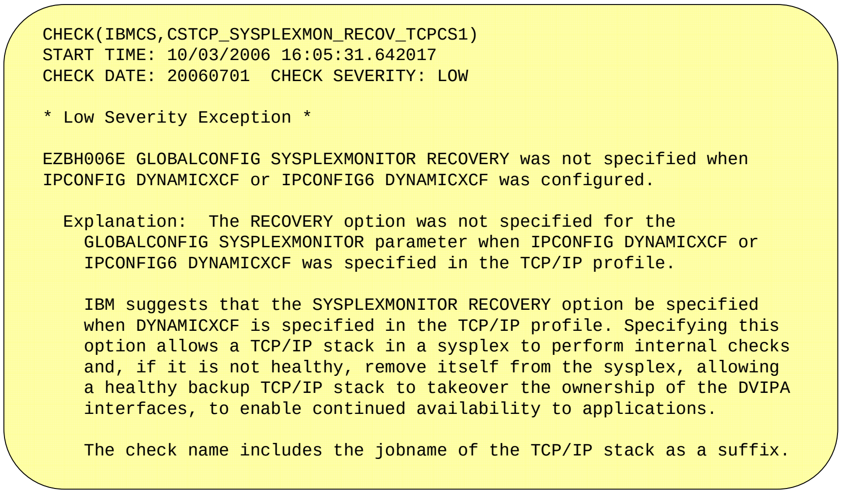
NOTES

This display shows the success message that is issued when the CSTCP_SYSPLEXMON_RECOV_*tcpipstackname* check is run and finds that the configuration agrees with IBM’s suggested ‘best practice’.

TCP/IP Check Exception Message

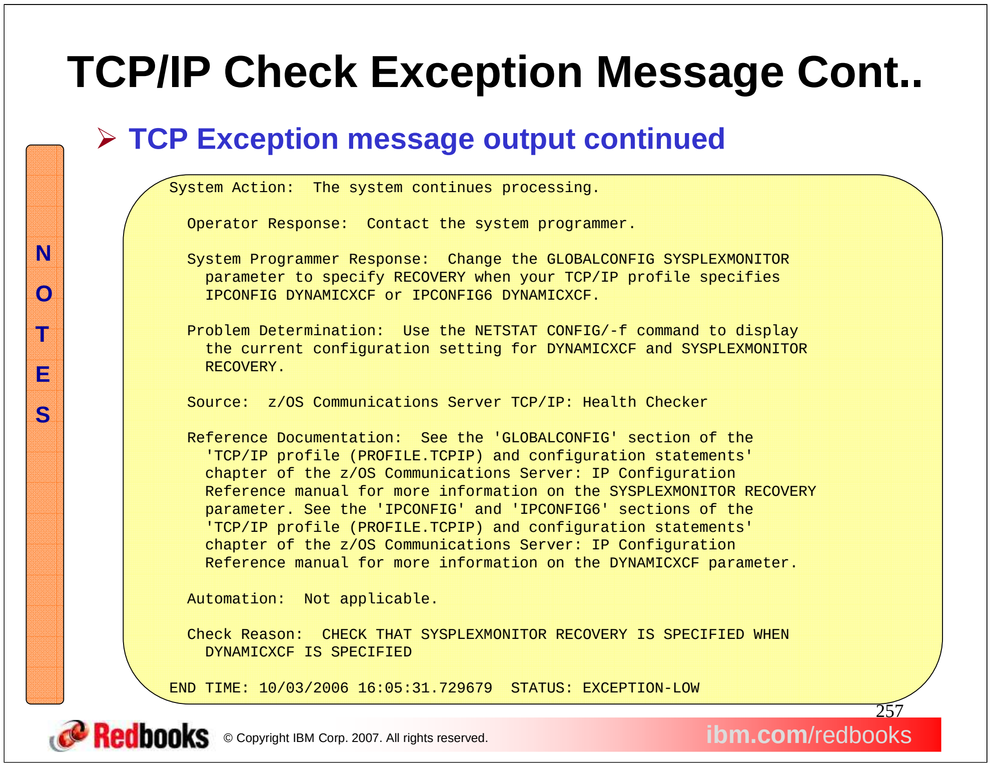
- Issue SDSF CK and select CSTCP_SYSPLEXMON_RECOV_TCPCS1

N O T E S



```
CHECK(IBMCS,CSTCP_SYSPLEXMON_RECOV_TCPCS1)
START TIME: 10/03/2006 16:05:31.642017
CHECK DATE: 20060701  CHECK SEVERITY: LOW

* Low Severity Exception *

EZBH006E GLOBALCONFIG SYSPLEXMONITOR RECOVERY was not specified when
IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF was configured.

  Explanation:  The RECOVERY option was not specified for the
    GLOBALCONFIG SYSPLEXMONITOR parameter when IPCONFIG DYNAMICXCF or
    IPCONFIG6 DYNAMICXCF was specified in the TCP/IP profile.

    IBM suggests that the SYSPLEXMONITOR RECOVERY option be specified
    when DYNAMICXCF is specified in the TCP/IP profile. Specifying this
    option allows a TCP/IP stack in a sysplex to perform internal checks
    and, if it is not healthy, remove itself from the sysplex, allowing
    a healthy backup TCP/IP stack to takeover the ownership of the DVIPA
    interfaces, to enable continued availability to applications.

    The check name includes the jobname of the TCP/IP stack as a suffix.
```

This display shows the first part of the full exception message that is written to the message buffer when the CSTCP_SYSPLEXMON_RECOV_tcpipstackname check detects a problem. This slide and the next slide show that the exception message contains the full message text just as would be found in the z/OS Communications Server: IP Messages manual.

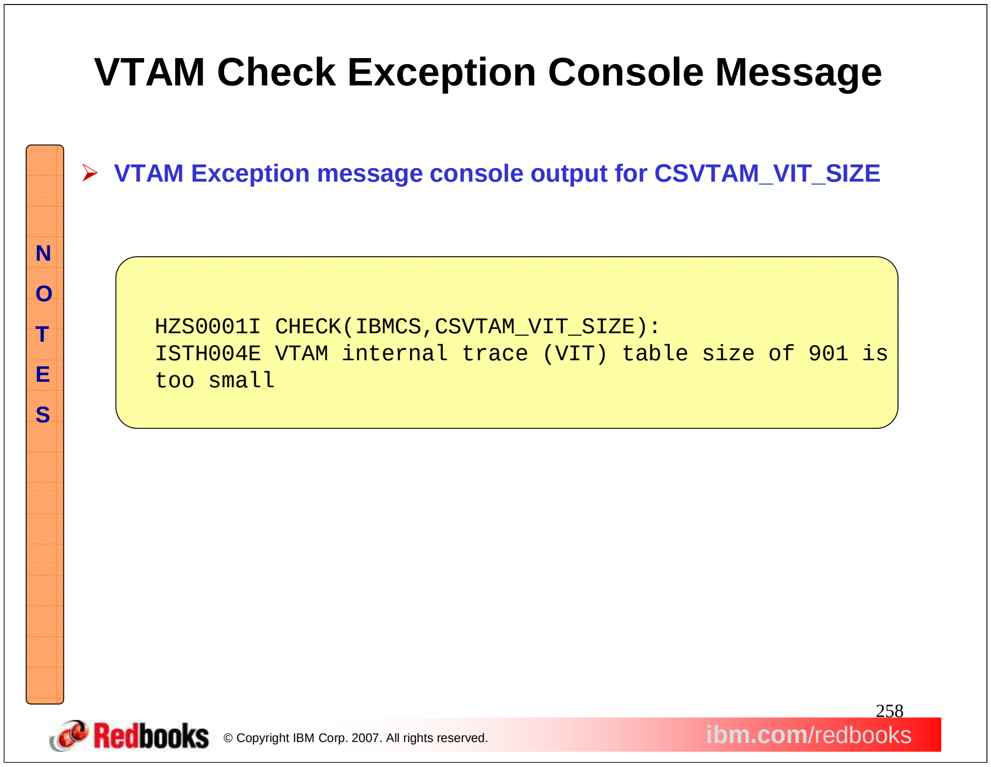


TCP/IP Check Exception Message Cont..

- TCP Exception message output continued

```
System Action:  The system continues processing.

  Operator Response:  Contact the system programmer.

  System Programmer Response:  Change the GLOBALCONFIG SYSPLEXMONITOR
    parameter to specify RECOVERY when your TCP/IP profile specifies
    IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF.

  Problem Determination:  Use the NETSTAT CONFIG/-f command to display
    the current configuration setting for DYNAMICXCF and SYSPLEXMONITOR
    RECOVERY.

  Source:  z/OS Communications Server TCP/IP: Health Checker

  Reference Documentation:  See the 'GLOBALCONFIG' section of the
    'TCP/IP profile (PROFILE.TCPIP) and configuration statements'
    chapter of the z/OS Communications Server: IP Configuration
    Reference manual for more information on the SYSPLEXMONITOR RECOVERY
    parameter. See the 'IPCONFIG' and 'IPCONFIG6' sections of the
    'TCP/IP profile (PROFILE.TCPIP) and configuration statements'
    chapter of the z/OS Communications Server: IP Configuration
    Reference manual for more information on the DYNAMICXCF parameter.

  Automation:  Not applicable.

  Check Reason:  CHECK THAT SYSPLEXMONITOR RECOVERY IS SPECIFIED WHEN
    DYNAMICXCF IS SPECIFIED

END TIME: 10/03/2006 16:05:31.729679  STATUS: EXCEPTION-LOW
```

This slide is a continuation of the display of the exception message for CSTCP_SYSPLEXMON_RECOV_tcpipstackname check.



VTAM Check Exception Console Message

N
O
T
E
S

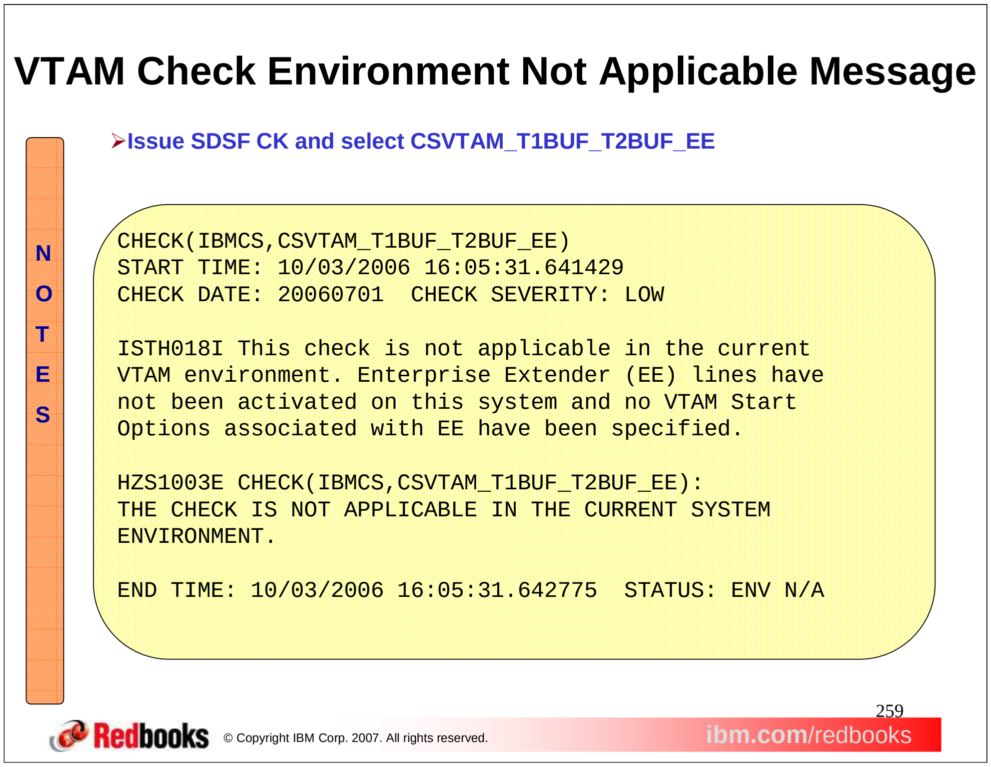
- VTAM Exception message console output for CSVTAM_VIT_SIZE

```
HZS0001I CHECK(IBMCS,CSVTAM_VIT_SIZE):
ISTH004E VTAM internal trace (VIT) table size of 901 is
too small
```

Redbooks

ibm.com/redbooks

This display shows the console message that is issued for the CSVTAM_VIT_SIZE check if a problem is found with the configuration.

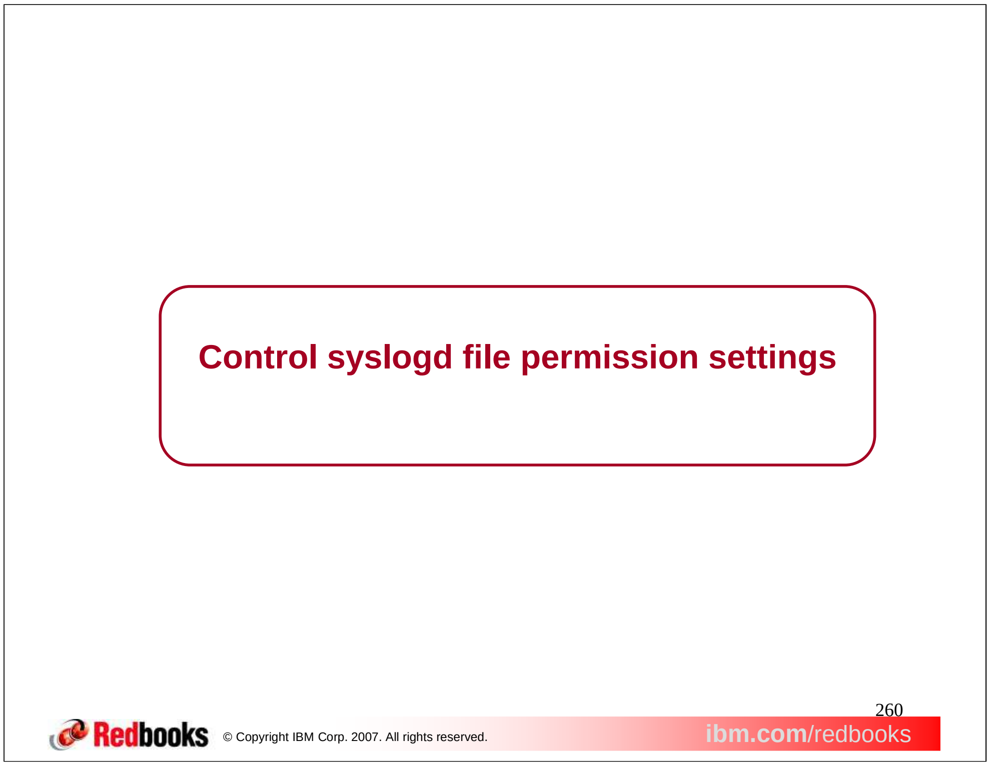


VTAM Check Environment Not Applicable Message

- Issue SDSF CK and select CSVTAM_T1BUF_T2BUF_EE

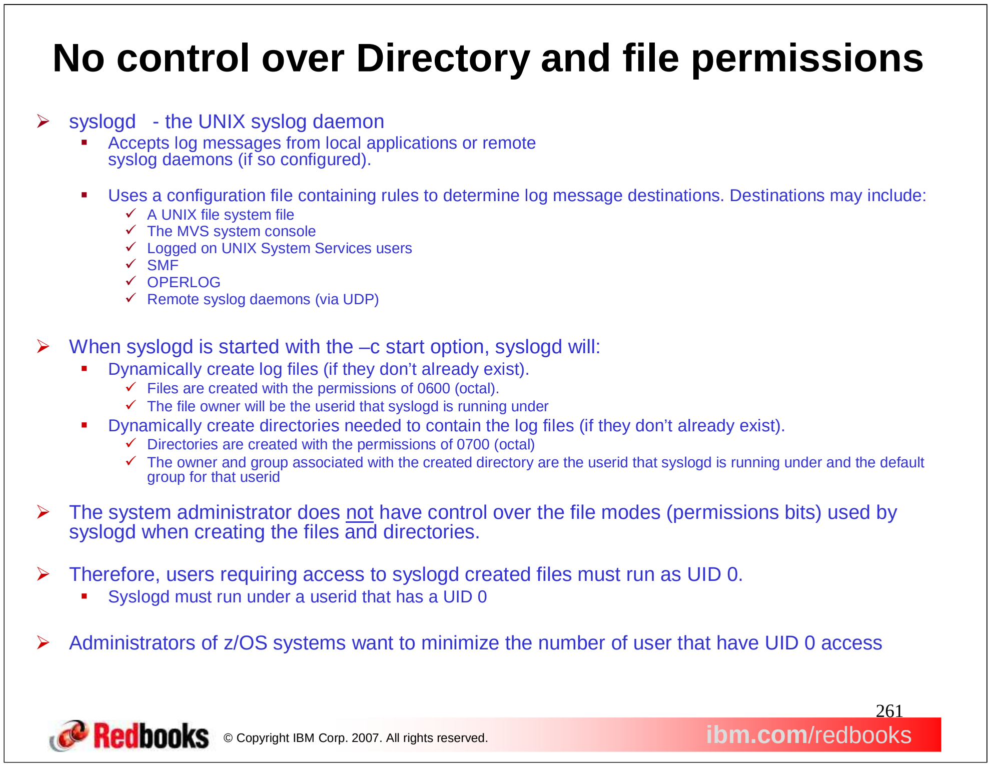
```
CHECK(IBMCS,CSVTAM_T1BUF_T2BUF_EE)
START TIME: 10/03/2006 16:05:31.641429
CHECK DATE: 20060701  CHECK SEVERITY: LOW

ISTH018I This check is not applicable in the current
VTAM environment. Enterprise Extender (EE) lines have
not been activated on this system and no VTAM Start
Options associated with EE have been specified.

HZS1003E CHECK(IBMCS,CSVTAM_T1BUF_T2BUF_EE):
THE CHECK IS NOT APPLICABLE IN THE CURRENT SYSTEM
ENVIRONMENT.

END TIME: 10/03/2006 16:05:31.642775  STATUS: ENV N/A
```

This display shows the message that is written to the message buffer when it is determined that the CSVTAM_T1BUF_T2BUF_EE check is not applicable in the current VTAM environment. If no EE lines are active, and the IPADDR and TCPNAME start options have not been specified when VTAM was started, it is assumed that EE will not be used on this system.



Control syslogd file permission settings

This section describes the changes to control syslogd file permission settings.



No control over Directory and file permissions

- syslogd - the UNIX syslog daemon
  - Accepts log messages from local applications or remote syslog daemons (if so configured).
  - Uses a configuration file containing rules to determine log message destinations. Destinations may include:
    - ✓ A UNIX file system file
    - ✓ The MVS system console
    - ✓ Logged on UNIX System Services users
    - ✓ SMF
    - ✓ OPERLOG
    - ✓ Remote syslog daemons (via UDP)
- When syslogd is started with the –c start option, syslogd will:
  - Dynamically create log files (if they don't already exist).
    - ✓ Files are created with the permissions of 0600 (octal).
    - ✓ The file owner will be the userid that syslogd is running under
  - Dynamically create directories needed to contain the log files (if they don't already exist).
    - ✓ Directories are created with the permissions of 0700 (octal)
    - ✓ The owner and group associated with the created directory are the userid that syslogd is running under and the default group for that userid
- The system administrator does not have control over the file modes (permissions bits) used by syslogd when creating the files and directories.
- Therefore, users requiring access to syslogd created files must run as UID 0.
  - Syslogd must run under a userid that has a UID 0
- Administrators of z/OS systems want to minimize the number of user that have UID 0 access

261

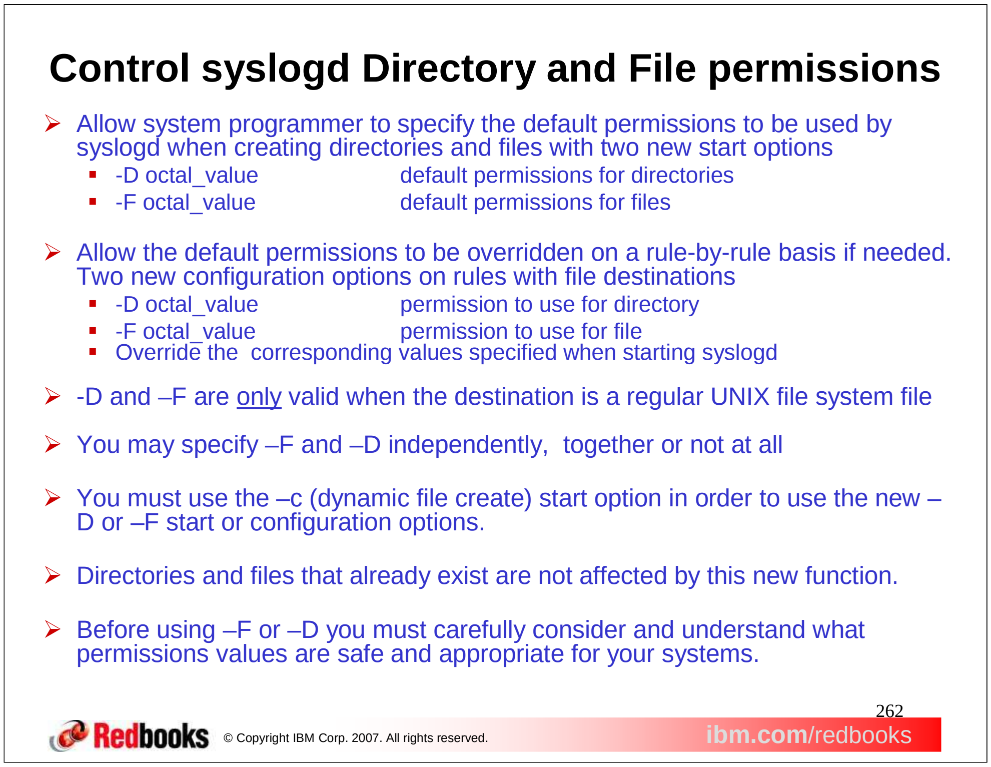
Redbooks

ibm.com/redbooks

syslogd is the UNIX syslog daemon. It is a program that runs on z/OS and accepts log messages generated by local applications or from remote syslog daemons running on other systems in the network. syslogd uses a default configuration file named /etc/syslog.conf or you may start it by specifying a configuration file as a start parameter. The syslogd configuration file contains rules that tell syslogd where to log messages based on the message's facility and priority. The z/OS syslogd supports the following destinations for messages: A UNIX file system file, the MVS system console, logged on users, SMF, OPERLOG and forwarding via UDP to remote syslog daemons elsewhere in the network.

A common destination for syslogd messages is a local UNIX file system file. By default, syslogd will not create log files if they do not exist. However, if the –c option is used when starting syslogd, syslogd will create the destination file if it does not exist. syslogd will also create directories if needed to contain the file. The ability to create files dynamically allows for improved organization since log files can be named using current dates or days of the week. However, the ownership and access permissions for the files and directories created by syslogd has always been fixed and not user-modifiable. The owner of files and directories created by syslogd is always UID 0 since syslogd must run under a UID 0 userid. The default permissions used by syslogd when creating files or directories require that any user that needs access to the files must also be UID 0. Administrators of z/OS systems want to minimize the number of users that have UID 0 access.

Syslogd creates files with permissions bits of 0600 (octal). This octal value indicates that the owner of the file has: read access and write access. Syslogd creates directories with the permissions bits of 0700 (octal). This octal value indicates that the owner of the directory has: read access, write access and traverse access. See the description of the chmod command in the UNIX System Services Command Reference book for more information on permissions for directories and files. The syslog daemon must always be run by a userid that has UID 0.



Control syslogd Directory and File permissions

- Allow system programmer to specify the default permissions to be used by syslogd when creating directories and files with two new start options

-D octal_value	default permissions for directories
-F octal_value	default permissions for files

- Allow the default permissions to be overridden on a rule-by-rule basis if needed. Two new configuration options on rules with file destinations

-D octal_value	permission to use for directory
-F octal_value	permission to use for file

  - Override the corresponding values specified when starting syslogd
- -D and –F are only valid when the destination is a regular UNIX file system file
- You may specify –F and –D independently, together or not at all
- You must use the –c (dynamic file create) start option in order to use the new –D or –F start or configuration options.
- Directories and files that already exist are not affected by this new function.
- Before using –F or –D you must carefully consider and understand what permissions values are safe and appropriate for your systems.

262

Redbooks ibm.com/redbooks

With this solution, UID 0 userids are no longer required in order to access syslogd log files. With z/OS V1R9, syslogd can be started with two new start options. The –D option allows the system programmer to specify the default permissions that should be used by syslogd when creating a directory. The –F option allows the system programmer to specify the default permissions that should be used by syslogd when creating a file. These are optional and one or both may be used.

Additionally, the default directory and file permissions can be overridden on a rule-by-rule basis. The –D option is used to specify the permission bits to use for creating a directory for the rule. The –F option is used to specify the permission bits to be used when creating a file for the rule. These are optional and one or both may be used. If used, the values override the corresponding start option values for –D and –F. These options are only valid when the destination is a regular USS file system file. You may specify –D or –F independently or together. If a value is not overridden on a rule with –D or –F, the value from the corresponding start option is used. If the corresponding start option was not specified, then the default value is used. The default permissions for directories is 0700 octal and the default permissions value for files is 0600 octal. Note that the new options are case-sensitive and must be entered in uppercase.

As always, in order for syslogd to create any directories or files, the –c start option must be used.

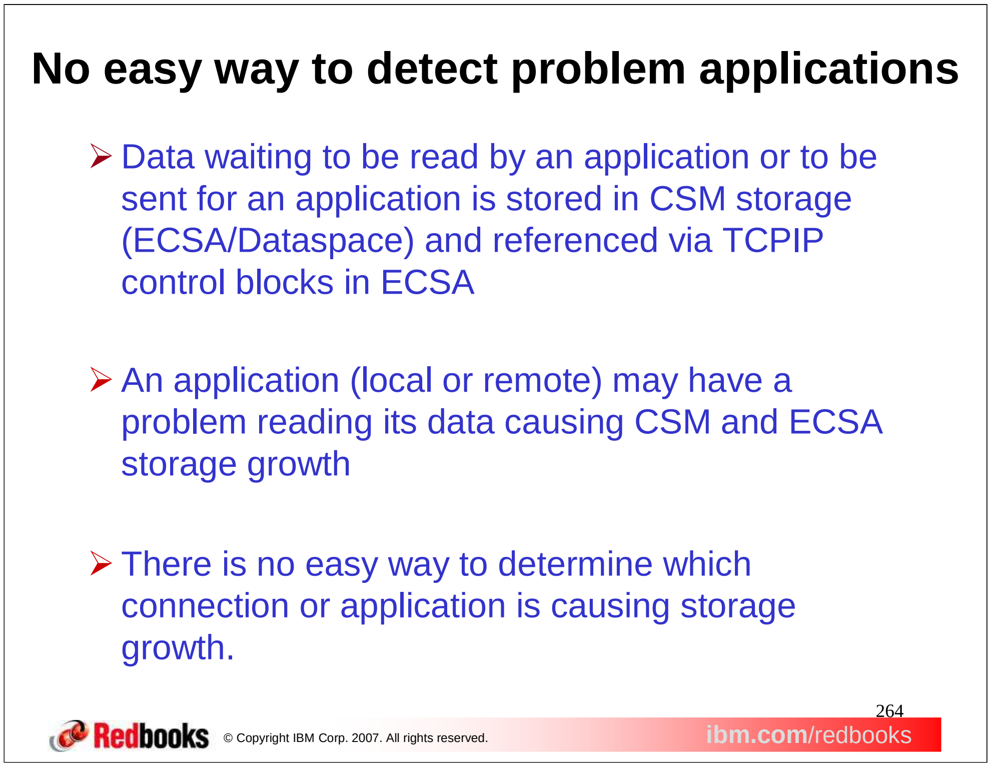
You must use the –c option in order to use the new –D and –F start options or the new –D and –F configuration options. Directories and files that already exist are not affected by the –D or –F start options or configuration options. Before changing the default values for directory and file permissions you must be sure you understand what values are safe and appropriate for your particular systems.

Enhance NETSTAT ALL/-A report to indicate sockets storage use

263

ibm.com/redbooks

This section describes the enhancements made to the Netstat ALL/-A commands to assist in determining a sockets storage usage.



No easy way to detect problem applications

- Data waiting to be read by an application or to be sent for an application is stored in CSM storage (ECSA/Dataspace) and referenced via TCPIP control blocks in ECSA

- An application (local or remote) may have a problem reading its data causing CSM and ECSA storage growth

- There is no easy way to determine which connection or application is causing storage growth.

For each TCP connection TCPIP maintains a send and receive buffer.   The size used is specified on the TCPCONFIG parameters TCPRCVBufrSize and TCPSENDBfrsize or overridden by the application by using the SO_SNDBUF and SO_RCVBUF options in a SetSockopt command.

The size of the receive buffer is passed to the remote side of the connection during connection establishment as the window size. The window size is used to control how much data the remote side can send and varies during the connection as data accumulates on the receive queue waiting for the local application to issue read requests.

The size of the send buffer is used to control the amount of data being sent by the application to the remote host. If TCPIP is unable to immediately send the data when asked by the application (usually because the remote side has lowered their window size) then TCPIP will hold the data on the send queue. Once the queue reaches the send buffer size then TCPIP will not honor any additional send requests from the application on that connection until TCPIP is able to send some of the data already waiting on the send queue.

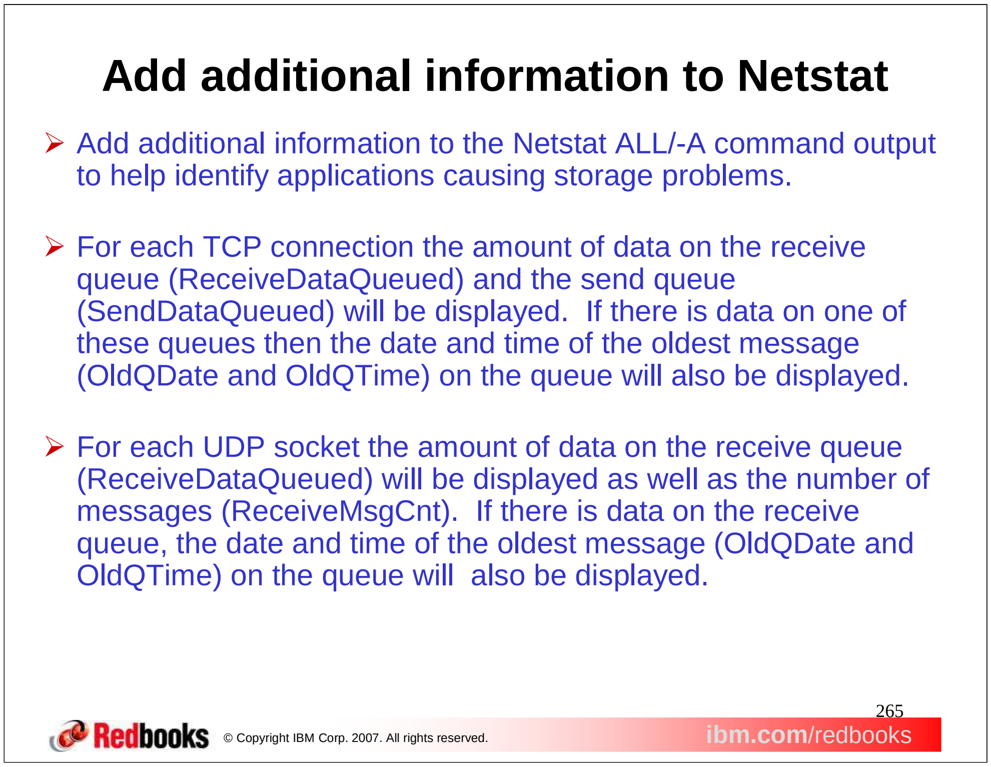
For each UDP socket TCPIP does not queue send data but will queue up received data waiting for the application to read it.   The only limit on how much received data can be queued is by specifying UDPQUEUELIMIT on the UDPCONFIG statement, otherwise there is no limit on how much can be queued.

For both UDP sockets and TCP connections the message data is stored in CSM buffers. These may be in either ECSA or dataspace.   Each message also has a control block structure that points to the message and contains information about the message and queue pointers. This control block is stored in ECSA.

TCPIP related ECSA and CSM storage growth may be attributed to application problems.   A problem application may have stopped issuing reads for data for some reason or may be running too slowly to keep up with the speed that data is being received from it’s remote partner. This can occur on either side of the connection and either can affect storage build up on z/OS TCPIP.

Note: the storage accumulating in these cases will be greater than just the message data on the send/receive queues since there are control blocks that are used to maintain these queues. Also, since multiple messages may be contained within a single CSM buffer, one held message can prevent the entire buffer from being released.

Currently there is not a way to easily tell if a CSM and ECSA storage growth is due to an application read problem and to identify which application.



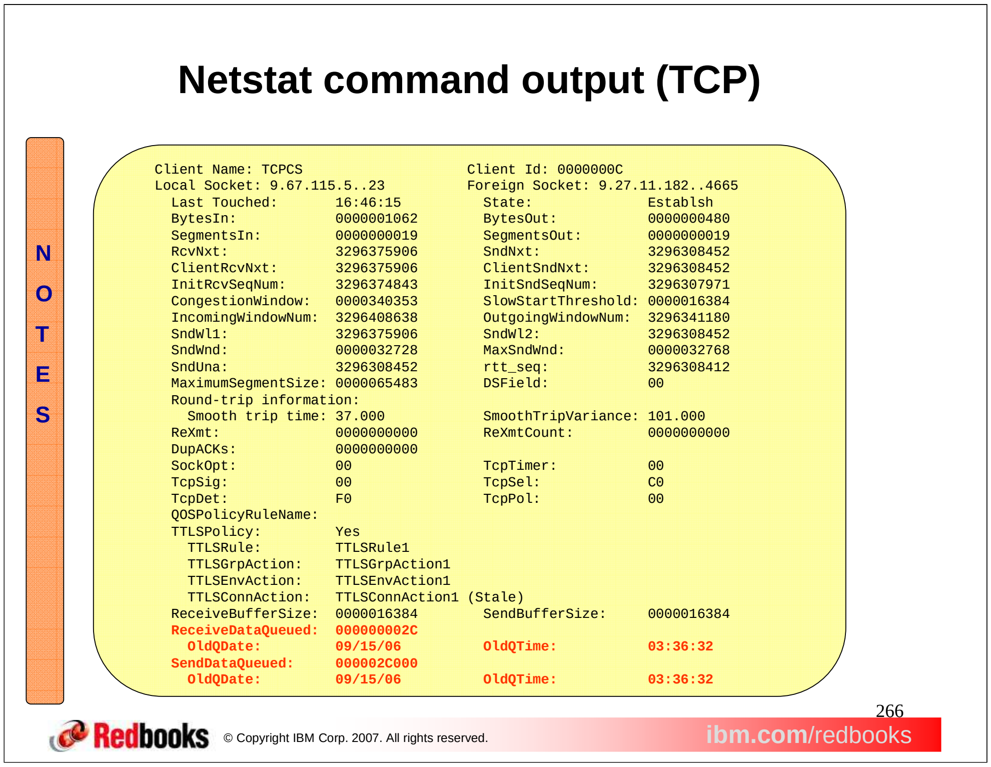
Add additional information to Netstat

- Add additional information to the Netstat ALL/-A command output to help identify applications causing storage problems.
- For each TCP connection the amount of data on the receive queue (ReceiveDataQueued) and the send queue (SendDataQueued) will be displayed. If there is data on one of these queues then the date and time of the oldest message (OldQDate and OldQTime) on the queue will also be displayed.
- For each UDP socket the amount of data on the receive queue (ReceiveDataQueued) will be displayed as well as the number of messages (ReceiveMsgCnt). If there is data on the receive queue, the date and time of the oldest message (OldQDate and OldQTime) on the queue will also be displayed.

Redbooks

ibm.com/redbooks

The Netstat ALL/-A command output provides detailed information about all connections. Additional messages have been added to the TCP connection and UDP socket information to specifically list receive and send queue data byte counts and the date and time of the oldest messages on these queues. For UDP sockets TCPIP maintains a count of the number of messages on the receive queue and this data is also now displayed.

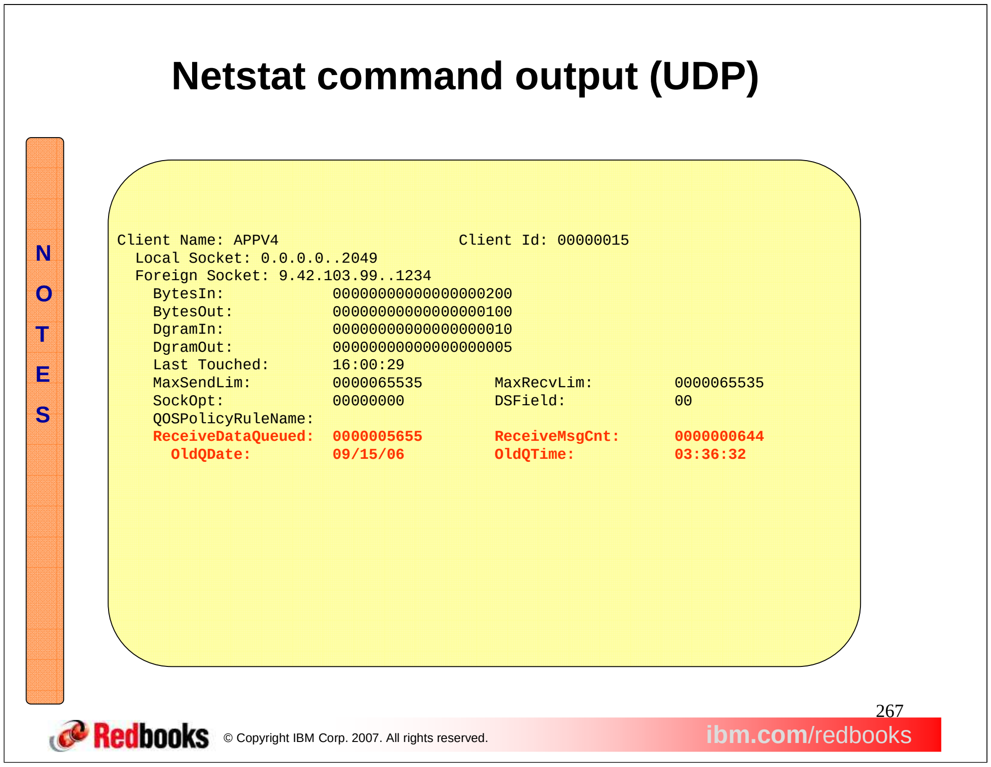


Netstat command output (TCP)

NOTES

```
Client Name: TCPCS                    Client Id: 0000000C
Local Socket: 9.67.115.5..23          Foreign Socket: 9.27.11.182..4665
  Last Touched:       16:46:15          State:              Establsh
  BytesIn:            0000001062        BytesOut:           0000000480
  SegmentsIn:         0000000019        SegmentsOut:        0000000019
  RcvNxt:             3296375906        SndNxt:             3296308452
  ClientRcvNxt:       3296375906        ClientSndNxt:       3296308452
  InitRcvSeqNum:      3296374843        InitSndSeqNum:      3296307971
  CongestionWindow:   0000340353        SlowStartThreshold: 0000016384
  IncomingWindowNum:  3296408638        OutgoingWindowNum:  3296341180
  SndWl1:             3296375906        SndWl2:             3296308452
  SndWnd:             0000032728        MaxSndWnd:          0000032768
  SndUna:             3296308452        rtt_seq:            3296308412
  MaximumSegmentSize: 0000065483        DSField:            00
  Round-trip information:
    Smooth trip time: 37.000            SmoothTripVariance: 101.000
  ReXmt:              0000000000        ReXmtCount:         0000000000
  DupACKs:            0000000000
  SockOpt:            00                TcpTimer:           00
  TcpSig:             00                TcpSel:             C0
  TcpDet:             F0                TcpPol:             00
  QOSPolicyRuleName:
  TTLSPolicy:         Yes
    TTLSRule:         TTLSRule1
    TTLSGrpAction:    TTLSGrpAction1
    TTLSEnvAction:    TTLSEnvAction1
    TTLSConnAction:   TTLSConnAction1 (Stale)
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ReceiveDataQueued:  000000002C
    OldQDate:         09/15/06          OldQTime:           03:36:32
  SendDataQueued:     000002C000
    OldQDate:         09/15/06          OldQTime:           03:36:32
```

This slide shows the output for a TCP connection. New messages/fields are shown in RED. If there is no data on a specific queue then the OldQDate and OldQTime message will not be displayed for that queue.



Netstat command output (UDP)

N
O
T
E
S

```
Client Name: APPV4                    Client Id: 00000015
  Local Socket: 0.0.0.0..2049
  Foreign Socket: 9.42.103.99..1234
    BytesIn:            00000000000000000200
    BytesOut:           00000000000000000100
    DgramIn:            00000000000000000010
    DgramOut:           00000000000000000005
    Last Touched:       16:00:29
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicyRuleName:
    ReceiveDataQueued:  0000005655        ReceiveMsgCnt:      0000000644
      OldQDate:         09/15/06          OldQTime:           03:36:32
```

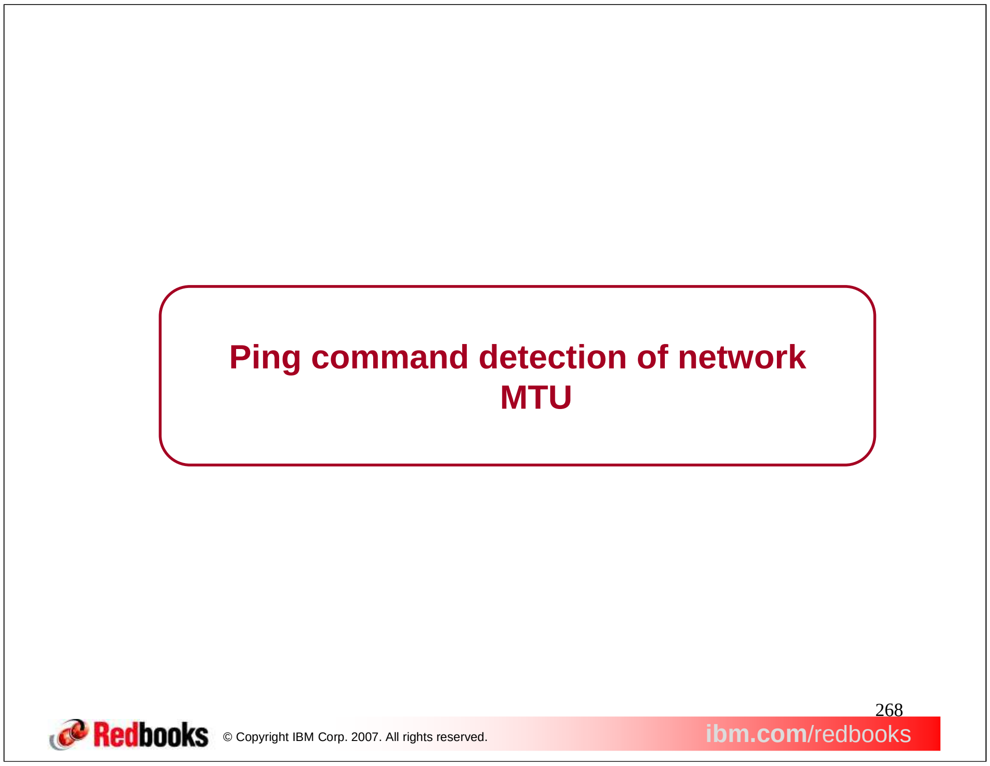
267

Redbooks ibm.com/redbooks

This slide shows the output for a UDP socket.

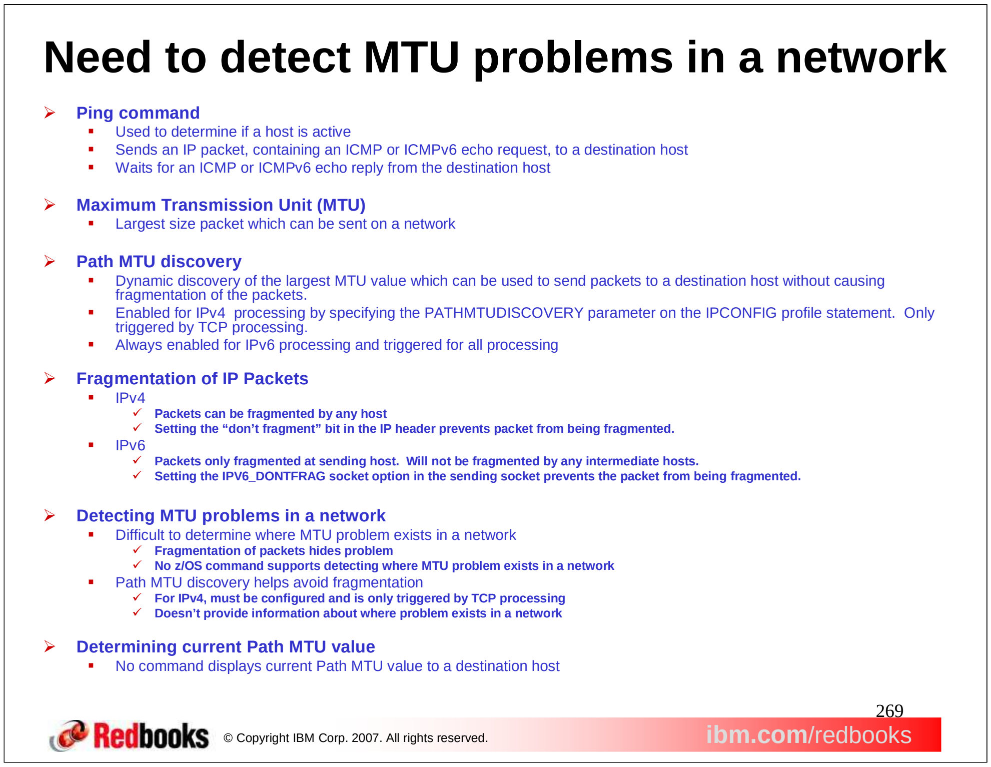
New messages/fields are shown in RED.

If there is no data on a specific queue then the OldQDate and OldQTime message will not be displayed for that queue.



Ping command detection of network MTU

This section describes the changes to the Ping command to provide detection of network MTU size.



Need to detect MTU problems in a network

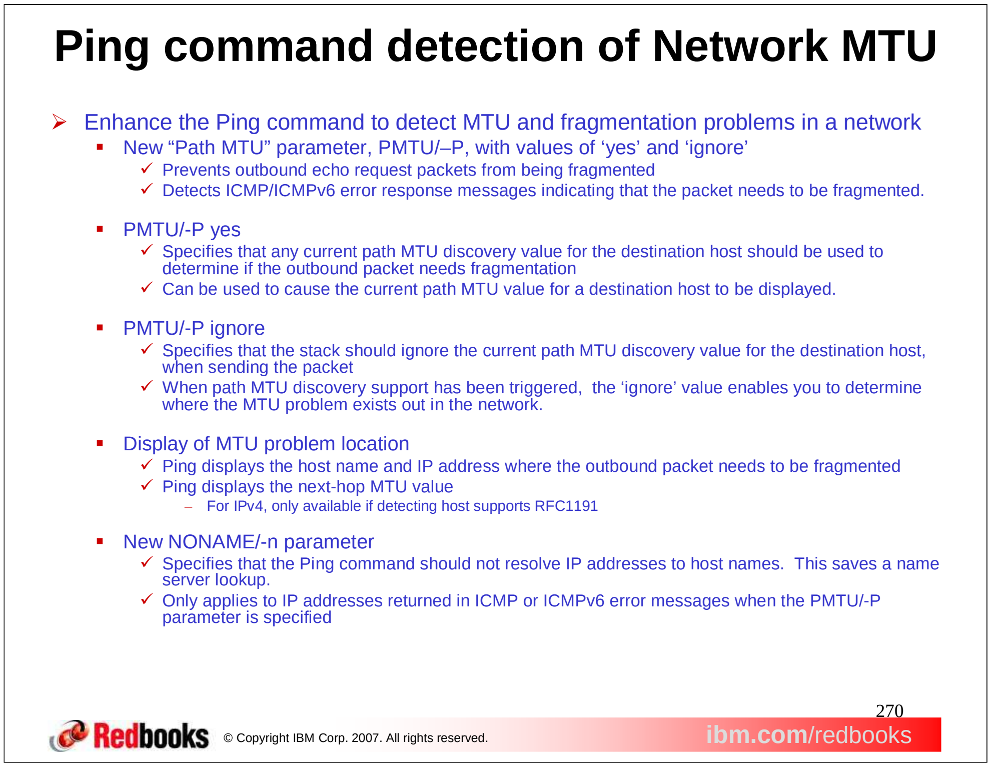
- **Ping command**
  - Used to determine if a host is active
  - Sends an IP packet, containing an ICMP or ICMPv6 echo request, to a destination host
  - Waits for an ICMP or ICMPv6 echo reply from the destination host
- **Maximum Transmission Unit (MTU)**
  - Largest size packet which can be sent on a network
- **Path MTU discovery**
  - Dynamic discovery of the largest MTU value which can be used to send packets to a destination host without causing fragmentation of the packets.
  - Enabled for IPv4 processing by specifying the PATHMTUDISCOVERY parameter on the IPCONFIG profile statement. Only triggered by TCP processing.
  - Always enabled for IPv6 processing and triggered for all processing
- **Fragmentation of IP Packets**
  - IPv4
    - ✓ **Packets can be fragmented by any host**
    - ✓ **Setting the “don’t fragment” bit in the IP header prevents packet from being fragmented.**
  - IPv6
    - ✓ **Packets only fragmented at sending host. Will not be fragmented by any intermediate hosts.**
    - ✓ **Setting the IPV6_DONTFRAG socket option in the sending socket prevents the packet from being fragmented.**
- **Detecting MTU problems in a network**
  - Difficult to determine where MTU problem exists in a network
    - ✓ **Fragmentation of packets hides problem**
    - ✓ **No z/OS command supports detecting where MTU problem exists in a network**
  - Path MTU discovery helps avoid fragmentation
    - ✓ **For IPv4, must be configured and is only triggered by TCP processing**
    - ✓ **Doesn’t provide information about where problem exists in a network**
- **Determining current Path MTU value**
  - No command displays current Path MTU value to a destination host

269

Redbooks ibm.com/redbooks

The Ping command is used to determine if a host is active. The Maximum Transmission Unit (MTU) is the largest size packet which can be sent on a network. Path MTU discovery is the process of dynamically determining the largest MTU value which can be used to send packets to a destination host without causing fragmentation of the packets.

MTU problems can exist in large networks. The problem occurs where the MTU for a segment of the network is smaller then the network segments to which it is connected. Detecting MTU problems is difficult because, when the problem occurs, the IPv4 packets are normally fragmented. And z/OS CS currently does not provide any command to detect MTU problems. Path MTU discovery support can help avoid fragmentation by determining the smallest MTU value for the path to a destination host. But it does not provide information about where the problem is located in the network.



Ping command detection of Network MTU

- Enhance the Ping command to detect MTU and fragmentation problems in a network
  - New “Path MTU” parameter, PMTU/–P, with values of ‘yes’ and ‘ignore’
    - ✓ Prevents outbound echo request packets from being fragmented
    - ✓ Detects ICMP/ICMPv6 error response messages indicating that the packet needs to be fragmented.
  - PMTU/-P yes
    - ✓ Specifies that any current path MTU discovery value for the destination host should be used to determine if the outbound packet needs fragmentation
    - ✓ Can be used to cause the current path MTU value for a destination host to be displayed.
  - PMTU/-P ignore
    - ✓ Specifies that the stack should ignore the current path MTU discovery value for the destination host, when sending the packet
    - ✓ When path MTU discovery support has been triggered, the ‘ignore’ value enables you to determine where the MTU problem exists out in the network.
  - Display of MTU problem location
    - ✓ Ping displays the host name and IP address where the outbound packet needs to be fragmented
    - ✓ Ping displays the next-hop MTU value
      - – For IPv4, only available if detecting host supports RFC1191
  - New NONAME/-n parameter
    - ✓ Specifies that the Ping command should not resolve IP addresses to host names. This saves a name server lookup.
    - ✓ Only applies to IP addresses returned in ICMP or ICMPv6 error messages when the PMTU/-P parameter is specified

270

Redbooks

ibm.com/redbooks

In V1R9 we have provided a method for determining where in the network an MTU problem exists, and for determining what the current path MTU value is to a destination host.

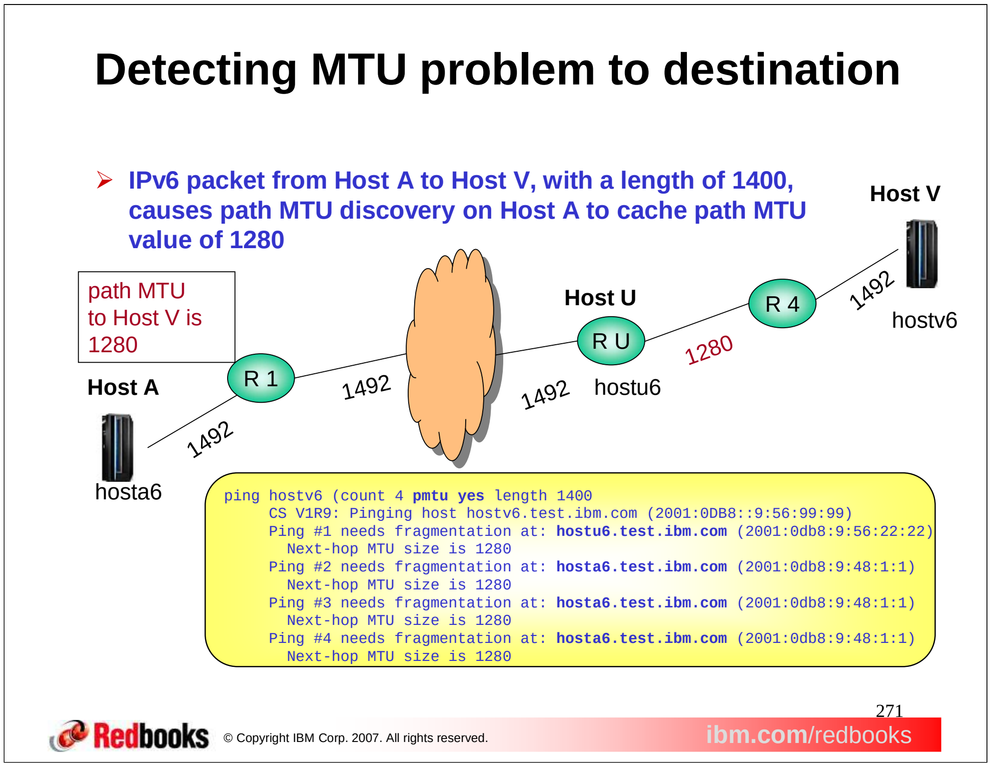
A new parameter, TSO PMTU or z/OS UNIX –P, has been added to the Ping command. Specifying this parameter prevents the outbound echo request packets from being fragmented. The values for this parameter are ‘yes’ and ‘ignore’.

A value of ‘yes’ specifies that the current path MTU value to the destination should be used to determine if the outbound packet needs to be fragmented. If path MTU discovery is not enabled, or has not been triggered, then the MTU associated with the outbound route is used to determine if the packet needs to be fragmented.

A value of ‘ignore’ specifies that any path MTU value to the destination should be ignored and, as long as the packet is not too large for the outbound interface, the packet should be sent out to the destination.

If the packet needs to be fragmented, either at the sending TCP/IP stack, or out in the network, the Ping command displays the host name and IP address of the host where fragmentation is needed. It also displays the next-hop MTU value returned by the host where fragmentation is needed. Hosts which support RFC1191 (Path MTU Discovery) should return the next-hop MTU value in the ICMP/ICMPv6 error message.

Another new parameter, TSO NONAME or z/OS UNIX –n, has been added to the Ping command. Specifying this parameter causes the Ping command to bypass the name server lookup of the host name, for the IP address of the host where fragmentation is needed. This Can be useful when the name server is slow or unresponsive or the environment does not support DNS reverse lookups (IP address to hostname). This parameter is only in effect when the PMTU or –P parameter is specified.



Detecting MTU problem to destination

- ➤ IPv6 packet from Host A to Host V, with a length of 1400, causes path MTU discovery on Host A to cache path MTU value of 1280

path MTU
to Host V is
1280

```
ping hostv6 (count 4 pmtu yes length 1400
     CS V1R9: Pinging host hostv6.test.ibm.com (2001:0DB8::9:56:99:99)
     Ping #1 needs fragmentation at: hostu6.test.ibm.com (2001:0db8:9:56:22:22)
       Next-hop MTU size is 1280
     Ping #2 needs fragmentation at: hosta6.test.ibm.com (2001:0db8:9:48:1:1)
       Next-hop MTU size is 1280
     Ping #3 needs fragmentation at: hosta6.test.ibm.com (2001:0db8:9:48:1:1)
       Next-hop MTU size is 1280
     Ping #4 needs fragmentation at: hosta6.test.ibm.com (2001:0db8:9:48:1:1)
       Next-hop MTU size is 1280
```

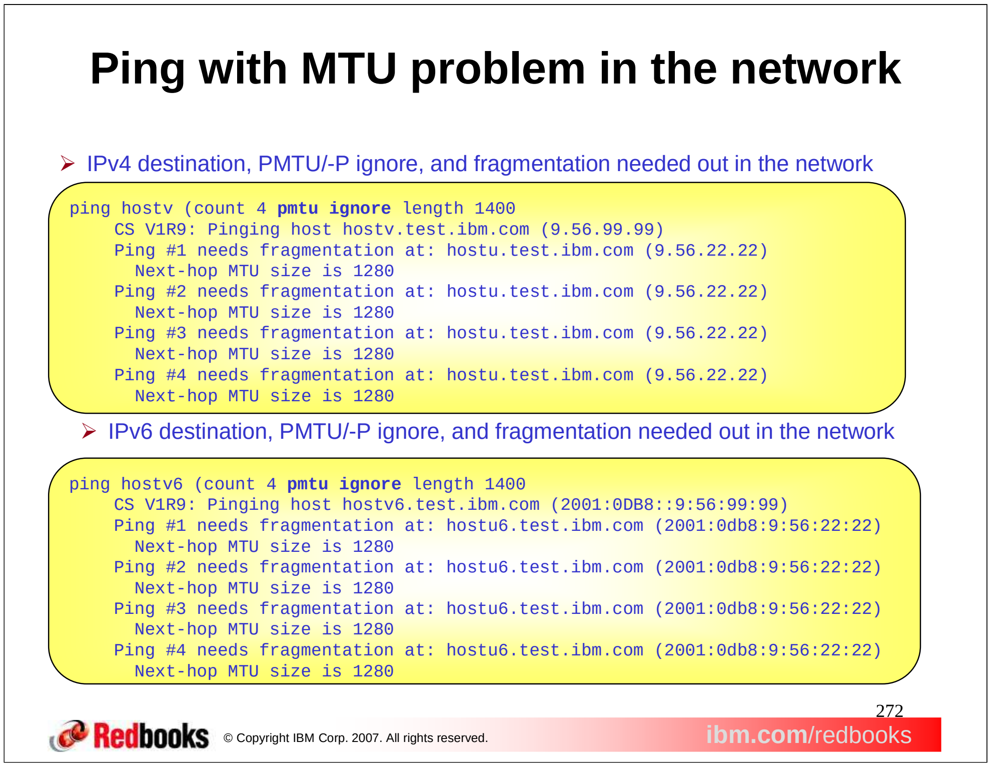
271

ibm.com/redbooks

This example shows the output from the Ping command which could have caused the path MTU value of 1280 to be set. Path MTU discovery is always enabled for IPV6 processing. So when Host U indicated that fragmentation was needed for packet #1, with a next-hop MTU size of 1280, this value was cached as the path MTU size to Host V.

Since the **PMTU YES** parameter was specified on the Ping command, the outbound Ping packets will not be fragmented and the TCP/IP stack will use the path MTU value to compare against the packet length.

So, when the packet #2 is sent out, the TCP/IP stack on Host A compares the size of the packet to the cached path MTU value and fails the send request because the packet is too large to be sent without fragmenting it. This same error occurs for packets #3 and #4. The cached path MTU value is in effect for 20 minutes (assuming that, during that time, no other ICMPv6 error messages are received to decrease the path MTU size to Host V).



Ping with MTU problem in the network

- IPv4 destination, PMTU/-P ignore, and fragmentation needed out in the network

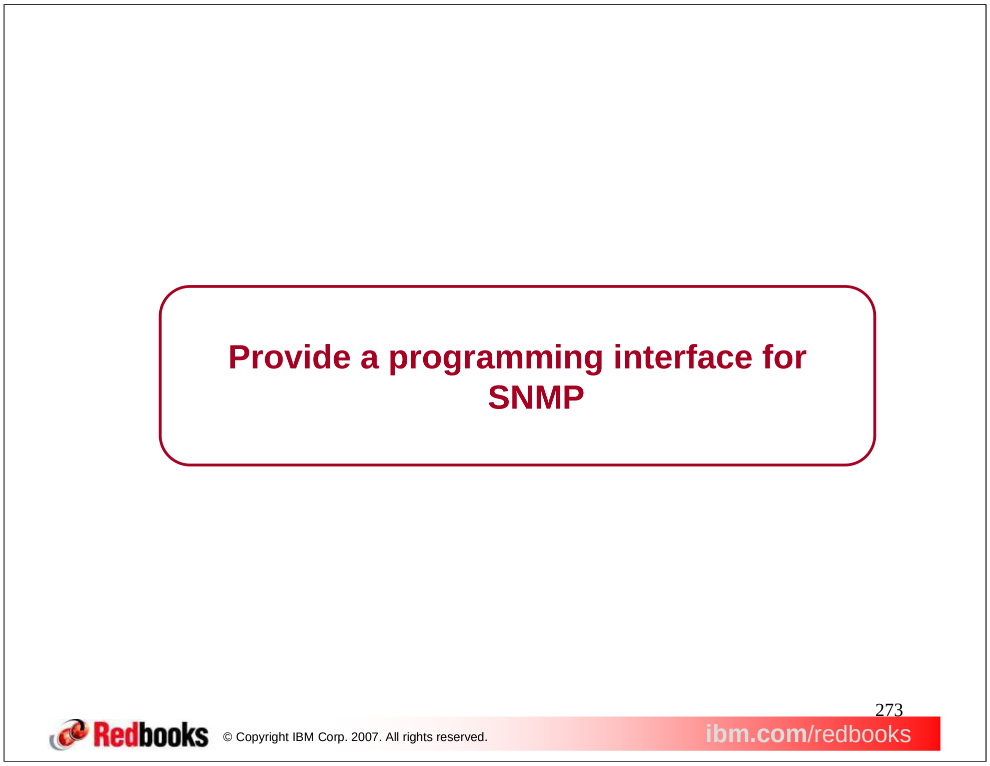
```
ping hostv (count 4 pmtu ignore length 1400
    CS V1R9: Pinging host hostv.test.ibm.com (9.56.99.99)
    Ping #1 needs fragmentation at: hostu.test.ibm.com (9.56.22.22)
      Next-hop MTU size is 1280
    Ping #2 needs fragmentation at: hostu.test.ibm.com (9.56.22.22)
      Next-hop MTU size is 1280
    Ping #3 needs fragmentation at: hostu.test.ibm.com (9.56.22.22)
      Next-hop MTU size is 1280
    Ping #4 needs fragmentation at: hostu.test.ibm.com (9.56.22.22)
      Next-hop MTU size is 1280
```

- IPv6 destination, PMTU/-P ignore, and fragmentation needed out in the network

```
ping hostv6 (count 4 pmtu ignore length 1400
    CS V1R9: Pinging host hostv6.test.ibm.com (2001:0DB8::9:56:99:99)
    Ping #1 needs fragmentation at: hostu6.test.ibm.com (2001:0db8:9:56:22:22)
      Next-hop MTU size is 1280
    Ping #2 needs fragmentation at: hostu6.test.ibm.com (2001:0db8:9:56:22:22)
      Next-hop MTU size is 1280
    Ping #3 needs fragmentation at: hostu6.test.ibm.com (2001:0db8:9:56:22:22)
      Next-hop MTU size is 1280
    Ping #4 needs fragmentation at: hostu6.test.ibm.com (2001:0db8:9:56:22:22)
      Next-hop MTU size is 1280
```

This example shows the output of the Ping command which was used to detect the network MTU problem in the network on the previous slide. We invoked the Ping command on Host A, with a destination host of Host V. Since we suspected that the problem was out in the network somewhere, we specified **PMTU IGNORE** so that the packet would be sent out, even if path MTU discovery had determined that the path MTU size was smaller than the length of 1400.

The Ping responses show that the MTU problem is at Host U. And the next-hop MTU size from Host U to the network segment which leads to Host V is 1280.

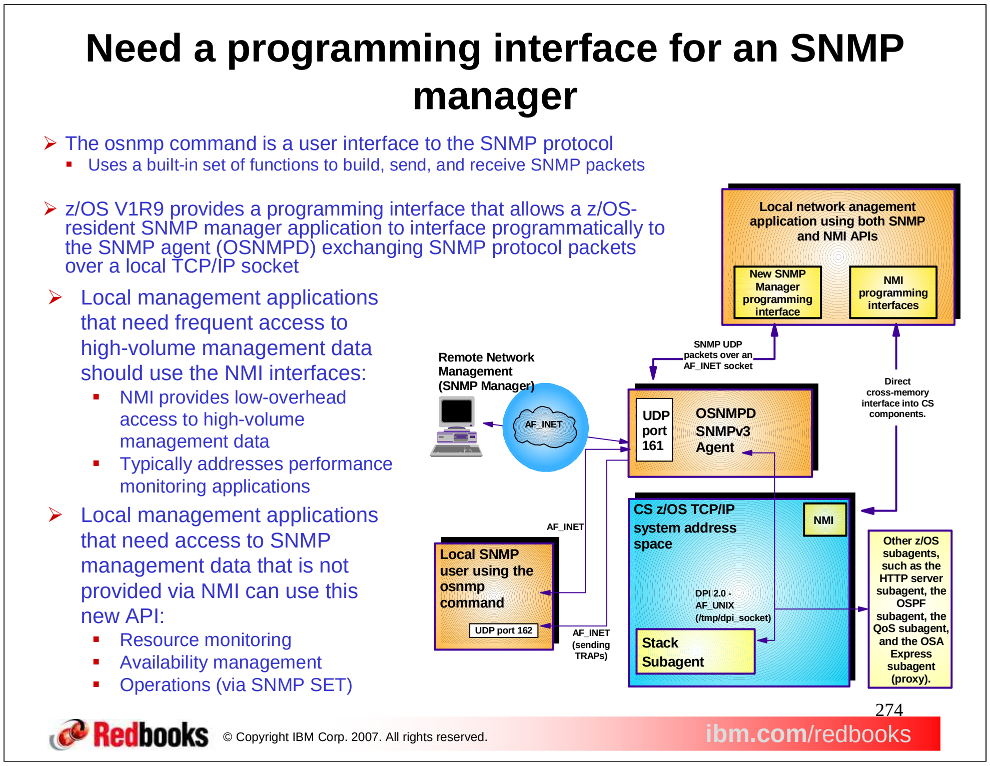


Provide a programming interface for SNMP

273

ibm.com/redbooks

This section describes the SNMP Manager API and the SNMP Notification API, which can be used to build an SNMP Manager application.



Need a programming interface for an SNMP manager

- The osnmp command is a user interface to the SNMP protocol
  - Uses a built-in set of functions to build, send, and receive SNMP packets
- z/OS V1R9 provides a programming interface that allows a z/OS-resident SNMP manager application to interface programmatically to the SNMP agent (OSNMPD) exchanging SNMP protocol packets over a local TCP/IP socket
- Local management applications that need frequent access to high-volume management data should use the NMI interfaces:
  - NMI provides low-overhead access to high-volume management data
  - Typically addresses performance monitoring applications
- Local management applications that need access to SNMP management data that is not provided via NMI can use this new API:
  - Resource monitoring
  - Availability management
  - Operations (via SNMP SET)

Local network agement application using both SNMP and NMI APIs
New SNMP Manager programming interface
NMI programming interfaces
SNMP UDP packets over an AF_INET socket
Direct cross-memory interface into CS components.
Remote Network Management (SNMP Manager)
AF_INET
UDP port 161
OSNMPD SNMPv3 Agent
AF_INET
CS z/OS TCP/IP system address space
NMI
Local SNMP user using the osnmp command
UDP port 162
AF_INET (sending TRAPs)
DPI 2.0 - AF_UNIX (/tmp/dpi_socket)
Stack Subagent
Other z/OS subagents, such as the HTTP server subagent, the OSPF subagent, the QoS subagent, and the OSA Express subagent (proxy).

274

Redbooks ibm.com/redbooks

SNMP is a standard's-based protocol for network management that is based upon the TCP/IP protocol (UDP). Part of the SNMP standards also includes a database structure specification for management objects called MIBs (Management Information Base). Use of SNMP is widespread, and work continues in the IETF mostly in the area of defining new MIBs. The SNMP protocol has been evolving for many years and has yielded several levels of the protocol, some of which were never adopted. Primarily, the supported protocol levels are: SNMPv1, SNMPv2c, and SNMPv3. SNMPv3 defines a user-based security model for SNMP, rather than the community-based model of SNMPv1 and SNMPv2c.

The key management entities of SNMP are:

•Agent - This entity implements the SNMP protocol stack and routes requests from managers to the appropriate subagents. Sometimes it is called the engine. It communicates with the subagents using the Distributed Protocol Interface (DPI) and with the Managers using the SNMP protocol. Subagents register their MIB objects with the Agent. For Comm Server, the agent is our osnmp daemon.

•Subagent - These entities are the providers of the MIB data. They communicate with the SNMP agents. In Comm Server, an example is the TCP/IP Subagent.

•Manager- These entities communicate using SNMP protocol requests with SNMP agents to retrieve management data. Comm Server only provides a command-line manager, the osnmp command. The manager function is typically part of management applications, such as Netview.

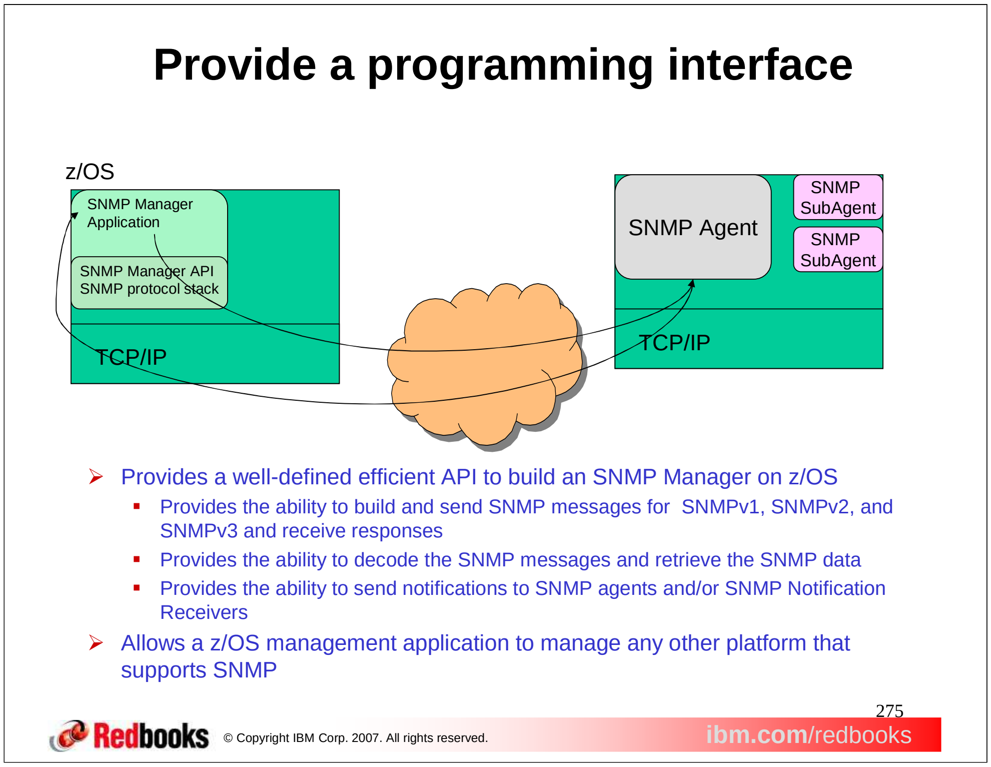
z/OS has not provided a formal, standards-based SNMP API for customer-driven manager applications. The osnmp command currently being shipped is, itself, a manager application, but customers have not been able to create their own application to manage SNMP, due to the absence of an external API.

z/OS Communications Server V1R9 provides a new application programming interface (API), the SNMP Manager API, for writing SNMP managers. Management application developers can use this API to build SNMP management functions to retrieve SNMP management data. This API provides the following functions:

•The ability to build and send SNMP messages for SNMPv1, SNMPv2, and SNMPv3 and receive responses

•The ability to decode the SNMP messages and retrieve the SNMP data

z/OS Communications Server also provides an extension of the SNMP Manager API, the SNMP Notification API, which leverages the functionality of the SNMP Manager API to send notifications to SNMP agents and/or SNMP Notification Receivers. Available notifications include Informs and both Version 1 and Version 2 Traps.



Provide a programming interface

z/OS

SNMP Manager Application

SNMP Manager API
SNMP protocol stack

TCP/IP

SNMP Agent

SNMP SubAgent

SNMP SubAgent

TCP/IP

- Provides a well-defined efficient API to build an SNMP Manager on z/OS
  - Provides the ability to build and send SNMP messages for SNMPv1, SNMPv2, and SNMPv3 and receive responses
  - Provides the ability to decode the SNMP messages and retrieve the SNMP data
  - Provides the ability to send notifications to SNMP agents and/or SNMP Notification Receivers
- Allows a z/OS management application to manage any other platform that supports SNMP

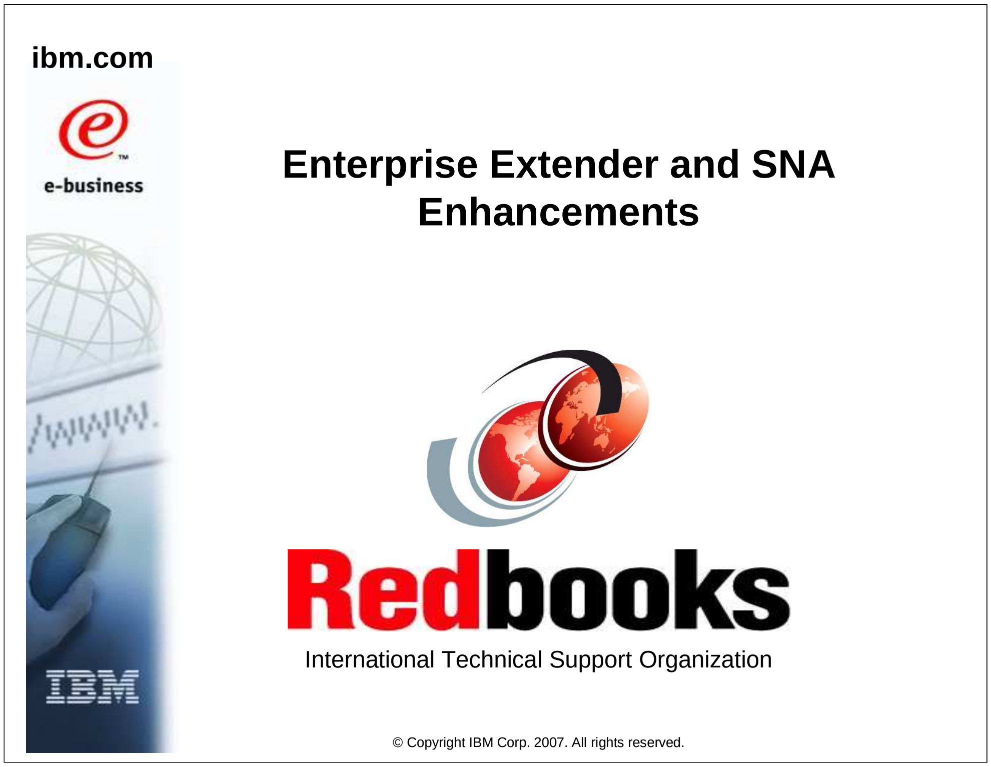
Redbooks

ibm.com/redbooks

z/OS Communications Server provides a new application programming interface (API), the SNMP Manager API, for writing SNMP managers. Management application developers can use this API to build SNMP management functions to retrieve SNMP management data.

z/OS Communications Server is also providing an extension of the SNMP Manager API, the SNMP Notification API, which leverages the functionality of the SNMP Manager API to send notifications to SNMP agents and/or SNMP Notification Receivers. Available notifications include Informs and both Version 1 and Version 2 Traps.

This API will allow z/OS management applications to manage any other platforms that support SNMP. This slide shows that an SNMP management application, running in z/OS, can manage any other platform that supports SNMP.

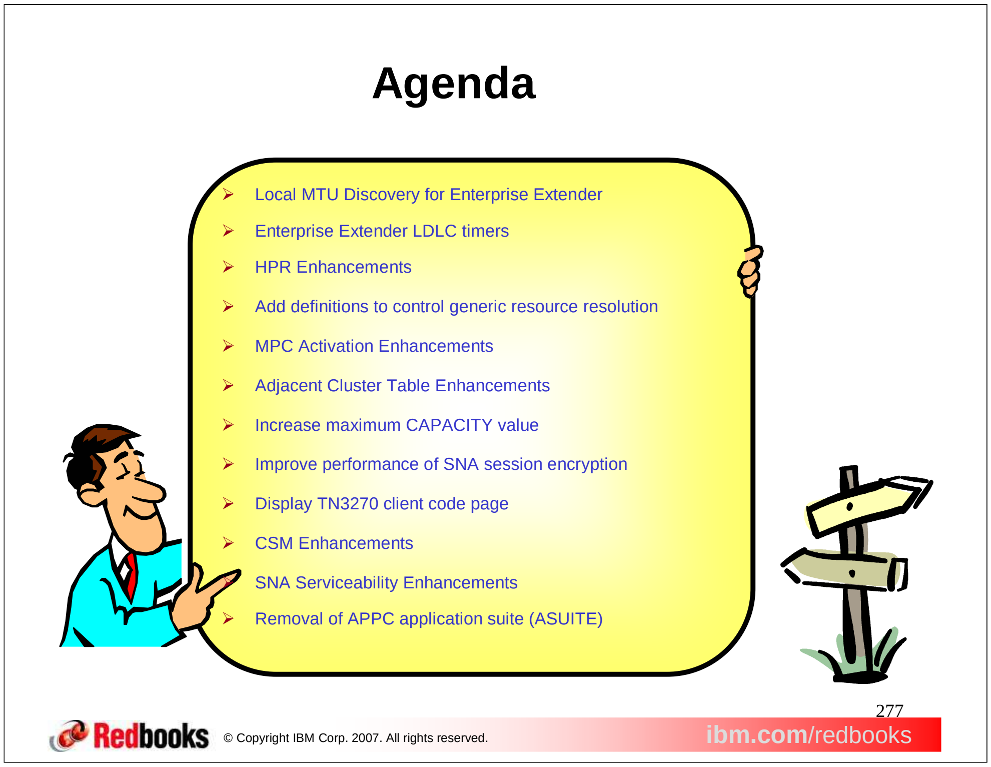


ibm.com

Enterprise Extender and SNA Enhancements

International Technical Support Organization

This presentation describes the Enterprise Extender and SNA enhancements for z/OS V1R9 Communications Server.



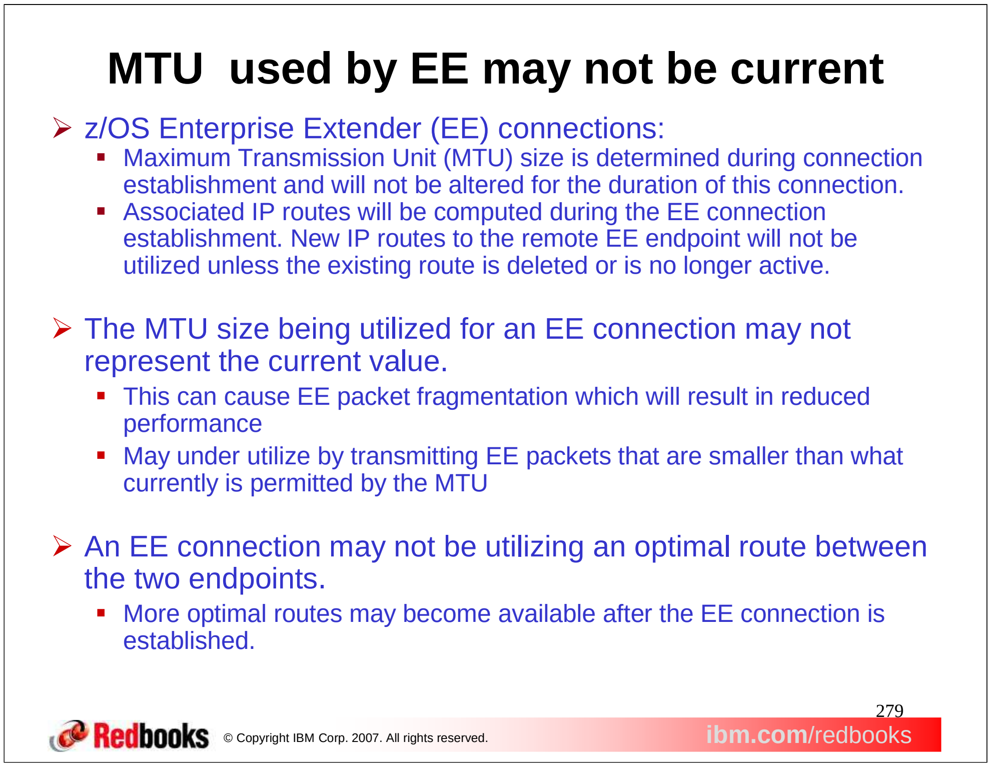
Agenda

- Local MTU Discovery for Enterprise Extender
- Enterprise Extender LDLC timers
- HPR Enhancements
- Add definitions to control generic resource resolution
- MPC Activation Enhancements
- Adjacent Cluster Table Enhancements
- Increase maximum CAPACITY value
- Improve performance of SNA session encryption
- Display TN3270 client code page
- CSM Enhancements
- SNA Serviceability Enhancements
- Removal of APPC application suite (ASUITE)

z/OS V1R9 Communications Server contains multiple Enterprise Extender and SNA enhancements. Each of these enhancements will be discussed in this presentation.

Local MTU Discovery for Enterprise Extender

This section describes the new capabilities to detect changes in an EE connection’s Maximum Transmission Unit (MTU) size and underlying IP route.



MTU used by EE may not be current

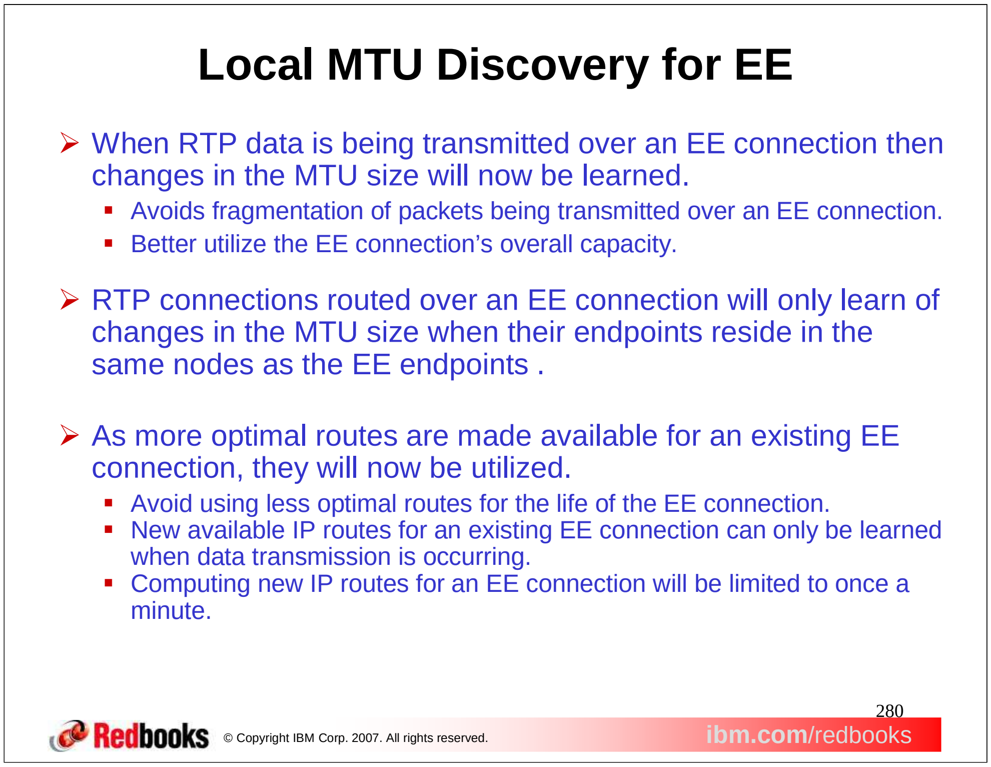
- z/OS Enterprise Extender (EE) connections:
  - Maximum Transmission Unit (MTU) size is determined during connection establishment and will not be altered for the duration of this connection.
  - Associated IP routes will be computed during the EE connection establishment. New IP routes to the remote EE endpoint will not be utilized unless the existing route is deleted or is no longer active.
- The MTU size being utilized for an EE connection may not represent the current value.
  - This can cause EE packet fragmentation which will result in reduced performance
  - May under utilize by transmitting EE packets that are smaller than what currently is permitted by the MTU
- An EE connection may not be utilizing an optimal route between the two endpoints.
  - More optimal routes may become available after the EE connection is established.

Redbooks

ibm.com/redbooks

In prior releases an Enterprise Extender (EE) connection will obtain the minimum MTU size permitted for packets being transmitted to a remote EE endpoint at the time that the connection is **initially** established. It is important that VTAM understand the maximum MTU size for this connection to avoid fragmentation by the TCP/IP stack which will greatly increase the path length for a transmitted EE packets in the TCP/IP stack. Therefore today if a new interface is being utilized by TCP/IP for an existing EE connection (previous route obtained could have been deleted or became inactive) and the new MTU size is smaller than was previously reported to VTAM (at EE connection initialization) then there are negative performance implications.

Additionally there are changes needed to allow TCP/IP to attempt to obtain a new route for an existing EE connection when updates have been made to the IP routing table (via OMPROUTE, policy changes etc.). When changes have been made to the IP routing table a more optimal route can perhaps be determined for an EE connection. Currently if TCP/IP obtains a route handle for an EE connection and that route is associated with a default route then there is no way to ever move from this default route without SNA terminating the EE connection. Therefore there have been a number of users that start their VTAM and TCP/IP connection where an EE connection is initiated by VTAM prior to TCP/IP learning about all of the potential routes from OMPROUTE and therefore the EE connection ends up utilizing the default route (which in many cases is not the optimal route).



Local MTU Discovery for EE

- When RTP data is being transmitted over an EE connection then changes in the MTU size will now be learned.
  - Avoids fragmentation of packets being transmitted over an EE connection.
  - Better utilize the EE connection’s overall capacity.
- RTP connections routed over an EE connection will only learn of changes in the MTU size when their endpoints reside in the same nodes as the EE endpoints .
- As more optimal routes are made available for an existing EE connection, they will now be utilized.
  - Avoid using less optimal routes for the life of the EE connection.
  - New available IP routes for an existing EE connection can only be learned when data transmission is occurring.
  - Computing new IP routes for an EE connection will be limited to once a minute.

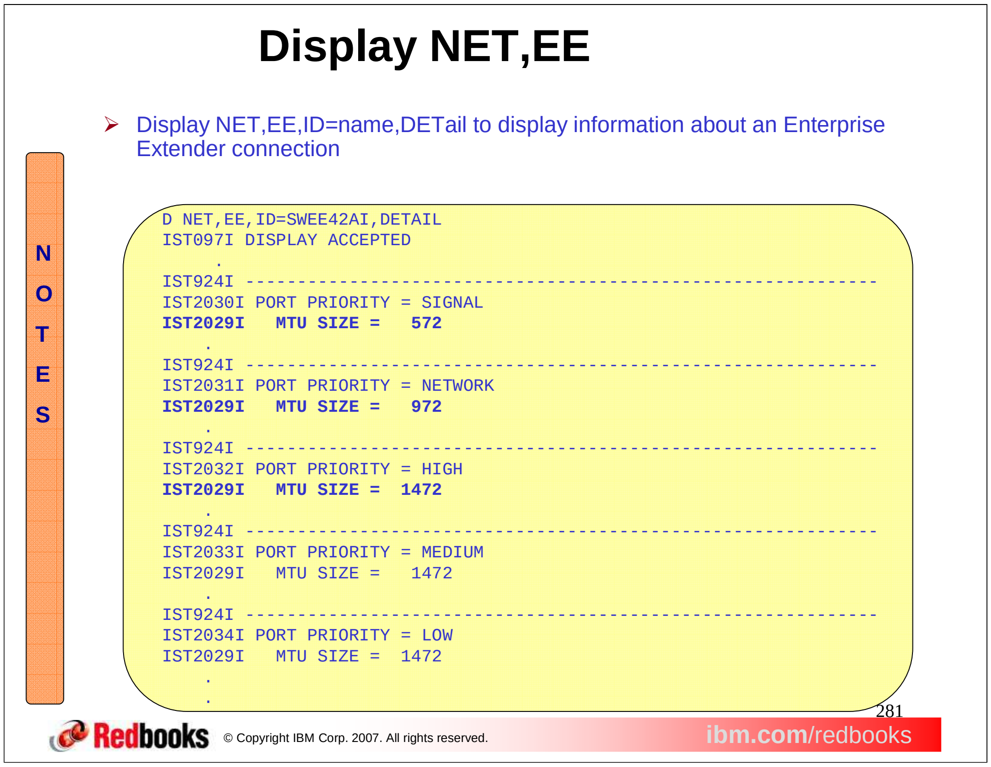
Enterprise Extender autonomics is associated with two potential performance enhancements:

1) Allow for VTAM to learn of changing MTU sizes associated with an Enterprise Extender connection. This permits the avoidance of packet fragmentation when the MTU size is decreased. And in some rare cases for VTAM to pass larger packets to TCP/IP to better utilize the current interface associated with an EE connection.

2) When new routes are learned by TCP/IP allow for the determination of a more optimal route for an existing Enterprise Extender connection.

An RTP connection routed over an Enterprise Extender (EE) connection will learn of changes in the MTU size only when their endpoints reside in the same nodes as the EE endpoints.

The RTP connection's network layer packet (NLP) size can only be increased to the maximum packet size returned by the Route_Setup signal exchanged during RTP initialization or path switch. The RTP connection’s NLP size will be negotiated to be a value no smaller than 1492 for the traversal of data over an EE connection. If the EE connection MTU size is less than 768 bytes, VTAM sets the maximum NLP packet size to 768 (this is the smallest maximum packet size allowed by VTAM for HPR packets). This limitation can cause TCP/IP to fragment but exists because the RTP layer cannot allow the HPR header to be segmented in the RTP layer.



Display NET,EE

- Display NET,EE,ID=name,DETail to display information about an Enterprise Extender connection

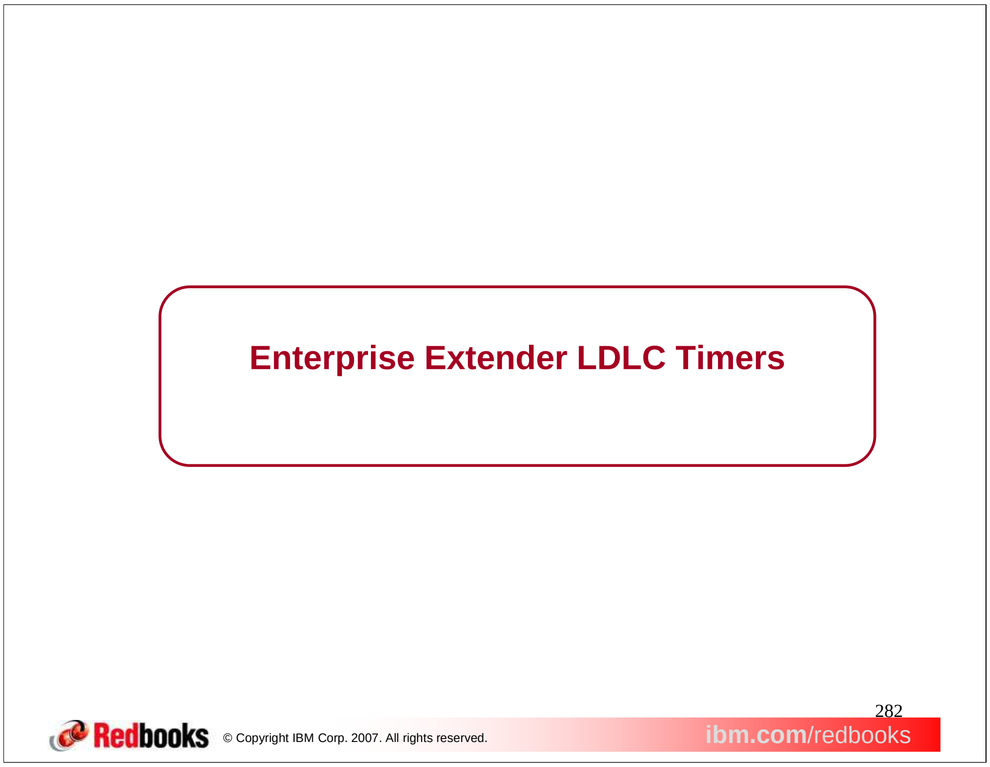
```
D NET,EE,ID=SWEE42AI,DETAIL
IST097I DISPLAY ACCEPTED
     .
IST924I -------------------------------------------------------------
IST2030I PORT PRIORITY = SIGNAL
IST2029I   MTU SIZE =   572
    .
IST924I -------------------------------------------------------------
IST2031I PORT PRIORITY = NETWORK
IST2029I   MTU SIZE =   972
    .
IST924I -------------------------------------------------------------
IST2032I PORT PRIORITY = HIGH
IST2029I   MTU SIZE =  1472
    .
IST924I -------------------------------------------------------------
IST2033I PORT PRIORITY = MEDIUM
IST2029I   MTU SIZE =   1472
    .
IST924I -------------------------------------------------------------
IST2034I PORT PRIORITY = LOW
IST2029I   MTU SIZE =  1472
    .
    .
```

N O T E S

281

Redbooks ibm.com/redbooks

When VTAM detects this condition the EE connection's MTU size during the transmission of an NLP then the MTU size will be altered (this change can be seen on the message IST2029I when you issue the DISPLAY NET,EE command). Policy-based routing is obviously being used in this example. Policy-based routing tables have been installed which directs the traffic for some of the ports over different interfaces which have different MTU sizes.

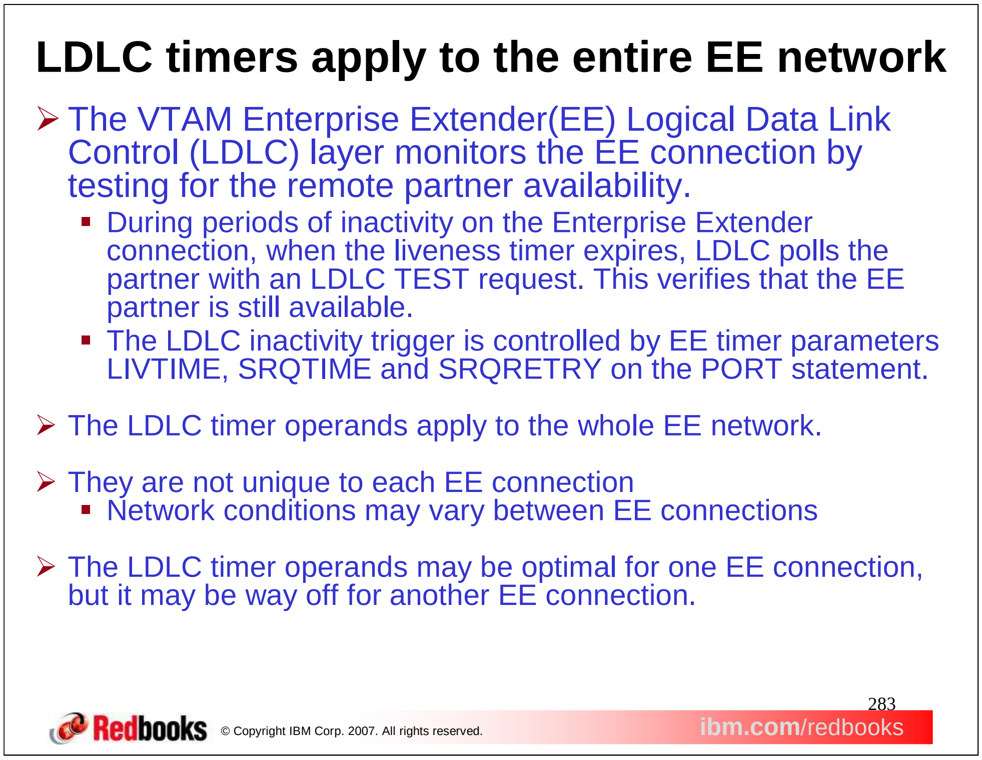


Enterprise Extender LDLC Timers

282

ibm.com/redbooks

This section describes the changes to the Enterprise Extender LDLC timers function.



LDLC timers apply to the entire EE network

- The VTAM Enterprise Extender(EE) Logical Data Link Control (LDLC) layer monitors the EE connection by testing for the remote partner availability.
  - During periods of inactivity on the Enterprise Extender connection, when the liveness timer expires, LDLC polls the partner with an LDLC TEST request. This verifies that the EE partner is still available.
  - The LDLC inactivity trigger is controlled by EE timer parameters LIVTIME, SRQTIME and SRQRETRY on the PORT statement.
- The LDLC timer operands apply to the whole EE network.
- They are not unique to each EE connection
  - Network conditions may vary between EE connections
- The LDLC timer operands may be optimal for one EE connection, but it may be way off for another EE connection.

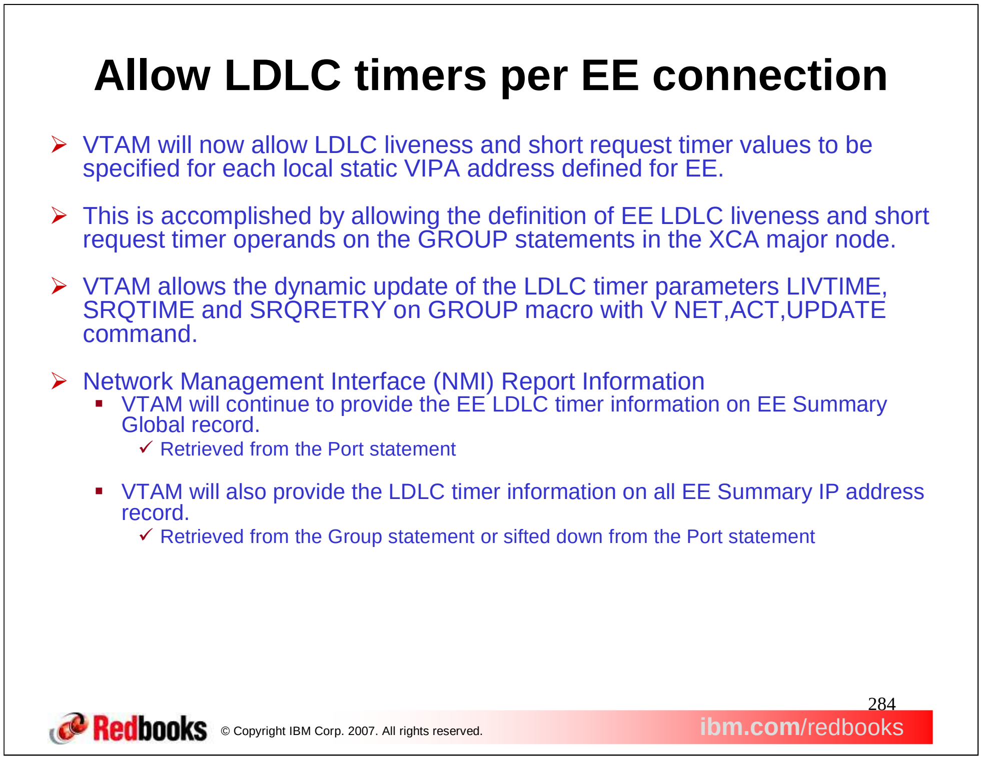
283

Redbooks

ibm.com/redbooks

LIVTIME specifies the Enterprise Extender logical data link control liveness timer interval range, in seconds. Two values can be specified on the LDLC liveness timer (LIVTIME). These values are optional with the first being the initial LIVTIME value (init_value) and the second the maximum LIVTIME value (max_value). Specifying a max_value larger than the init_value enables the EE LDLC Keep-Alive Reduction Function. This function enables the current LIVTIME window to expand and contract based on current network conditions. Expanding the current LIVTIME window reduces the number of LDLC test flows that occur during periods of inactivity. SRQTIME specifies the Enterprise Extender logical link control short request timer interval in seconds. SRQRETRY specifies the number of times the short request timer is retried before the port becomes inoperative. The LDLC layer monitors the EE connection, sending a test frame if no inbound activity is detected for the number of seconds specified by the LIVTIME operand as coded (or defaulted). If no response is received for the number of seconds specified by the SRQTIME operand, another test frame will be issued. As long as no response is received, LDLC retries SRQRETRY. If no response is received after the last retry, the EE link will be disconnected.

z/OS Communications Server currently does not provide flexibility to EE LDLC timers. The LDLC timer operands apply to the entire EE network. They are not unique to each EE connection. Since network conditions may vary between connections, the operands may be optimal for one EE connection and not optimal for another EE connection.



Allow LDLC timers per EE connection

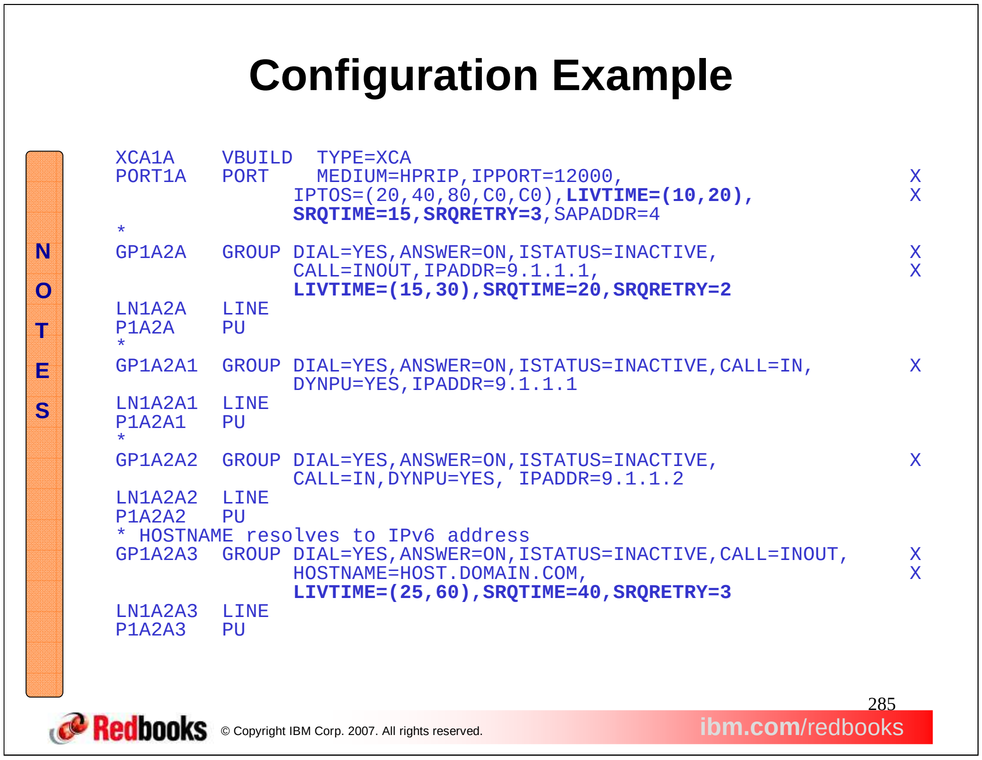
- VTAM will now allow LDLC liveness and short request timer values to be specified for each local static VIPA address defined for EE.
- This is accomplished by allowing the definition of EE LDLC liveness and short request timer operands on the GROUP statements in the XCA major node.
- VTAM allows the dynamic update of the LDLC timer parameters LIVTIME, SRQTIME and SRQRETRY on GROUP macro with V NET,ACT,UPDATE command.
- Network Management Interface (NMI) Report Information
  - VTAM will continue to provide the EE LDLC timer information on EE Summary Global record.
    - ✓ Retrieved from the Port statement
  - VTAM will also provide the LDLC timer information on all EE Summary IP address record.
    - ✓ Retrieved from the Group statement or sifted down from the Port statement

284

In z/OS V1R9, VTAM will provide the LDLC timers for each static VIPA by allowing the timer operands on the GROUP statement in an XCA major node. VTAM will allow the dynamic update of the LDLC timer parameters on GROUP statement with V NET,ACT,UPDATE command. The PORT statement defines the system wide EE LDLC timer parameters. These values are used if they are not specified on the GROUP statement. If the GROUP statement has one or more EE timer parameters, they will override the EE timer parameters of the PORT statement for this GROUP. If the GROUP statement has only one LDLC timer parameter specified, VTAM will sift down the other two LDLC timer parameters from PORT statement.

If two or more GROUPs are using the same static VIPA and they have the different EE timer parameters, VTAM will use the EE timer parameters of the first activated GROUP for that VIPA. Other GROUPs will use the EE timer parameters specified on the first activated GROUP.

VTAM provides EE LDLC timer parameters LIVTIME, SRQTIME and SRQRETRY information on EE Summary Global record. VTAM will also provide EE LDLC timer parameters LIVTIME, SRQTIME and SRQRETRY information on EE Summary IP address record. The EE Summary IP address record contains the following fields: EESumIP_Timer_LIVTIME, EESumIP_Timer_SRQTIME, and EESumIP_Timer_SRQRETRY.



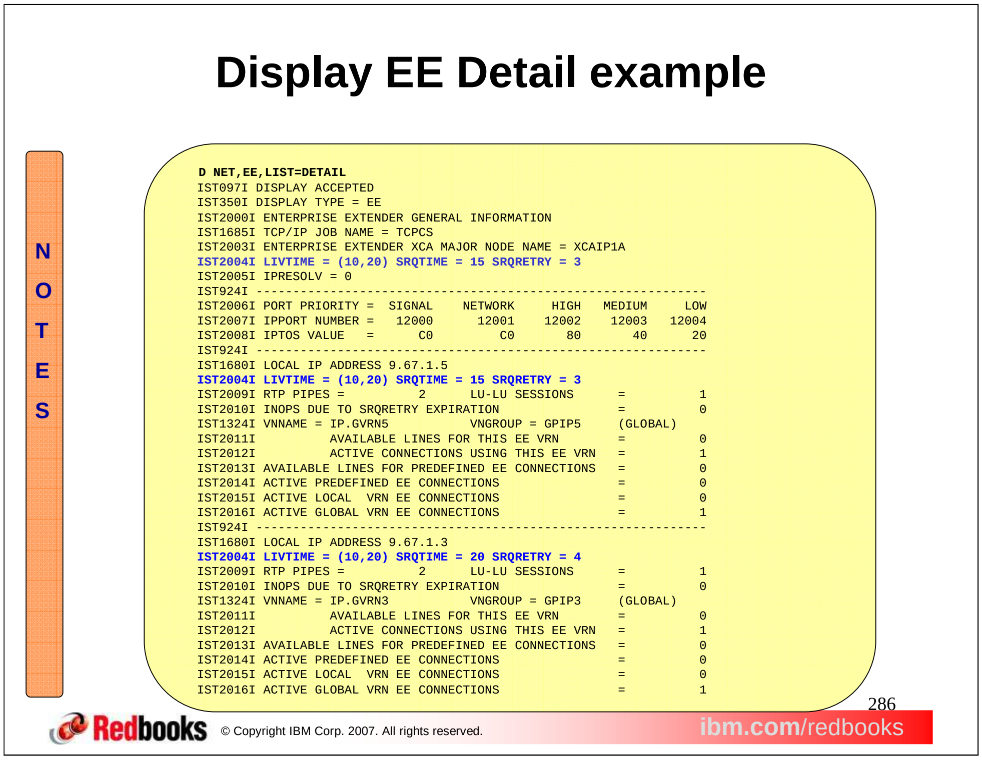
Configuration Example

```
XCA1A    VBUILD  TYPE=XCA
PORT1A   PORT    MEDIUM=HPRIP,IPPORT=12000,                        X
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,20),             X
               SRQTIME=15,SRQRETRY=3,SAPADDR=4
*
GP1A2A   GROUP DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,                X
               CALL=INOUT,IPADDR=9.1.1.1,                          X
               LIVTIME=(15,30),SRQTIME=20,SRQRETRY=2
LN1A2A   LINE
P1A2A    PU
*
GP1A2A1  GROUP DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,CALL=IN,        X
               DYNPU=YES,IPADDR=9.1.1.1
LN1A2A1  LINE
P1A2A1   PU
*
GP1A2A2  GROUP DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,                X
               CALL=IN,DYNPU=YES, IPADDR=9.1.1.2
LN1A2A2  LINE
P1A2A2   PU
* HOSTNAME resolves to IPv6 address
GP1A2A3  GROUP DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,CALL=INOUT,     X
               HOSTNAME=HOST.DOMAIN.COM,                           X
               LIVTIME=(25,60),SRQTIME=40,SRQRETRY=3
LN1A2A3  LINE
P1A2A3   PU
```

Redbooks

This is the example of sample XCA Major node definition. Two GROUPs, GP1A2A and GP1A2A1, have the same static VIPA IP address. Group GP1A2A has LDLC timer parameters specified and they are different than the LDLC timer parameters specified on PORT definition statement. Group GP1A2A1 does not have LDLC timer parameters specified so it will inherit the LDLC timer parameters value from the PORT definition statement (sift down effect). VTAM uses the LDLC timer parameters of the first activated group, GP1A2A or GP1A2A1, for this static VIPA. When the second GROUP is activated, it will receive the error message indicating that it is using the LDLC timer parameters of the first activated GROUP.

Group GP1A2A2 does not have LDLC timer parameters specified so it will inherit the LDLC timer parameters value from the PORT definition statement (sift down effect). Group GP1A2A3 has LDLC timer parameters specified and they are different than the LDLC timer parameters specified on PORT definition statement. Both GROUPs have different static VIPA address. So, they will use their own LDLC timer parameters.



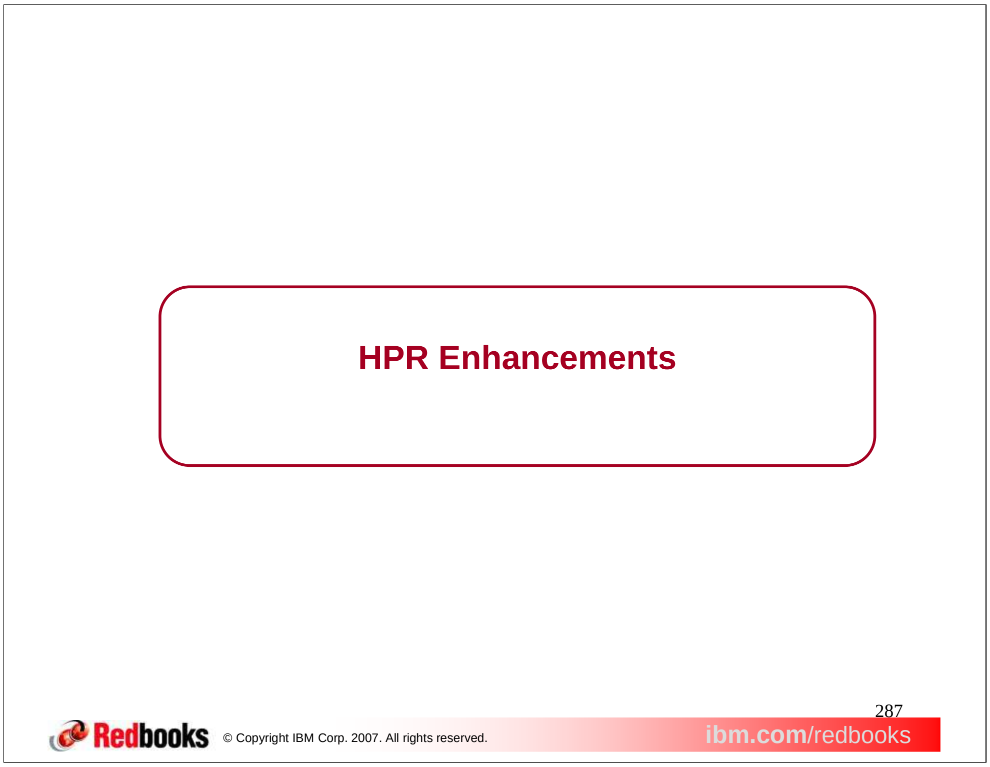
Display EE Detail example

```
D NET,EE,LIST=DETAIL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2000I ENTERPRISE EXTENDER GENERAL INFORMATION
IST1685I TCP/IP JOB NAME = TCPCS
IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = XCAIP1A
IST2004I LIVTIME = (10,20) SRQTIME = 15 SRQRETRY = 3
IST2005I IPRESOLV = 0
IST924I -------------------------------------------------------------
IST2006I PORT PRIORITY =  SIGNAL    NETWORK     HIGH   MEDIUM      LOW
IST2007I IPPORT NUMBER =   12000      12001    12002    12003    12004
IST2008I IPTOS VALUE   =      C0         C0       80       40       20
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 9.67.1.5
IST2004I LIVTIME = (10,20) SRQTIME = 15 SRQRETRY = 3
IST2009I RTP PIPES =          2      LU-LU SESSIONS      =          1
IST2010I INOPS DUE TO SRQRETRY EXPIRATION                =          0
IST1324I VNNAME = IP.GVRN5           VNGROUP = GPIP5     (GLOBAL)
IST2011I          AVAILABLE LINES FOR THIS EE VRN        =          0
IST2012I          ACTIVE CONNECTIONS USING THIS EE VRN   =          1
IST2013I AVAILABLE LINES FOR PREDEFINED EE CONNECTIONS   =          0
IST2014I ACTIVE PREDEFINED EE CONNECTIONS                =          0
IST2015I ACTIVE LOCAL  VRN EE CONNECTIONS                =          0
IST2016I ACTIVE GLOBAL VRN EE CONNECTIONS                =          1
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 9.67.1.3
IST2004I LIVTIME = (10,20) SRQTIME = 20 SRQRETRY = 4
IST2009I RTP PIPES =          2      LU-LU SESSIONS      =          1
IST2010I INOPS DUE TO SRQRETRY EXPIRATION                =          0
IST1324I VNNAME = IP.GVRN3           VNGROUP = GPIP3     (GLOBAL)
IST2011I          AVAILABLE LINES FOR THIS EE VRN        =          0
IST2012I          ACTIVE CONNECTIONS USING THIS EE VRN   =          1
IST2013I AVAILABLE LINES FOR PREDEFINED EE CONNECTIONS   =          0
IST2014I ACTIVE PREDEFINED EE CONNECTIONS                =          0
IST2015I ACTIVE LOCAL  VRN EE CONNECTIONS                =          0
IST2016I ACTIVE GLOBAL VRN EE CONNECTIONS                =          1
```

An excerpt of a Display EE Detail output is shown on this slide. The first IST2004I message shows the LDLC timer values from the PORT definition statement. Subsequent IST2004I messages show the LDLC timers definitions coded on GROUP definition statement or sifted down from PORT definition statement.

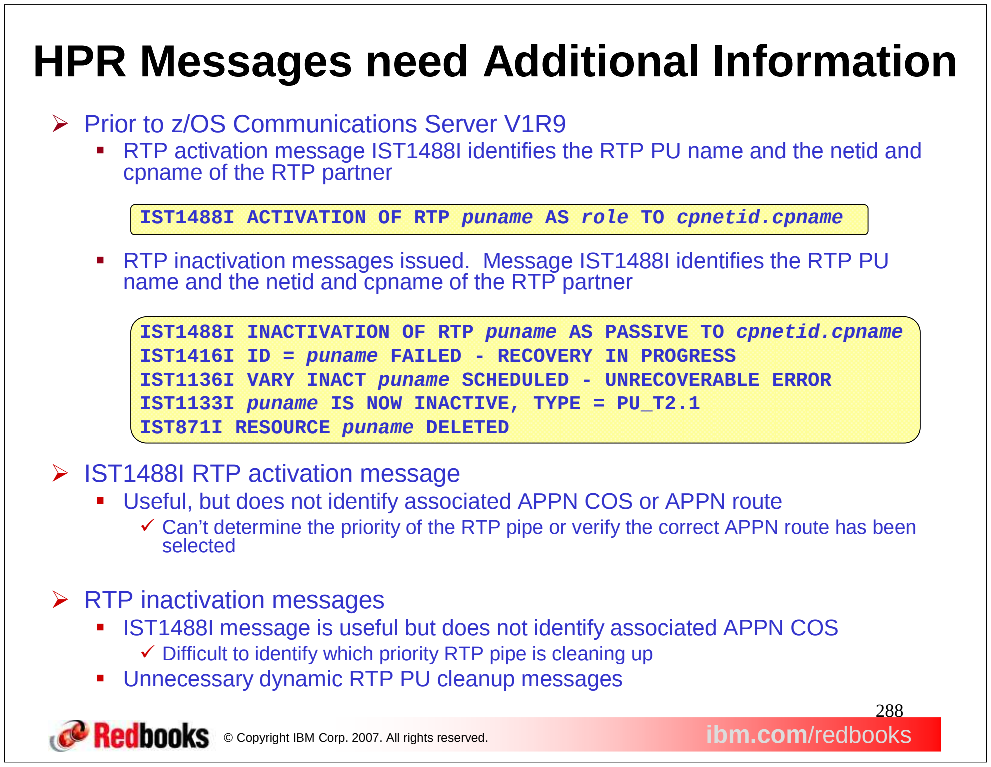
The D NET,EE,HOSTNAME= and the D NET,EEDIAG,HOSTNAME= reports also contain message IST2004I which shows the LDLC timer parameters defined on GROUP statement or sifted down from PORT definition statement.

The D NET,EE,LIST=SUMMARY report contains message IST2004I which shows the LDLC timers definitions coded or defaulted on PORT definition statement.



HPR Enhancements

This section describes the HPR enhancements made in the Communications Server for V1R9.



HPR Messages need Additional Information

- Prior to z/OS Communications Server V1R9
  - RTP activation message IST1488I identifies the RTP PU name and the netid and cpname of the RTP partner

```
IST1488I ACTIVATION OF RTP puname AS role TO cpnetid.cpname
```

  - RTP inactivation messages issued. Message IST1488I identifies the RTP PU name and the netid and cpname of the RTP partner

```
IST1488I INACTIVATION OF RTP puname AS PASSIVE TO cpnetid.cpname
IST1416I ID = puname FAILED - RECOVERY IN PROGRESS
IST1136I VARY INACT puname SCHEDULED - UNRECOVERABLE ERROR
IST1133I puname IS NOW INACTIVE, TYPE = PU_T2.1
IST871I RESOURCE puname DELETED
```

- IST1488I RTP activation message
  - Useful, but does not identify associated APPN COS or APPN route
    - ✓ Can't determine the priority of the RTP pipe or verify the correct APPN route has been selected
- RTP inactivation messages
  - IST1488I message is useful but does not identify associated APPN COS
    - ✓ Difficult to identify which priority RTP pipe is cleaning up
  - Unnecessary dynamic RTP PU cleanup messages

288

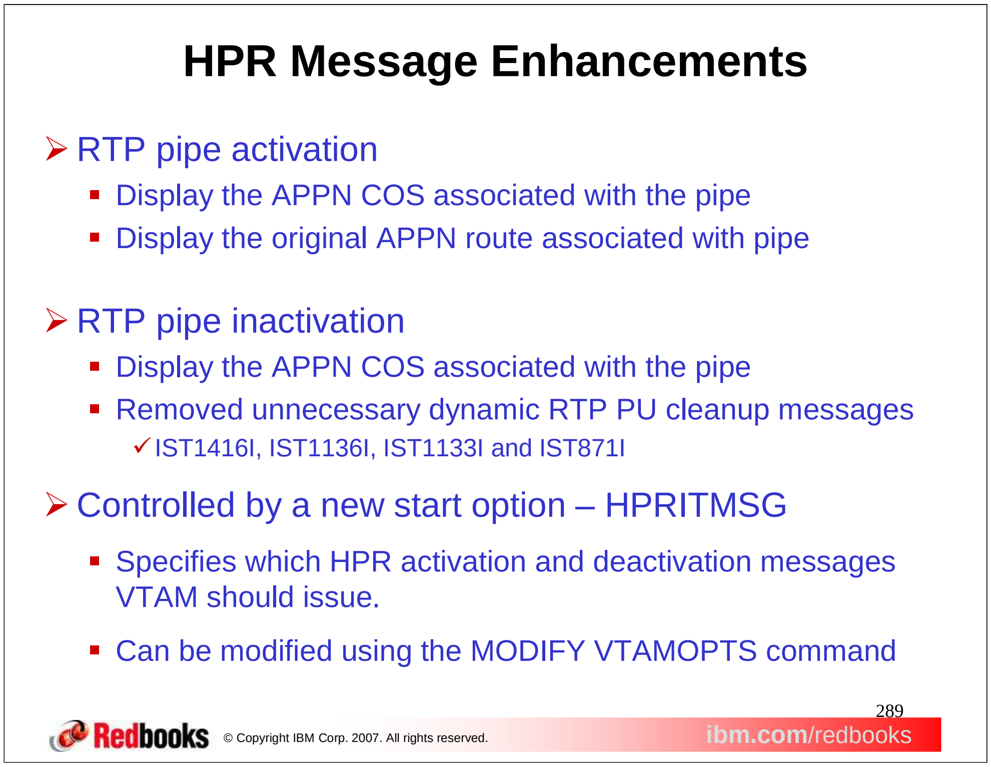
Redbooks

ibm.com/redbooks

In previous releases, when an RTP pipe is activated a single message, IST1488I, is issued by VTAM to identify the RTP PU name and the associated NETID and CPNAME of the RTP partner. When an RTP pipe is inactivated, VTAM also issued message IST1488I to identify the RTP PU name and the associated NETID and CPNAME of the RTP partner. In addition to the IST1488I message, VTAM issued a few other dynamic PU cleanup messages including IST1416I, IST1136I, ISt1133I ad IST871I.

During RTP pipe activation, VTAM does not identify the APPN COS or APPN route associated with the RTP pipe. Without this information, you cannot identify the priority of the RTP pipe or verify the correct APPN route has been selected.

During RTP pipe inactivation, VTAM does not identify the associated APPN COS. Without this information, it is difficult to identify which priority RTP pipe is cleaning up. Also, many of the dynamic RTP PU cleanup messages are unnecessarily issued. If a large number of RTP pipes are cleaning up, this may lead to hundreds or thousands of unnecessary messages being issued to the system console.



HPR Message Enhancements

- RTP pipe activation
  - Display the APPN COS associated with the pipe
  - Display the original APPN route associated with pipe
- RTP pipe inactivation
  - Display the APPN COS associated with the pipe
  - Removed unnecessary dynamic RTP PU cleanup messages
    - ✓ IST1416I, IST1136I, IST1133I and IST871I
- Controlled by a new start option – HPRITMSG
  - Specifies which HPR activation and deactivation messages VTAM should issue.
  - Can be modified using the MODIFY VTAMOPTS command

289

Redbooks ibm.com/redbooks

During RTP pipe activation, VTAM may now display the APPN COS or APPN route associated with the RTP pipe. With this new information, you can easily identify the priority of the RTP pipe which is activating. You can also verify the correct APPN route has been selected.

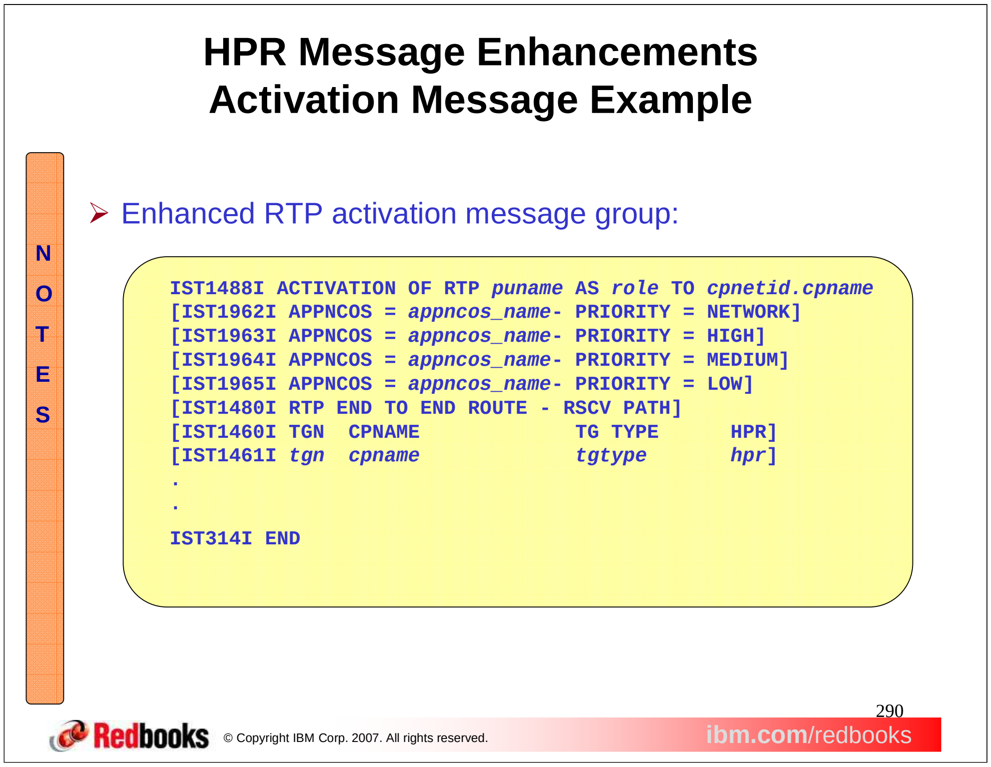
During RTP pipe inactivation, VTAM may now display the associated APPN COS. This new information makes it easier for the operator to understand which priority traffic is ending. Also, the dynamic RTP PU cleanup messages are no longer issued. Removing these unnecessary messages helps cleanup the system log so the system operator can focus on more important messages.

The new HPR inactivation and deactivation message enhancements are controlled by the new start option HPRITMSG. This start option may be specified as BASE or ENHANCED. The default is set to BASE which means VTAM will issue the base RTP pipe activation and inactivation messages. When this start option is set to ENHANCED, VTAM will issue the enhanced versions of the RTP pipe activation and inactivation messages.

Since some of the messages affected by this start option have different versions, the exact message affected in the output depends on the value specified on the MSGLEVEL start option.

If the current value of the HPRITMSG is not appropriate for your system, you can modify the HPRITMSG value by using the MODIFY VTAMOPTS command.

The HPRITMSG start option is only valid if VTAM provides RTP level HPR support.

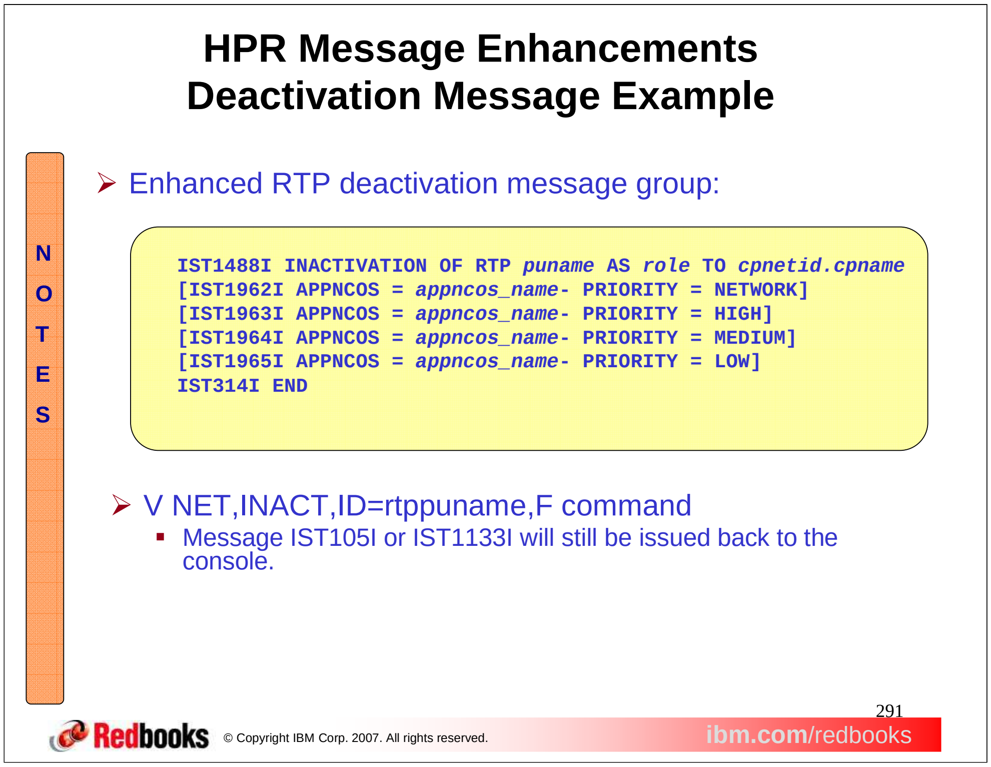


HPR Message Enhancements Activation Message Example

- Enhanced RTP activation message group:

```
IST1488I ACTIVATION OF RTP puname AS role TO cpnetid.cpname
[IST1962I APPNCOS = appncos_name- PRIORITY = NETWORK]
[IST1963I APPNCOS = appncos_name- PRIORITY = HIGH]
[IST1964I APPNCOS = appncos_name- PRIORITY = MEDIUM]
[IST1965I APPNCOS = appncos_name- PRIORITY = LOW]
[IST1480I RTP END TO END ROUTE - RSCV PATH]
[IST1460I TGN  CPNAME             TG TYPE      HPR]
[IST1461I tgn  cpname             tgtype       hpr]
.
.

IST314I END
```

When the HPRITMSG is set to ENHANCED, VTAM will now display the APPN COS and APPN route when an RTP pipe is activated.



HPR Message Enhancements
Deactivation Message Example

N O T E S

- Enhanced RTP deactivation message group:

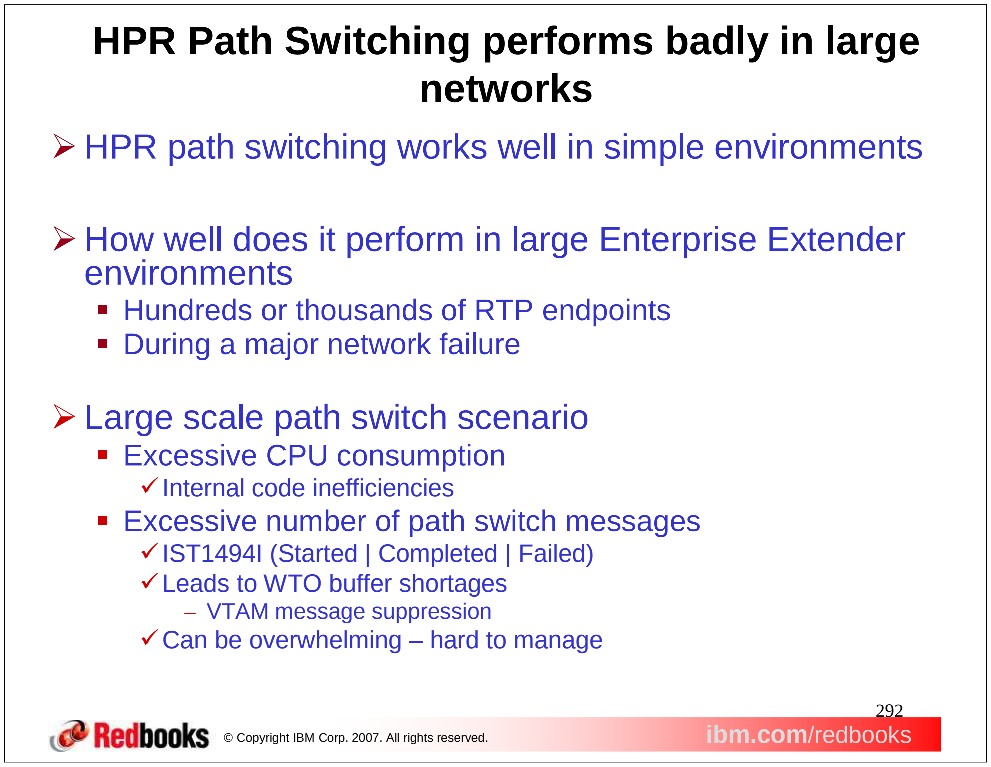
```
IST1488I INACTIVATION OF RTP puname AS role TO cpnetid.cpname
[IST1962I APPNCOS = appncos_name- PRIORITY = NETWORK]
[IST1963I APPNCOS = appncos_name- PRIORITY = HIGH]
[IST1964I APPNCOS = appncos_name- PRIORITY = MEDIUM]
[IST1965I APPNCOS = appncos_name- PRIORITY = LOW]
IST314I END
```

- V NET,INACT,ID=rtppuname,F command
  - Message IST105I or IST1133I will still be issued back to the console.

291

Redbooks ibm.com/redbooks

When the HPRITMSG is set to ENHANCED, VTAM will now display the APPN COS when an RTP pipe is inactivated. VTAM also no longer issues the dynamic PU cleanup messages when an RTP pipe is inactivated. One exception is when a vary inactivate command is issued against an RTP pipe. In this case, message IST105I or IST1133I is still issued to the console so the operator receives a response to the vary inactivate command.



HPR Path Switching performs badly in large networks

- ➢ HPR path switching works well in simple environments
- ➢ How well does it perform in large Enterprise Extender environments
  - Hundreds or thousands of RTP endpoints
  - During a major network failure
- ➢ Large scale path switch scenario
  - Excessive CPU consumption
    - ✓Internal code inefficiencies
  - Excessive number of path switch messages
    - ✓IST1494I (Started | Completed | Failed)
    - ✓Leads to WTO buffer shortages
      - – VTAM message suppression
    - ✓Can be overwhelming – hard to manage

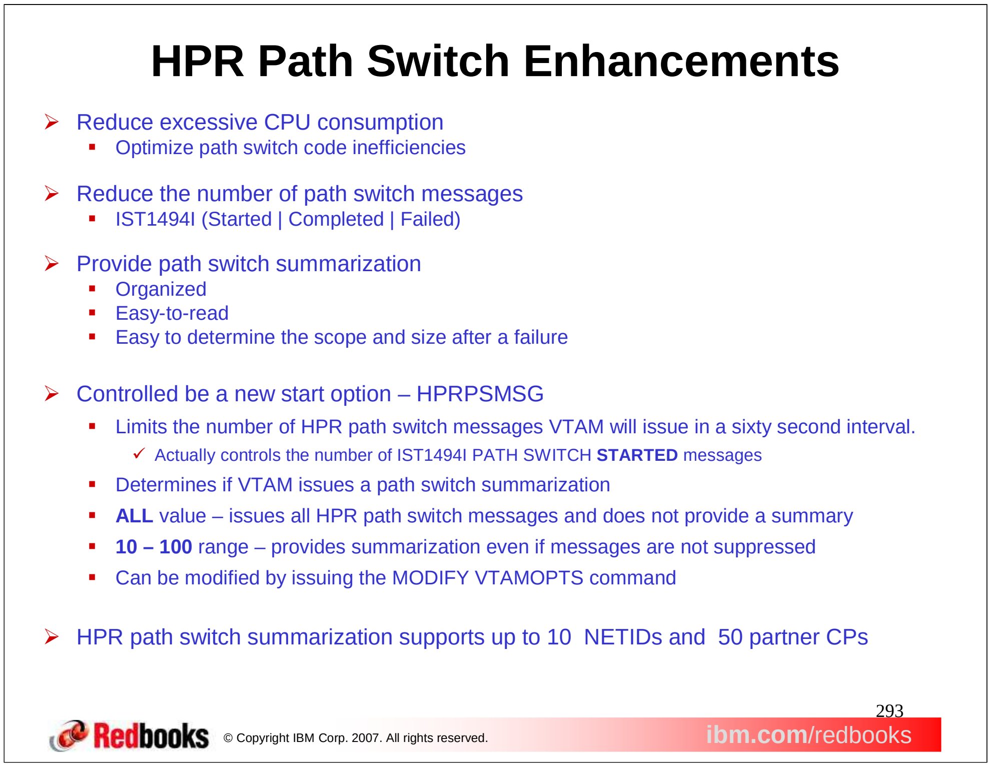
292

Redbooks

ibm.com/redbooks

HPR path switching works well in simple environments, but there have been concerns raised on how well it performs during a large network failure.

During a large scale path switch scenario, VTAM consumes too much CPU and issues too many path switch messages. In addition, the HPR path switch messages can be overwhelming and hard to manage.



HPR Path Switch Enhancements

- Reduce excessive CPU consumption
  - Optimize path switch code inefficiencies
- Reduce the number of path switch messages
  - IST1494I (Started | Completed | Failed)
- Provide path switch summarization
  - Organized
  - Easy-to-read
  - Easy to determine the scope and size after a failure
- Controlled be a new start option – HPRPSMSG
  - Limits the number of HPR path switch messages VTAM will issue in a sixty second interval.
    - ✓ Actually controls the number of IST1494I PATH SWITCH **STARTED** messages
  - Determines if VTAM issues a path switch summarization
  - **ALL** value – issues all HPR path switch messages and does not provide a summary
  - **10 – 100** range – provides summarization even if messages are not suppressed
  - Can be modified by issuing the MODIFY VTAMOPTS command
- HPR path switch summarization supports up to 10 NETIDs and 50 partner CPs

Changes have been made to the internal HPR path switch code to optimize path switch code to reduce CPU usage. Changes have also been made to reduce the number of path switch messages issued to the console. In addition, VTAM will also output a path switch summarization display to document all the associated path switch events which occurred during a given time interval.

The HPR path switch enhancements are controlled by a new start option, HPRPSMSG. The default of ALL issues all HPR path switch message and does not provide a summary display of the path switch events. Specifying a value between 10 and 100 allows VTAM to limit the number of HPR path switch messages issued to the console in a sixty second interval. If a STARTED message is issued for a pipe, the COMPLETED or FAILED message is always issued. The Sixty second interval starts when a path switch event occurs.

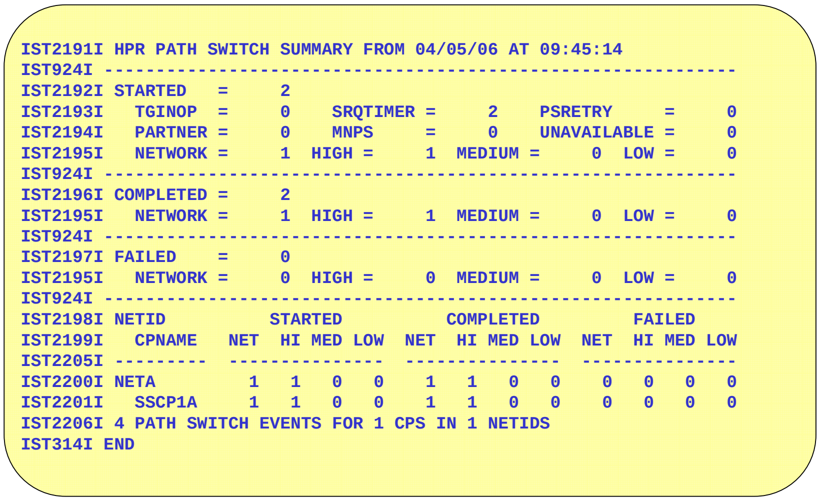
The path switch message summary is always issued to the console at the end of the time interval, whether or not messages were suppressed. The summary will include all path switch events which occurred during the given path switch event time interval, including path switch event information that was issued to the console.

The HPRPSMSG start option is only valid when VTAM provides RTP level HPR support. If the current value of the HPRPSMSG is not appropriate for your system, you can modify the HPRPSMSG value by using the MODIFY VTAMOPTS command.

HPR path switch summarization supports 10 NETIDs and 50 partner CPs. If a large outage exceeds these limits, the report will not contain all NETID and CP specific counts. Message IST2206I will show greater than 10 NETIDs if there was a NETID overflow and greater than 50 CPs if there was a CP overflow. The path switch started (IST2192I), completed (IST2196I), and failed (IST2197I) message counts are always accurate. Even if overflow occurs.) The NETID and CP specific counts are limited to 999. IST2200I and IST2201I will display 999 when count is 999 or higher. You may have multiple path switch summaries per outage because HPRPST may have different values for each pipe priorities.

HPR Path Switch Enhancements Summarization Example 1

N O T E S



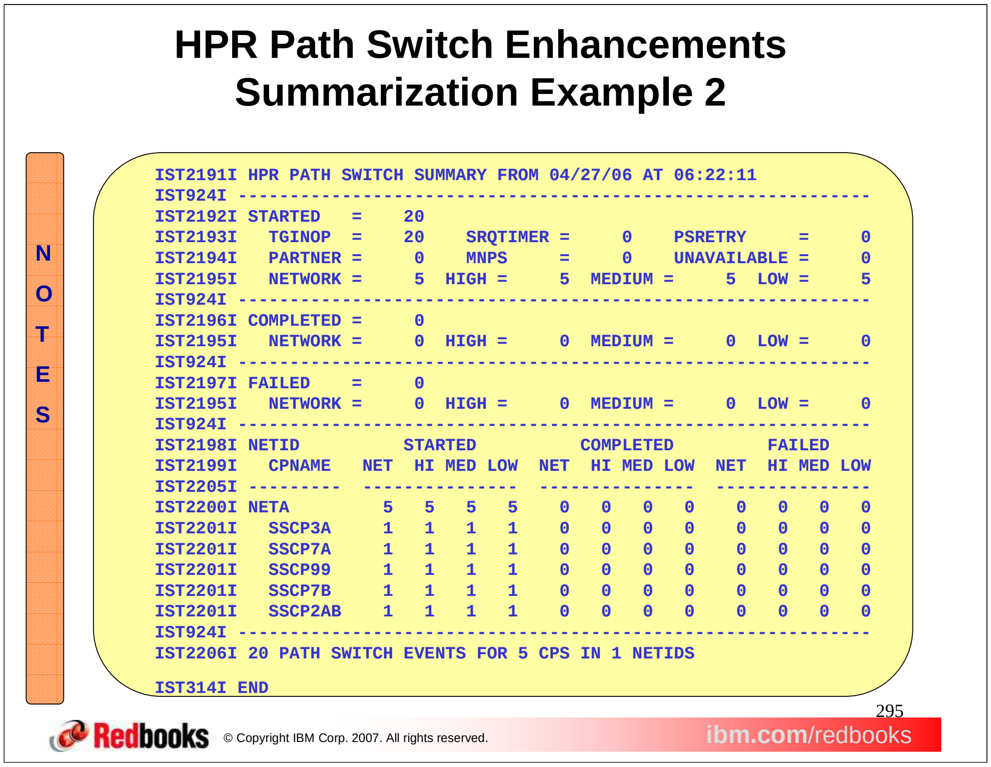
```
IST2191I HPR PATH SWITCH SUMMARY FROM 04/05/06 AT 09:45:14
IST924I -------------------------------------------------------------
IST2192I STARTED   =     2
IST2193I   TGINOP  =     0    SRQTIMER =     2    PSRETRY     =     0
IST2194I   PARTNER =     0    MNPS     =     0    UNAVAILABLE =     0
IST2195I   NETWORK =     1  HIGH =     1  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2196I COMPLETED =     2
IST2195I   NETWORK =     1  HIGH =     1  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2197I FAILED    =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2198I NETID          STARTED          COMPLETED        FAILED
IST2199I   CPNAME   NET  HI MED LOW  NET  HI MED LOW  NET  HI MED LOW
IST2205I ---------  ---------------  ---------------  ---------------
IST2200I NETA         1   1   0   0    1   1   0   0    0   0   0   0
IST2201I   SSCP1A     1   1   0   0    1   1   0   0    0   0   0   0
IST2206I 4 PATH SWITCH EVENTS FOR 1 CPS IN 1 NETIDS
IST314I END
```

294

ibm.com/redbooks

This is an example of an HPR path switch summarization display. The output is organized into four basic sections. The first section display the number of RTP pipes which entered path switch during this interval. This section also displays counts by path switch reason and by the associated pipe priority. The second section display the number of RTP pipes which successfully completed path switch. This section also display the path switch counts by the associated pipe priority. The third section displays the number of RTP pipes which unsuccessfully path switched. This section also displays the path switch counts by the associated pipe priority. The fourth section organizes the path switch information by NETID and CPNAME.

The path switch summarization shows all the path switch activity from 09:45:14 until the current time (approximately 09:46:14). During this interval, 2 RTP pipes path switched due to timeouts (Short Request Timer expiration). These 2 pipes consisted of 1 network priority pipe (CP-CP) and 1 high priority pipe. During this interval, the same two RTP pipes successfully completed path switch. During this interval, no RTP pipes failed path switch. The display shows a breakdown of path switch events by each CP within each NETID. In this example, the output clearly identifies the problem is isolated to the connectivity specific to this one partner node.



HPR Path Switch Enhancements Summarization Example 2

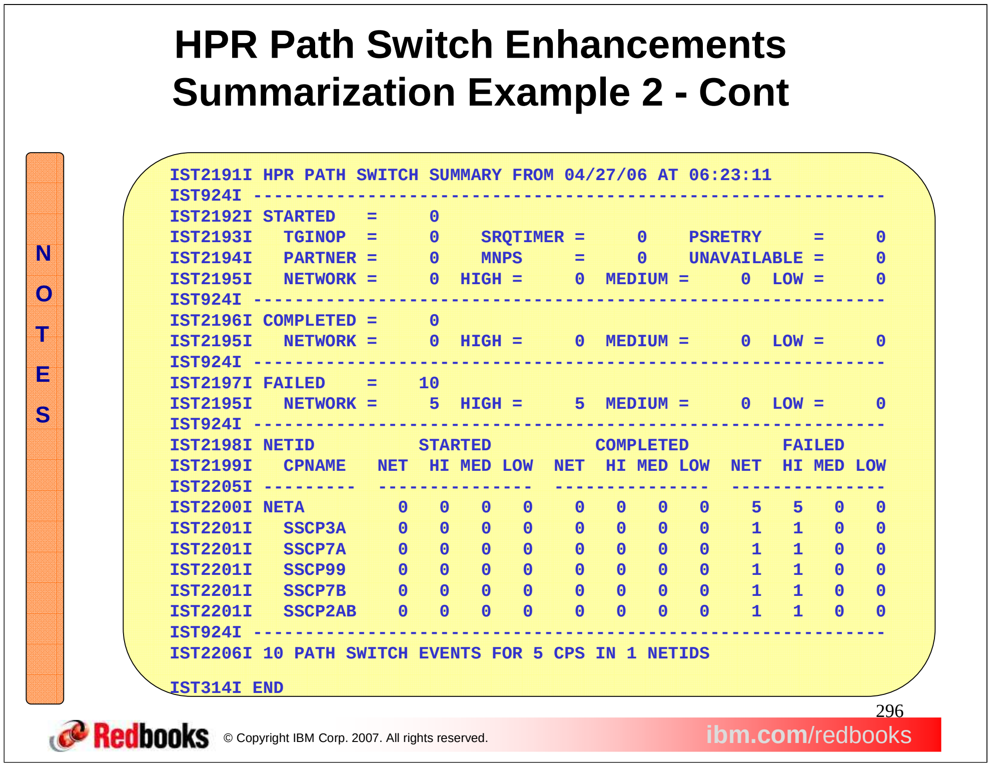
N O T E S

```
IST2191I HPR PATH SWITCH SUMMARY FROM 04/27/06 AT 06:22:11
IST924I -------------------------------------------------------------
IST2192I STARTED   =    20
IST2193I   TGINOP  =    20    SRQTIMER =     0    PSRETRY     =     0
IST2194I   PARTNER =     0    MNPS     =     0    UNAVAILABLE =     0
IST2195I   NETWORK =     5  HIGH =     5  MEDIUM =     5  LOW =     5
IST924I -------------------------------------------------------------
IST2196I COMPLETED =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2197I FAILED    =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2198I NETID          STARTED          COMPLETED         FAILED
IST2199I   CPNAME   NET  HI MED LOW  NET  HI MED LOW  NET  HI MED LOW
IST2205I ---------  ---------------  ---------------  ---------------
IST2200I NETA         5   5   5   5    0   0   0   0    0   0   0   0
IST2201I   SSCP3A     1   1   1   1    0   0   0   0    0   0   0   0
IST2201I   SSCP7A     1   1   1   1    0   0   0   0    0   0   0   0
IST2201I   SSCP99     1   1   1   1    0   0   0   0    0   0   0   0
IST2201I   SSCP7B     1   1   1   1    0   0   0   0    0   0   0   0
IST2201I   SSCP2AB    1   1   1   1    0   0   0   0    0   0   0   0
IST924I -------------------------------------------------------------
IST2206I 20 PATH SWITCH EVENTS FOR 5 CPS IN 1 NETIDS

IST314I END
```

295

Redbooks ibm.com/redbooks

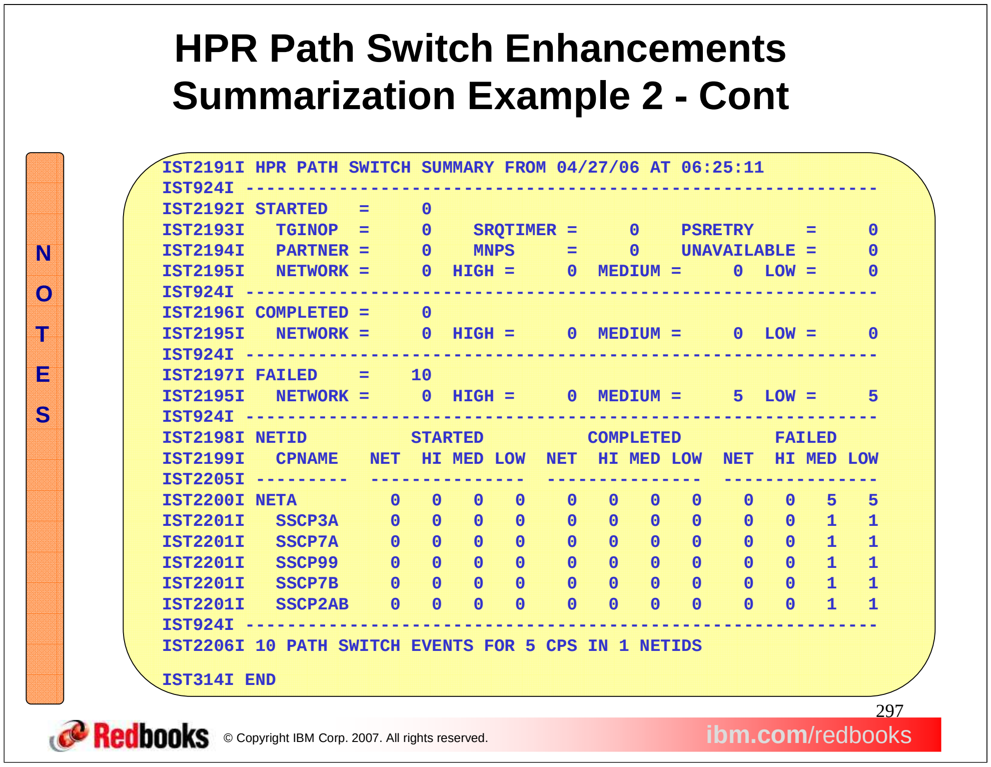
In some cases, when path switches do not complete quickly, the path switch information associated with a given RTP pipe may be spread across multiple path switch summary message groups. For example, this may occur when HPRPSMSG=10 and HPRPST =(4M,4M,2M,2M) are specified, and twenty RTP pipes (5 low_priority, 5 medium_priority, 5 high_priority, 5 network_priority) enter path switch state. The path switch message event time interval is set when the first pipe enters path switch state. The first ten RTP pipes will have IST1494I (Started) message groups issued to the console. The next ten RTP pipes will not have IST1494I(Started) message groups issued to the console. In this example, there is not an alternate route available. The RTP pipes will stay in path switch state until their respective HPRPST timers expire, at which point the path switches will fail. Since the HPRPST values for the various priorities are specified as different values, the path switches will fail at different times. When the current path switch message event interval ends, a summary of twenty path switch started events is issued to the console.



HPR Path Switch Enhancements Summarization Example 2 - Cont

```
IST2191I HPR PATH SWITCH SUMMARY FROM 04/27/06 AT 06:23:11
IST924I -------------------------------------------------------------
IST2192I STARTED   =     0
IST2193I   TGINOP  =     0    SRQTIMER =     0    PSRETRY     =     0
IST2194I   PARTNER =     0    MNPS     =     0    UNAVAILABLE =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2196I COMPLETED =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2197I FAILED    =    10
IST2195I   NETWORK =     5  HIGH =     5  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2198I NETID          STARTED          COMPLETED         FAILED
IST2199I   CPNAME   NET  HI MED LOW  NET  HI MED LOW  NET  HI MED LOW
IST2205I ---------  ---------------  ---------------  ---------------
IST2200I NETA         0   0   0   0    0   0   0   0    5   5   0   0
IST2201I   SSCP3A     0   0   0   0    0   0   0   0    1   1   0   0
IST2201I   SSCP7A     0   0   0   0    0   0   0   0    1   1   0   0
IST2201I   SSCP99     0   0   0   0    0   0   0   0    1   1   0   0
IST2201I   SSCP7B     0   0   0   0    0   0   0   0    1   1   0   0
IST2201I   SSCP2AB    0   0   0   0    0   0   0   0    1   1   0   0
IST924I -------------------------------------------------------------
IST2206I 10 PATH SWITCH EVENTS FOR 5 CPS IN 1 NETIDS

IST314I END
```

At the end of two minutes, the five network and five high priority RTP pipes will fail to path switch successfully. At this point, a new sixty second path switch interval is started. Again, the IST1494I(Failed) messages will only be issued for the RTP pipes which had IST1494I(Started) messages issued earlier. When this interval expires, the IST2191I path switch summary message group is issued.



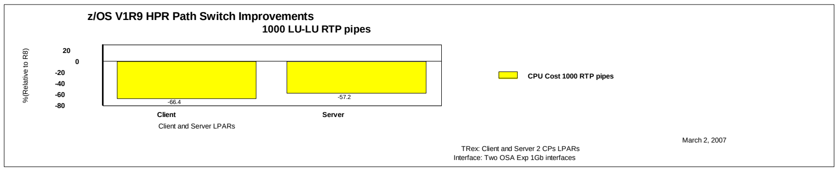
HPR Path Switch Enhancements Summarization Example 2 - Cont

```
IST2191I HPR PATH SWITCH SUMMARY FROM 04/27/06 AT 06:25:11
IST924I -------------------------------------------------------------
IST2192I STARTED   =     0
IST2193I   TGINOP  =     0    SRQTIMER =     0    PSRETRY     =     0
IST2194I   PARTNER =     0    MNPS     =     0    UNAVAILABLE =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2196I COMPLETED =     0
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     0  LOW =     0
IST924I -------------------------------------------------------------
IST2197I FAILED    =    10
IST2195I   NETWORK =     0  HIGH =     0  MEDIUM =     5  LOW =     5
IST924I -------------------------------------------------------------
IST2198I NETID          STARTED          COMPLETED         FAILED
IST2199I   CPNAME   NET  HI MED LOW  NET  HI MED LOW  NET  HI MED LOW
IST2205I ---------  ---------------  ---------------  ---------------
IST2200I NETA         0   0   0   0    0   0   0   0    0   0   5   5
IST2201I   SSCP3A     0   0   0   0    0   0   0   0    0   0   1   1
IST2201I   SSCP7A     0   0   0   0    0   0   0   0    0   0   1   1
IST2201I   SSCP99     0   0   0   0    0   0   0   0    0   0   1   1
IST2201I   SSCP7B     0   0   0   0    0   0   0   0    0   0   1   1
IST2201I   SSCP2AB    0   0   0   0    0   0   0   0    0   0   1   1
IST924I -------------------------------------------------------------
IST2206I 10 PATH SWITCH EVENTS FOR 5 CPS IN 1 NETIDS

IST314I END
```

The same occurs at the four minute mark with the final ten RTP pipes fail to path switch successfully.

HPR Path Switch Enhancements Performance Measurements

➢ How Well does it Perform?



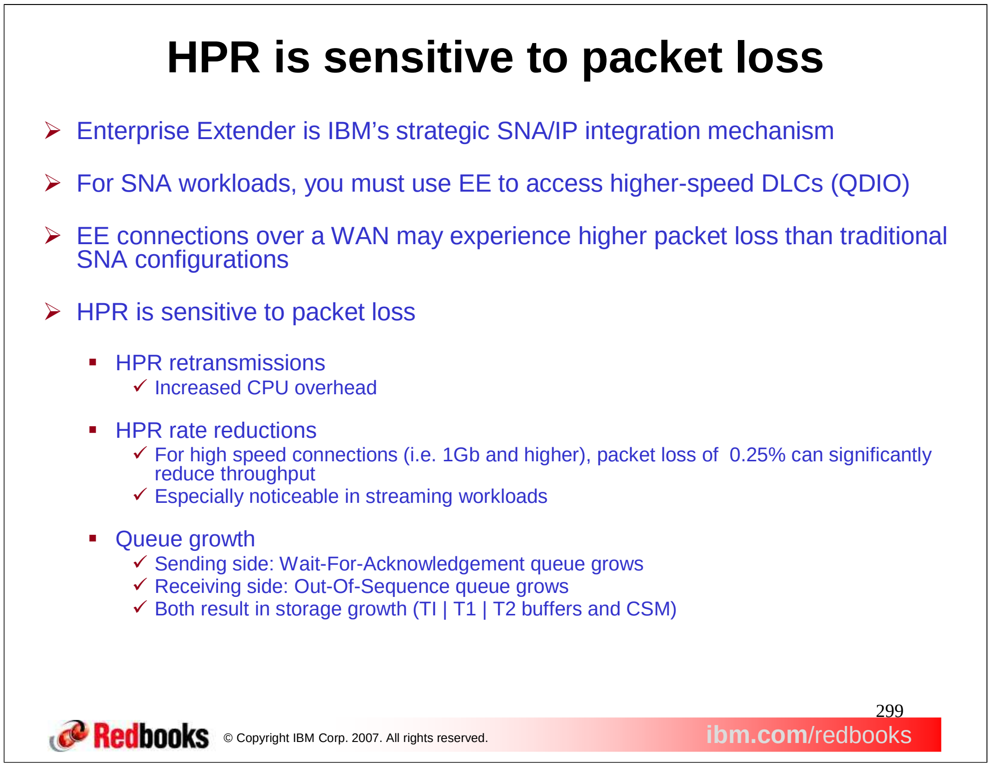
- V1R8 versus V1R9 path switch performance comparison
  - ✓ Network Node – Network Node configuration over one-hop EE (1Gb OSA)
    - Two EE TGs defined between the network nodes
  - ✓ 1000 LU-LU RTP pipes path switched from one TG to the other
  - ✓ V1R8
    - Message IST1494I added to the VTAM message-flooding prevention table
  - ✓ V1R9
    - Message IST1494I removed from the VTAM message-flooding prevention table
    - HPRPSMSG = 10
  - ✓ 57% - 66% CPU savings for this scenario

➢ If you use the VTAM message-flooding prevention table

- IST1494I should be removed before enabling the HPR path switch message reduction and summarization function
- If not, this will affect the number of path switch messages you receive on the system console

Performance measurements were taken for this line item to verify the changes made in V1R9 are beneficial. The configuration used consisted of two network nodes connected to one another using two Enterprise Extender transmission groups. One thousand LU-LU RTP pipes were established over the first transmission group. For V1R8, message IST1494I was added to the VTAM message-flooding prevention table. For V1R9, message IST1494I was removed from the VTAM message-flooding prevention table. Also, the new HPRPSMSG start option was coded to 10. During this scenario, the first EE TG was inactivated, causing the one thousand RTP pipes to path switch to the other EE transmission group. This test measures the CPU costs when performed in a V1R8 versus V1R9 environment. In the end, CPU savings ranged from 57 to 66 percent in the V1R9 scenario. These CPU savings will vary depending on the configuration.

If the HPR path switch message reduction and summarization function is enabled, message IST1494I should not be specified in the VTAM message-flooding prevention table. If it is not removed and this new function is enabled, you will not receive the expected number of path switch messages on the system console.



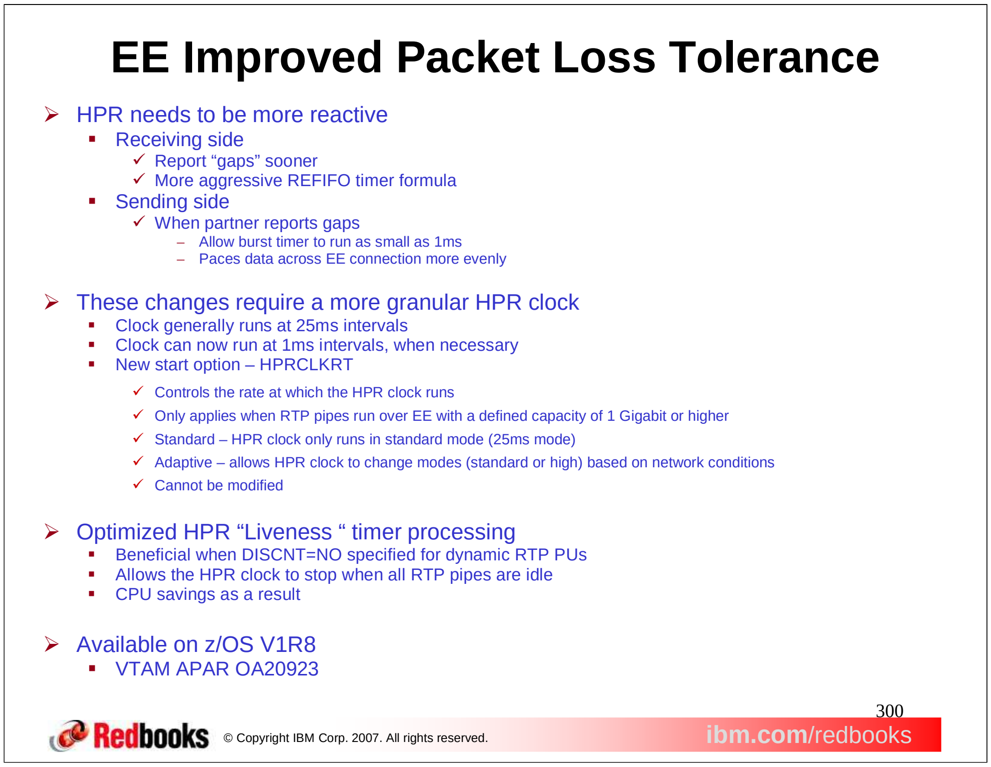
HPR is sensitive to packet loss

- ➢ Enterprise Extender is IBM’s strategic SNA/IP integration mechanism
- ➢ For SNA workloads, you must use EE to access higher-speed DLCs (QDIO)
- ➢ EE connections over a WAN may experience higher packet loss than traditional SNA configurations
- ➢ HPR is sensitive to packet loss
  - ▪ HPR retransmissions
    - ✓ Increased CPU overhead
  - ▪ HPR rate reductions
    - ✓ For high speed connections (i.e. 1Gb and higher), packet loss of 0.25% can significantly reduce throughput
    - ✓ Especially noticeable in streaming workloads
  - ▪ Queue growth
    - ✓ Sending side: Wait-For-Acknowledgement queue grows
    - ✓ Receiving side: Out-Of-Sequence queue grows
    - ✓ Both result in storage growth (TI | T1 | T2 buffers and CSM)

Redbooks ibm.com/redbooks

Enterprise Extender is IBM’s strategic SNA over IP integration mechanism. Depending on the reliability of the IP backbone, some Enterprise Extender connections may experience higher packet loss than traditional SNA configurations.

HPR is sensitive to packet loss. If packet loss occurs this may cause HPR retransmissions, rate reductions and queue growth. As a result, you may see increased CPU overhead, higher storage utilization and significantly reduced throughput for an RTP pipe suffering from packet loss.



EE Improved Packet Loss Tolerance

- HPR needs to be more reactive
  - Receiving side
    - ✓ Report “gaps” sooner
    - ✓ More aggressive REFIFO timer formula
  - Sending side
    - ✓ When partner reports gaps
      - Allow burst timer to run as small as 1ms
      - Paces data across EE connection more evenly
- These changes require a more granular HPR clock
  - Clock generally runs at 25ms intervals
  - Clock can now run at 1ms intervals, when necessary
  - New start option – HPRCLKRT
    - ✓ Controls the rate at which the HPR clock runs
    - ✓ Only applies when RTP pipes run over EE with a defined capacity of 1 Gigabit or higher
    - ✓ Standard – HPR clock only runs in standard mode (25ms mode)
    - ✓ Adaptive – allows HPR clock to change modes (standard or high) based on network conditions
    - ✓ Cannot be modified
- Optimized HPR “Liveness “ timer processing
  - Beneficial when DISCNT=NO specified for dynamic RTP PUs
  - Allows the HPR clock to stop when all RTP pipes are idle
  - CPU savings as a result
- Available on z/OS V1R8
  - VTAM APAR OA20923

300

HPR has been changed to be more tolerant of packet loss. To begin with, a more aggressive REFIFO timer formula has been implemented to allow the receiver to report gaps sooner to the partner. The REFIFO timer is used by RTP pipes to delay reporting missing packets to the partner to avoid unnecessary transmissions. If a missing packet is detected, the RTP pipe will set the REFIFO timer. When the timer expires and the packet is still missing it is reported to the partner so it can be sent again.

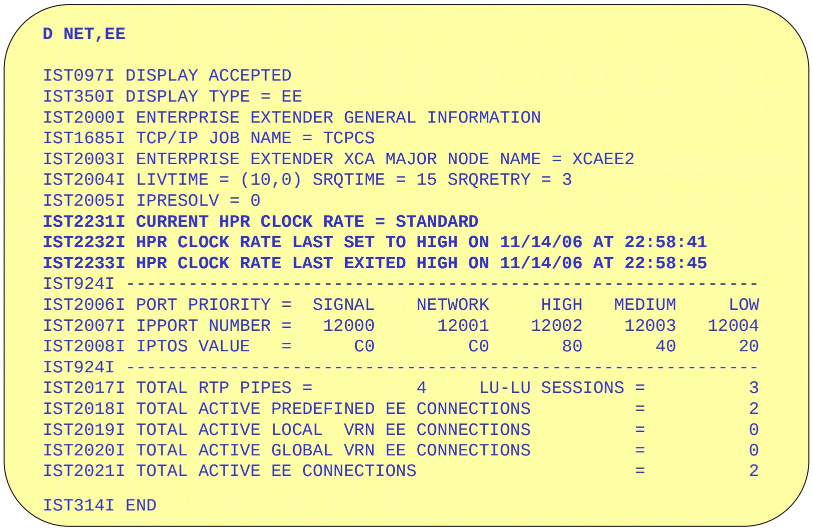
The BURST timer is used by an RTP pipe to pace the data across the connection at specific intervals. Depending on the speed of the RTP connection, the amount of data which can be sent in a burst interval varies. Generally, the BURST timer runs at 25ms intervals. Now, the sending side has been changed to allow the BURST timer to run as small as one millisecond. This will allow the RTP pipe to better pace the data across the connection. When necessary, the HPR clock must now be allowed to run at a one millisecond rate to support these new timer changes.

To support these changes, a new start option, HPRCLKRT, has been introduced to control this function. Specifying STANDARD requires the HPR clock always run in standard mode or twenty-five millisecond mode. When specifying ADAPTIVE, the HPR clock is allowed to change modes based on network conditions. This start option is only valid when RTP pipes are directly connected to an Enterprise Extender link with a defined capacity of one gigabit or higher.

One last change was made to optimize the HPR liveness timer processing. The HPR liveness timer is used to send keep-alive signals to the partner to see if they are still active. If you set DISCNT to no for dynamic RTP physical units, this means the RTP pipe will not inactivate when the last session ends. Instead the RTP pipe stays idle until a new session utilizes it. While it is active, liveness timer processing may still occur. In an environment where there is little to no activity at certain times of the day, there is still HPR clock overhead necessary to perform the liveness timer processing. This liveness timer processing has been optimized for this type of environment to allow the HPR clock to stop when all RTP pipes are idle. If all RTP pipes are idle, this change may save CPU overhead as a result. The liveness optimization change is in the base code and is not controlled by any VTAM start option.

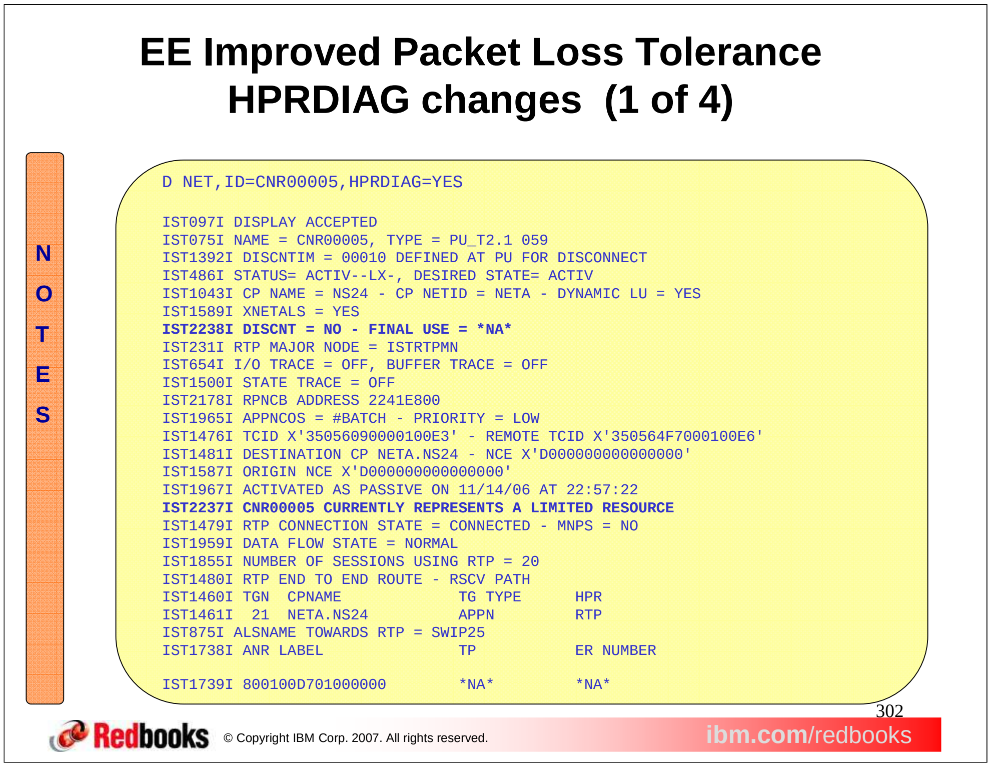
EE Improved Packet Loss Tolerance
Display NET,EE changes

N O T E S



```
D NET,EE

IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2000I ENTERPRISE EXTENDER GENERAL INFORMATION
IST1685I TCP/IP JOB NAME = TCPCS
IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = XCAEE2
IST2004I LIVTIME = (10,0) SRQTIME = 15 SRQRETRY = 3
IST2005I IPRESOLV = 0
IST2231I CURRENT HPR CLOCK RATE = STANDARD
IST2232I HPR CLOCK RATE LAST SET TO HIGH ON 11/14/06 AT 22:58:41
IST2233I HPR CLOCK RATE LAST EXITED HIGH ON 11/14/06 AT 22:58:45
IST924I -------------------------------------------------------------
IST2006I PORT PRIORITY =  SIGNAL    NETWORK     HIGH   MEDIUM     LOW
IST2007I IPPORT NUMBER =   12000      12001    12002    12003   12004
IST2008I IPTOS VALUE   =      C0         C0       80       40      20
IST924I -------------------------------------------------------------
IST2017I TOTAL RTP PIPES =          4      LU-LU SESSIONS =          3
IST2018I TOTAL ACTIVE PREDEFINED EE CONNECTIONS          =          2
IST2019I TOTAL ACTIVE LOCAL  VRN EE CONNECTIONS          =          0
IST2020I TOTAL ACTIVE GLOBAL VRN EE CONNECTIONS          =          0
IST2021I TOTAL ACTIVE EE CONNECTIONS                     =          2

IST314I END
```

This display has been enhanced to display new message IST2231I to indicate the current mode of the HPR clock rate. Message IST2232I displays the last time the HPR clock entered high mode. Message IST2233I displays the last time the HPR clock exited high mode. In this case, since message IST2232I and IST2233I are present we know that the HPRCLKRT start option has been set to adaptive mode, but the current mode of the HPR clock is standard mode.



EE Improved Packet Loss Tolerance HPRDIAG changes (1 of 4)

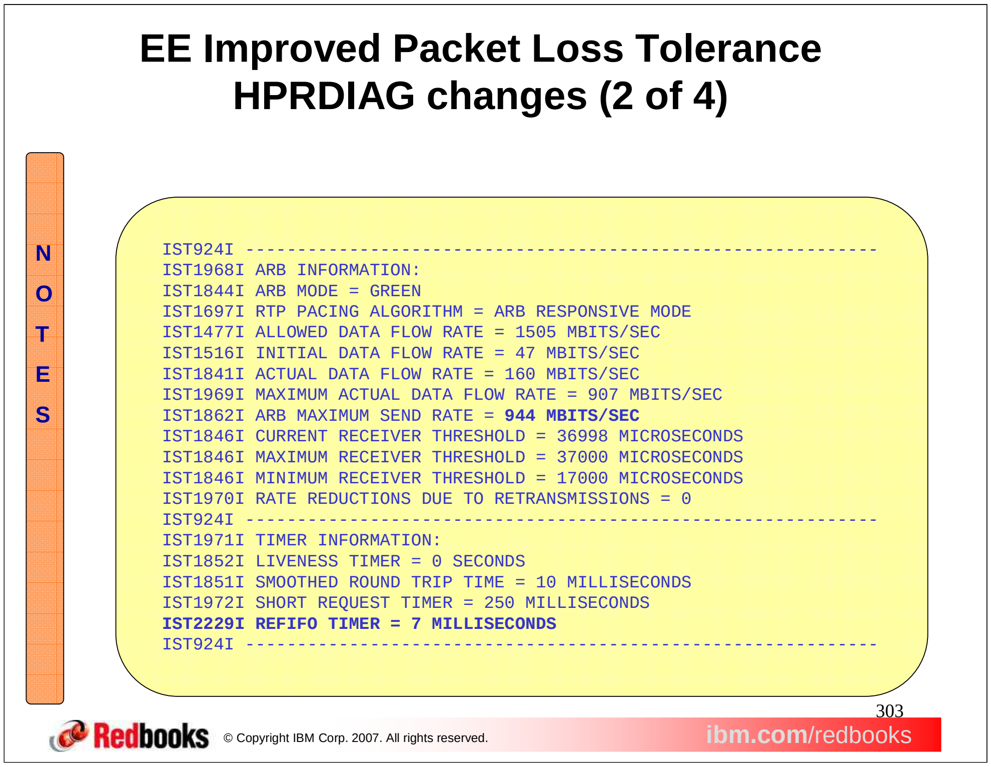
NOTES

```
D NET,ID=CNR00005,HPRDIAG=YES

IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00005, TYPE = PU_T2.1 059
IST1392I DISCNTIM = 00010 DEFINED AT PU FOR DISCONNECT
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = NS24 - CP NETID = NETA - DYNAMIC LU = YES
IST1589I XNETALS = YES
IST2238I DISCNT = NO - FINAL USE = *NA*
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST2178I RPNCB ADDRESS 2241E800
IST1965I APPNCOS = #BATCH - PRIORITY = LOW
IST1476I TCID X'35056090000100E3' - REMOTE TCID X'350564F7000100E6'
IST1481I DESTINATION CP NETA.NS24 - NCE X'D000000000000000'
IST1587I ORIGIN NCE X'D000000000000000'
IST1967I ACTIVATED AS PASSIVE ON 11/14/06 AT 22:57:22
IST2237I CNR00005 CURRENTLY REPRESENTS A LIMITED RESOURCE
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
IST1855I NUMBER OF SESSIONS USING RTP = 20
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  NETA.NS24          APPN         RTP
IST875I ALSNAME TOWARDS RTP = SWIP25
IST1738I ANR LABEL               TP           ER NUMBER

IST1739I 800100D701000000        *NA*         *NA*
```

302

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

Message IST2237I is issued to identify this RTP pipe as a limited resource. For an HPR PU, this really means the underlying DLC has the DISCNT parameter specified as DELAY or YES.



EE Improved Packet Loss Tolerance HPRDIAG changes (2 of 4)

N O T E S

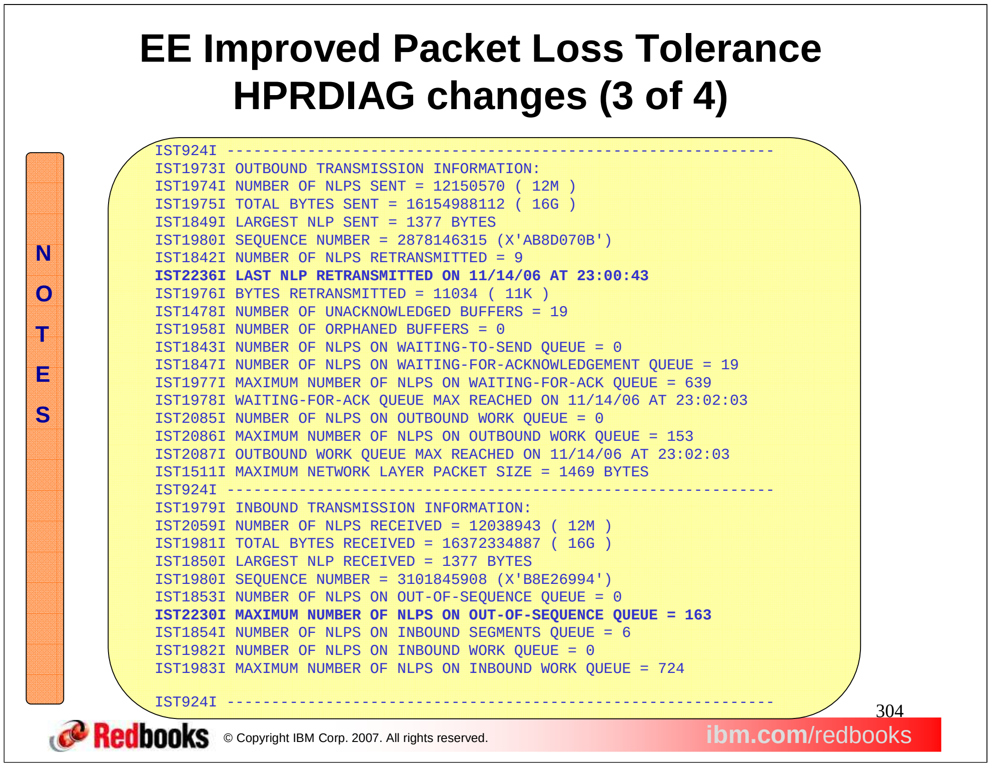
```
IST924I -------------------------------------------------------------
IST1968I ARB INFORMATION:
IST1844I ARB MODE = GREEN
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1477I ALLOWED DATA FLOW RATE = 1505 MBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 47 MBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 160 MBITS/SEC
IST1969I MAXIMUM ACTUAL DATA FLOW RATE = 907 MBITS/SEC
IST1862I ARB MAXIMUM SEND RATE = 944 MBITS/SEC
IST1846I CURRENT RECEIVER THRESHOLD = 36998 MICROSECONDS
IST1846I MAXIMUM RECEIVER THRESHOLD = 37000 MICROSECONDS
IST1846I MINIMUM RECEIVER THRESHOLD = 17000 MICROSECONDS
IST1970I RATE REDUCTIONS DUE TO RETRANSMISSIONS = 0
IST924I -------------------------------------------------------------
IST1971I TIMER INFORMATION:
IST1852I LIVENESS TIMER = 0 SECONDS
IST1851I SMOOTHED ROUND TRIP TIME = 10 MILLISECONDS
IST1972I SHORT REQUEST TIMER = 250 MILLISECONDS
IST2229I REFIFO TIMER = 7 MILLISECONDS
IST924I -------------------------------------------------------------
```

Redbooks

ibm.com/redbooks

New message IST2229I displays the current value of the HPR REFIFO timer. This is the amount of time this end of the HPR pipe waits before reporting missing packets (gaps) to the partner.

Notice that message IST1862I indicates an ARB maximum send rate of 944 MBITS/SEC. When CAPACITY is coded at 1000M or 1G on the underlying PU definition, the connection actually displays 944 MBITS/SEC.



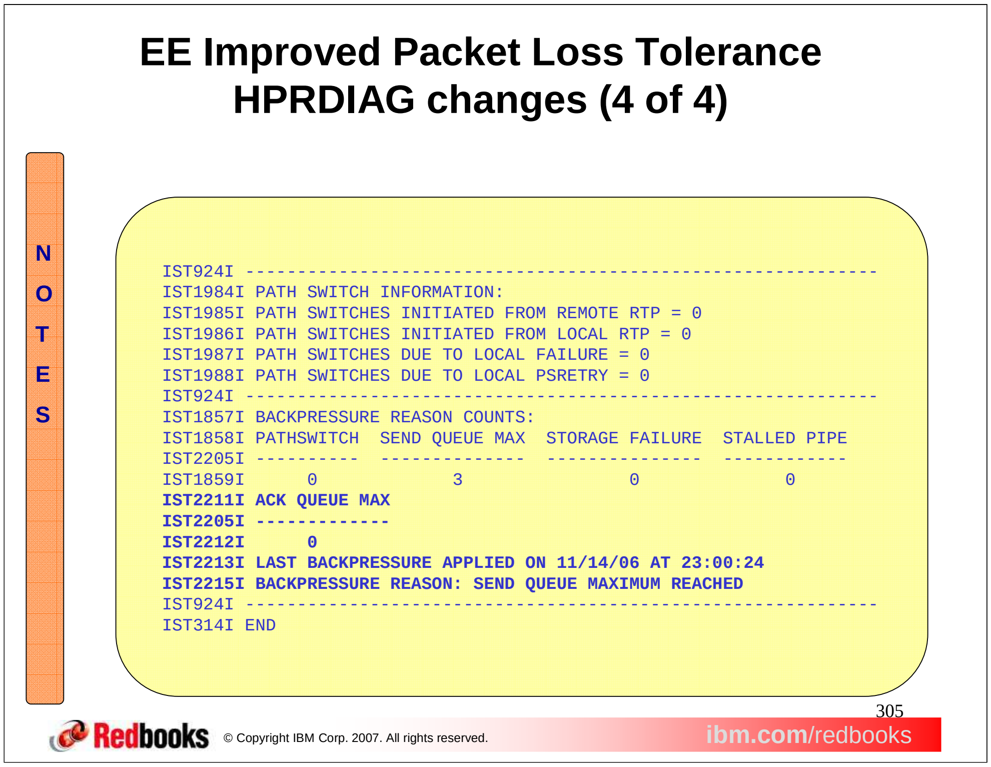
EE Improved Packet Loss Tolerance HPRDIAG changes (3 of 4)

N O T E S

```
IST924I -------------------------------------------------------------
IST1973I OUTBOUND TRANSMISSION INFORMATION:
IST1974I NUMBER OF NLPS SENT = 12150570 ( 12M )
IST1975I TOTAL BYTES SENT = 16154988112 ( 16G )
IST1849I LARGEST NLP SENT = 1377 BYTES
IST1980I SEQUENCE NUMBER = 2878146315 (X'AB8D070B')
IST1842I NUMBER OF NLPS RETRANSMITTED = 9
IST2236I LAST NLP RETRANSMITTED ON 11/14/06 AT 23:00:43
IST1976I BYTES RETRANSMITTED = 11034 ( 11K )
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 19
IST1958I NUMBER OF ORPHANED BUFFERS = 0
IST1843I NUMBER OF NLPS ON WAITING-TO-SEND QUEUE = 0
IST1847I NUMBER OF NLPS ON WAITING-FOR-ACKNOWLEDGEMENT QUEUE = 19
IST1977I MAXIMUM NUMBER OF NLPS ON WAITING-FOR-ACK QUEUE = 639
IST1978I WAITING-FOR-ACK QUEUE MAX REACHED ON 11/14/06 AT 23:02:03
IST2085I NUMBER OF NLPS ON OUTBOUND WORK QUEUE = 0
IST2086I MAXIMUM NUMBER OF NLPS ON OUTBOUND WORK QUEUE = 153
IST2087I OUTBOUND WORK QUEUE MAX REACHED ON 11/14/06 AT 23:02:03
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 1469 BYTES
IST924I -------------------------------------------------------------
IST1979I INBOUND TRANSMISSION INFORMATION:
IST2059I NUMBER OF NLPS RECEIVED = 12038943 ( 12M )
IST1981I TOTAL BYTES RECEIVED = 16372334887 ( 16G )
IST1850I LARGEST NLP RECEIVED = 1377 BYTES
IST1980I SEQUENCE NUMBER = 3101845908 (X'B8E26994')
IST1853I NUMBER OF NLPS ON OUT-OF-SEQUENCE QUEUE = 0
IST2230I MAXIMUM NUMBER OF NLPS ON OUT-OF-SEQUENCE QUEUE = 163
IST1854I NUMBER OF NLPS ON INBOUND SEGMENTS QUEUE = 6
IST1982I NUMBER OF NLPS ON INBOUND WORK QUEUE = 0
IST1983I MAXIMUM NUMBER OF NLPS ON INBOUND WORK QUEUE = 724

IST924I -------------------------------------------------------------
```

ibm.com/redbooks

New message IST2236I displays the date and time of when the last NLP was retransmitted. Message IST2230I displays the high-water-mark for the RTP out-of-sequence queue. In this example 163 network layer packets (NLPs) have been on the out-of-sequence queue since the HPR pipe was started.



EE Improved Packet Loss Tolerance HPRDIAG changes (4 of 4)

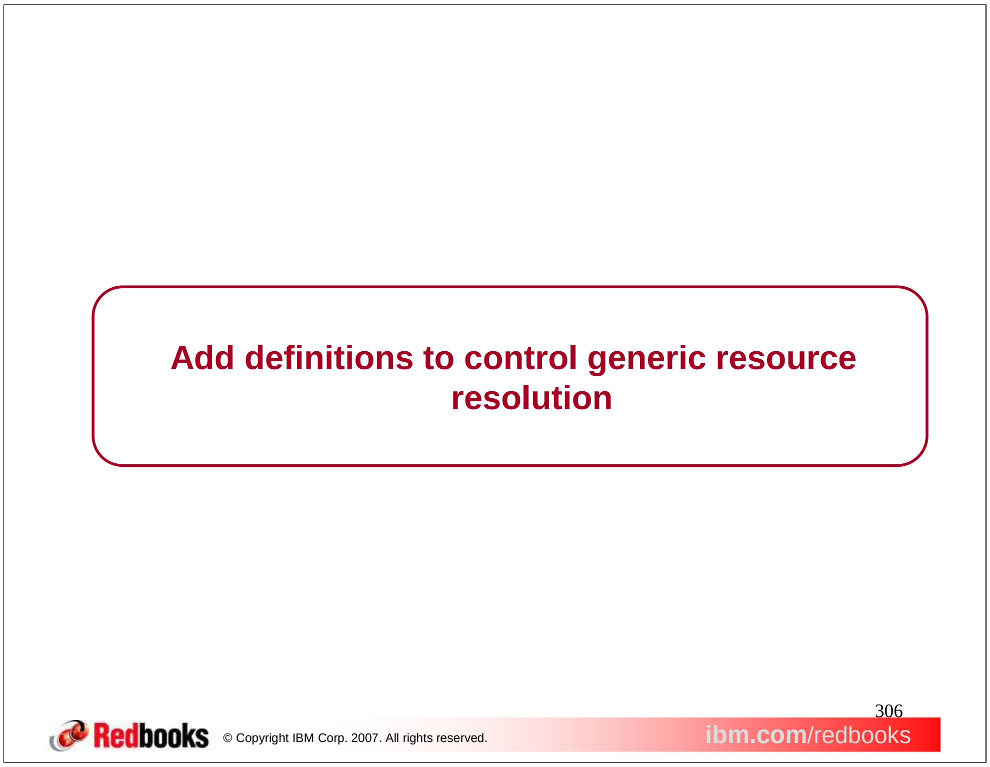
NOTES

```
IST924I -------------------------------------------------------------
IST1984I PATH SWITCH INFORMATION:
IST1985I PATH SWITCHES INITIATED FROM REMOTE RTP = 0
IST1986I PATH SWITCHES INITIATED FROM LOCAL RTP = 0
IST1987I PATH SWITCHES DUE TO LOCAL FAILURE = 0
IST1988I PATH SWITCHES DUE TO LOCAL PSRETRY = 0
IST924I -------------------------------------------------------------
IST1857I BACKPRESSURE REASON COUNTS:
IST1858I PATHSWITCH  SEND QUEUE MAX  STORAGE FAILURE  STALLED PIPE
IST2205I ----------  --------------  ---------------  ------------
IST1859I      0             3                0              0
IST2211I ACK QUEUE MAX
IST2205I -------------
IST2212I      0
IST2213I LAST BACKPRESSURE APPLIED ON 11/14/06 AT 23:00:24
IST2215I BACKPRESSURE REASON: SEND QUEUE MAXIMUM REACHED
IST924I -------------------------------------------------------------
IST314I END
```

305

The HPR backpressure section now displays the date and time of when back pressure was applied to this RTP pipe. If also displays the latest reason for the backpressure. This section also displays a new backpressure count titled “ACK QUEUE MAX”. This HPR backpressure is applied when the RTP waiting for acknowledgement queue reaches a depth of ten thousand elements. The backpressure is relieved when the queue depth returns to five thousand or less elements.

Message IST12212I displays the number of times that this Rapid Transfer Protocol (RTP) went into backpressure (holding up outbound data transmission) since the HPR PU was activated.

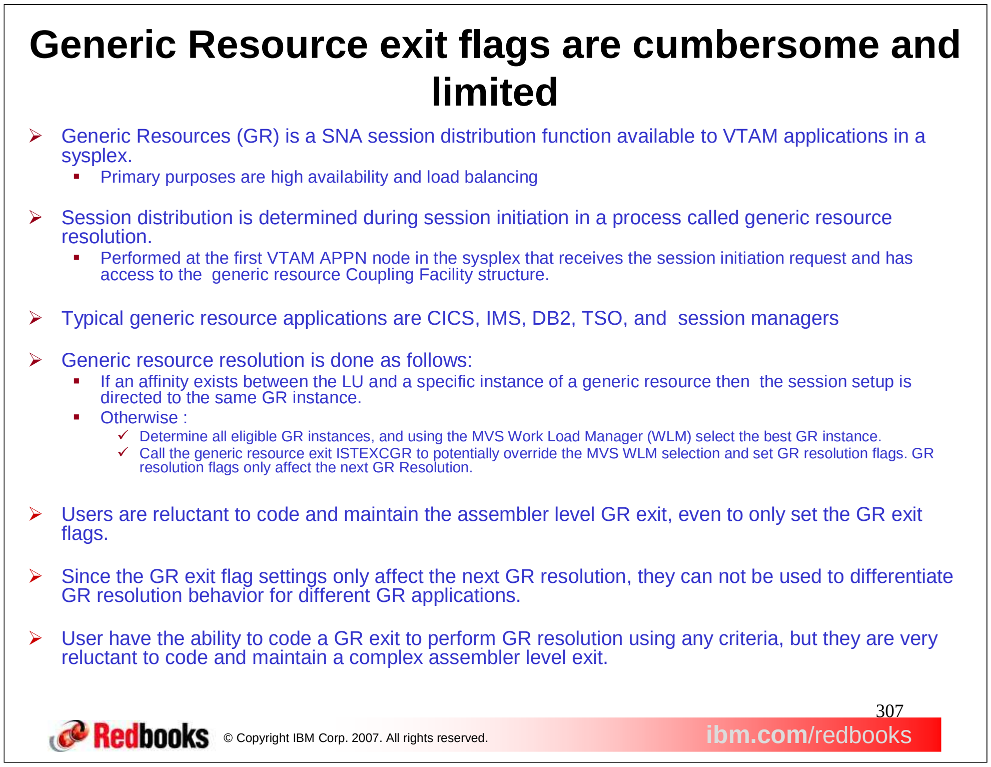


Add definitions to control generic resource resolution

306

ibm.com/redbooks

This section describes the new function that adds definitions to control generic resource resolution.



Generic Resource exit flags are cumbersome and limited

- Generic Resources (GR) is a SNA session distribution function available to VTAM applications in a sysplex.
  - Primary purposes are high availability and load balancing
- Session distribution is determined during session initiation in a process called generic resource resolution.
  - Performed at the first VTAM APPN node in the sysplex that receives the session initiation request and has access to the generic resource Coupling Facility structure.
- Typical generic resource applications are CICS, IMS, DB2, TSO, and session managers
- Generic resource resolution is done as follows:
  - If an affinity exists between the LU and a specific instance of a generic resource then the session setup is directed to the same GR instance.
  - Otherwise :
    - ✓ Determine all eligible GR instances, and using the MVS Work Load Manager (WLM) select the best GR instance.
    - ✓ Call the generic resource exit ISTEXCGR to potentially override the MVS WLM selection and set GR resolution flags. GR resolution flags only affect the next GR Resolution.
- Users are reluctant to code and maintain the assembler level GR exit, even to only set the GR exit flags.
- Since the GR exit flag settings only affect the next GR resolution, they can not be used to differentiate GR resolution behavior for different GR applications.
- User have the ability to code a GR exit to perform GR resolution using any criteria, but they are very reluctant to code and maintain a complex assembler level exit.

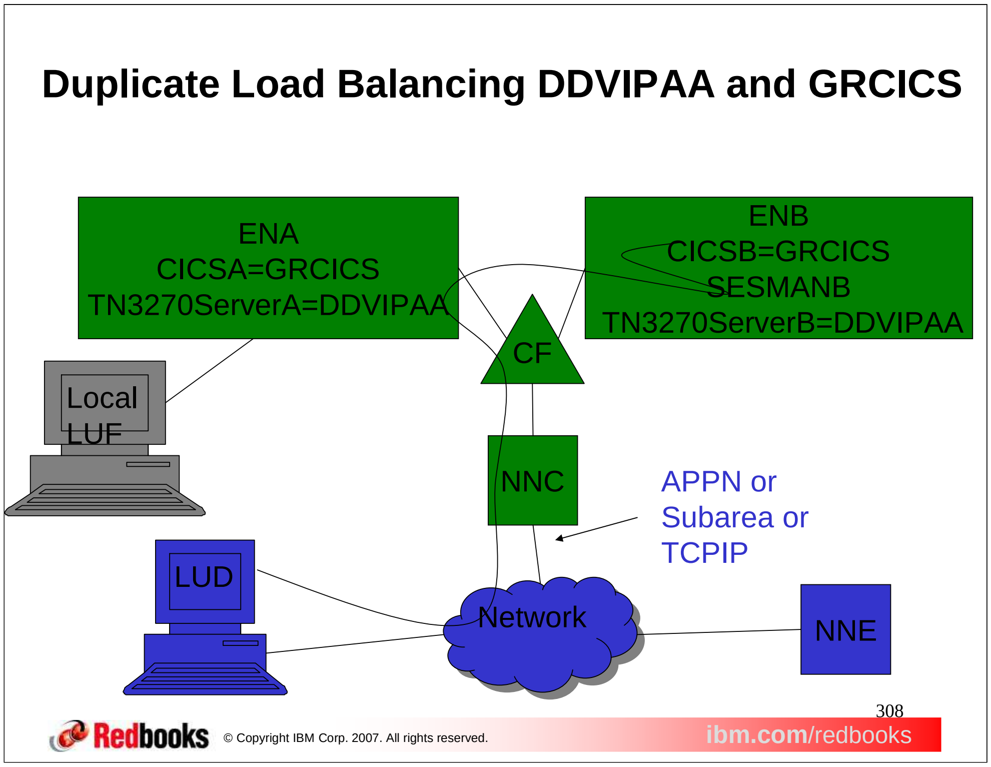
307

Redbooks ibm.com/redbooks

Generic Resources (GR) is a SNA session distribution function available to VTAM applications in a sysplex. Its primary purposes are high availability and load balancing. It does this by allowing multiple applications to be known by the same generic name. Applications must be modified to use SNA API commands to register and manage their generic name. When users logon using a generic name the session is distributed among eligible generic resource applications.

Generic Resources is an expansion of the older VTAM USERVAR function. For those of you familiar with TCP/IP, Generic Resources is analogous to the Distributed Dynamic Virtual Internet Protocol Address (DDVIPA) function in TCP/IP. The default generic resource resolution process is to first use an affinity to direct sessions from the same LU to the same generic resource instance. An affinity is created when the first session between an LU and a generic resource is started. An affinity maps the LU name and generic resource name to a specific instance of the generic resource.   If no affinity has been created yet, then the MVS Work Load Manager is called to identify the best generic resource instance. If the generic resource exit (ISTEXCGR) is active then it is called to potentially select a different generic resource instance than was selected by the MVS Work Load Manager and set generic resource resolution flags affecting the next generic resource resolution.
If no affinity exists then generic resource resolution can be modified by the GR resolution flags that are set in the GR Exit. If the GR flag GRRFNPLL is OFF (default) and if the Origin Logical Unit (OLU) is a Local SNA or Local non-SNA LU on this host, then prefer generic resources on this host. Prefer means that if one or more GR instances is active on this host then only these GR instances will be considered for GR resolution. However if no GR instance is active on this host or GRRFNPLL is ON then all active GR instances are eligible for GR resolution.   If the GR flag GRRFNPLA is OFF (default) and if the Origin Logical Unit (OLU) is an application on this host, then prefer generic resources on this host. If the GR flag GRRFWLMX is ON (default), then call the MVS Work Load Manager to identify the best eligible generic resource instance. Otherwise identify the best GR instance based on lowest active and pending session counts. If the GR flag GRRFUVX is ON (default ON for first call then set OFF by the default GR exit), then call the Generic Resource exit to identify the desired generic resource and set GR exit flags that will influence the next GR resolution. If all defaults are used and the default GR exit is not modified to select a generic resource instance, then the GR identified by WLM will be used.

The main points are that setting generic resource resolution flags in assembler level programs is not very usable. Also the flags cannot be changed in any meaningful way for different generic resources, since you can never know what generic resource will need to be resolved for the next session.



Duplicate Load Balancing DDVIPAA and GRCICS

ENA
CICSA=GRCICS
TN3270ServerA=DDVIPAA

ENB
CICSB=GRCICS
SESMANB
TN3270ServerB=DDVIPAA

CF

Local
LUF

NNC

APPN or
Subarea or
TCPIP

LUD

Network

NNE

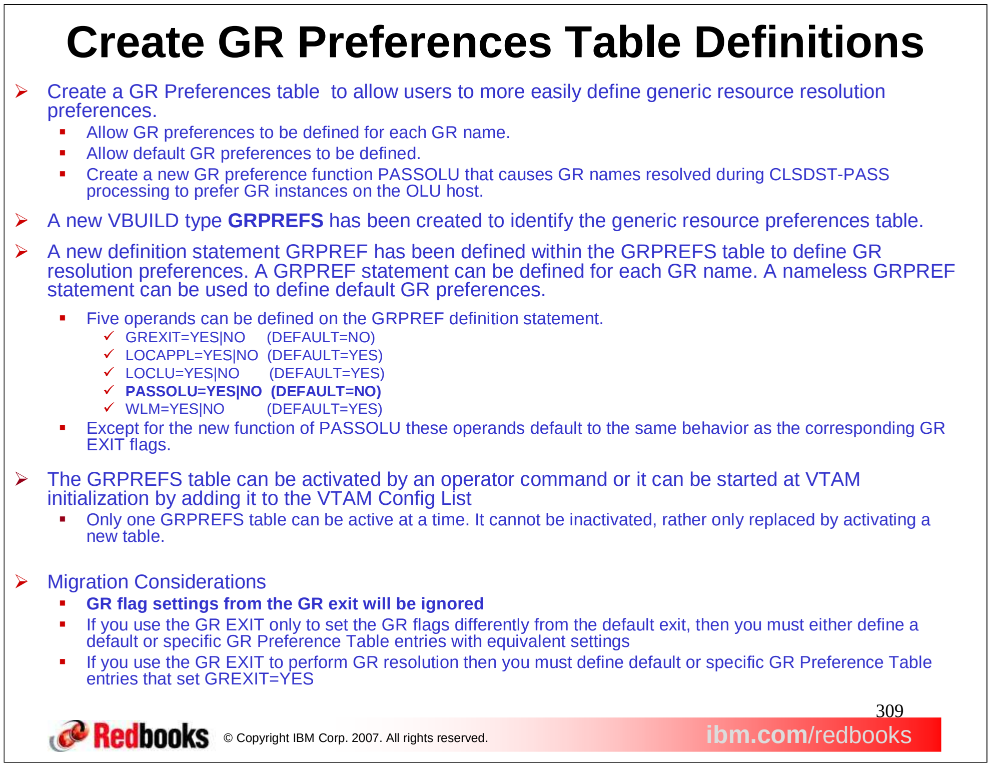
308

Redbooks

ibm.com/redbooks

Multiple load balancing functions for what are different stages of the same session could result in unnecessarily long session paths. During session setup it is possible that a session will pass through multiple load balancing functions (eg. TN3270 using DDVIPA, Session Manager using GR, and final application using GR). This could result in a final session path that needlessly passes through multiple hosts with little or even detrimental workload distribution value.

This shows a TCPIP connection that has been distributed to the TN3270 server A using DDVIPA workload distribution. In turn a SNA session is started from TN3270 server A to session manager SESMANB. A target generic resource application GRCICS is then selected at the session manager and it does a CLSDST-PASS to generic resource GRCICS. Generic resource resolution selects generic resource instance CICSB. Given that load balancing was done once for the connection to TN3270 server A it may be beneficial for the generic resource resolution done during CLSDST PASS processing at ENB to prefer a generic resource instance on the Origin Logical Unit host: that is CICSA on ENA. There is no way to do this today, unless you make substantial changes to the generic resource exit.



Create GR Preferences Table Definitions

- Create a GR Preferences table to allow users to more easily define generic resource resolution preferences.
  - Allow GR preferences to be defined for each GR name.
  - Allow default GR preferences to be defined.
  - Create a new GR preference function PASSOLU that causes GR names resolved during CLSDST-PASS processing to prefer GR instances on the OLU host.
- A new VBUILD type **GRPREFS** has been created to identify the generic resource preferences table.
- A new definition statement GRPREF has been defined within the GRPREFS table to define GR resolution preferences. A GRPREF statement can be defined for each GR name. A nameless GRPREF statement can be used to define default GR preferences.
  - Five operands can be defined on the GRPREF definition statement.
    - ✓ GREXIT=YES|NO   (DEFAULT=NO)
    - ✓ LOCAPPL=YES|NO (DEFAULT=YES)
    - ✓ LOCLU=YES|NO   (DEFAULT=YES)
    - ✓ **PASSOLU=YES|NO (DEFAULT=NO)**
    - ✓ WLM=YES|NO   (DEFAULT=YES)
  - Except for the new function of PASSOLU these operands default to the same behavior as the corresponding GR EXIT flags.
- The GRPREFS table can be activated by an operator command or it can be started at VTAM initialization by adding it to the VTAM Config List
  - Only one GRPREFS table can be active at a time. It cannot be inactivated, rather only replaced by activating a new table.
- Migration Considerations
  - **GR flag settings from the GR exit will be ignored**
  - If you use the GR EXIT only to set the GR flags differently from the default exit, then you must either define a default or specific GR Preference Table entries with equivalent settings
  - If you use the GR EXIT to perform GR resolution then you must define default or specific GR Preference Table entries that set GREXIT=YES

309

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

You can now create a generic resources preferences table to define generic resource preferences for each generic resource name.

The PASSOLU generic resource resolution preference was created to allow generic resource resolution to prefer generic resource instances on the OLU host. This is most useful for session managers in the sysplex that CLSDST-PASS to generic resources.

The new VBUILD type is GRPREFS. The new definition statement GRPREF can be used to identify the generic resource preferences of each generic name. A nameless GRPREF can be defined to identify default generic resource preferences.

The old generic resource exit flags map functionally to the new generic resource preferences operands. Generic resource preference PASSOLU is the new generic resource preferences operand.

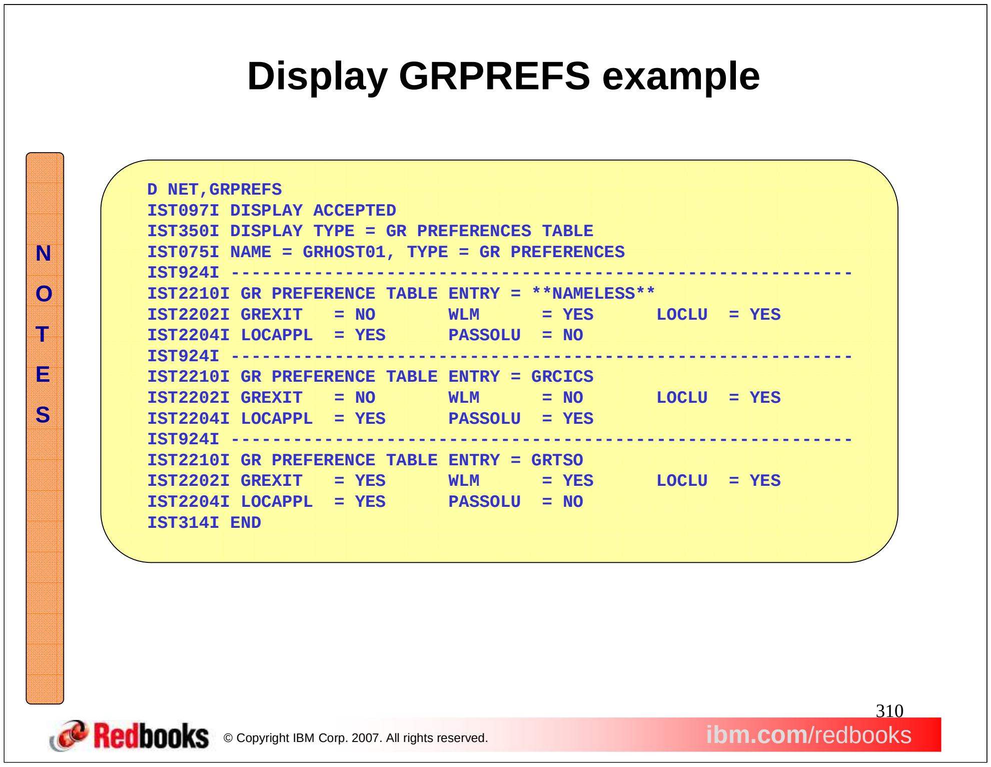
PASSOLU can be specified with a value of YES or NO. A value of YES - For third-party-initiated (CLSDST PASS) sessions, indicates generic resource resolution will prefer generic resource instances located on the OLU host (the host that has the session that is being passed). If no generic resource instances are available on the OLU host, then all instances of the generic resource are eligible for resolution. A value of NO - Indicates all instances of the generic resource are eligible for resolution (default).

PASSOLU does not correspond to any old generic resource exit flag function. PASSOLU could be useful if the original OLU had been load balanced before initiating a session to a session manager that will do a CLOSE DEST PASS to a generic resource. Note that both the LOCAPPL and LOCLU generic resource preferences could affect the PASSOLU preference. If either of the LOCAPPL or PASSOLU preferences are set to YES, then a CLOSE DEST PASS session from a local application to a generic resource will prefer a generic resource on the local host. If either of the LOCLU or PASSOLU preferences are set to YES, then a CLOSE DEST PASS session from a local LU to a generic resource will prefer a generic resource on the local host.

You can activate a GRPREFS table using the VARY NET,ACT,ID= command where the name of the table is the VTAMLST member name that contains the generic resource preferences definitions.

You can also start the GRPREFS table using the VTAM Config List using the same name. Since a table cannot be inactivated, to effectively inactivate a table activate a generic resource preferences table with a nameless entry and no operands.

The primary migration impact will be for users that use the exit today to set the generic resource exit flags differently than the settings in the default generic resource exit. If so then at a minimum you must code a default generic resource preference table entry using a nameless entry with generic resource preferences that match your current generic resource exit flag settings. If your generic resource exit does generic resource resolution you must also set the generic resource preference GREXIT=YES.



Display GRPREFS example

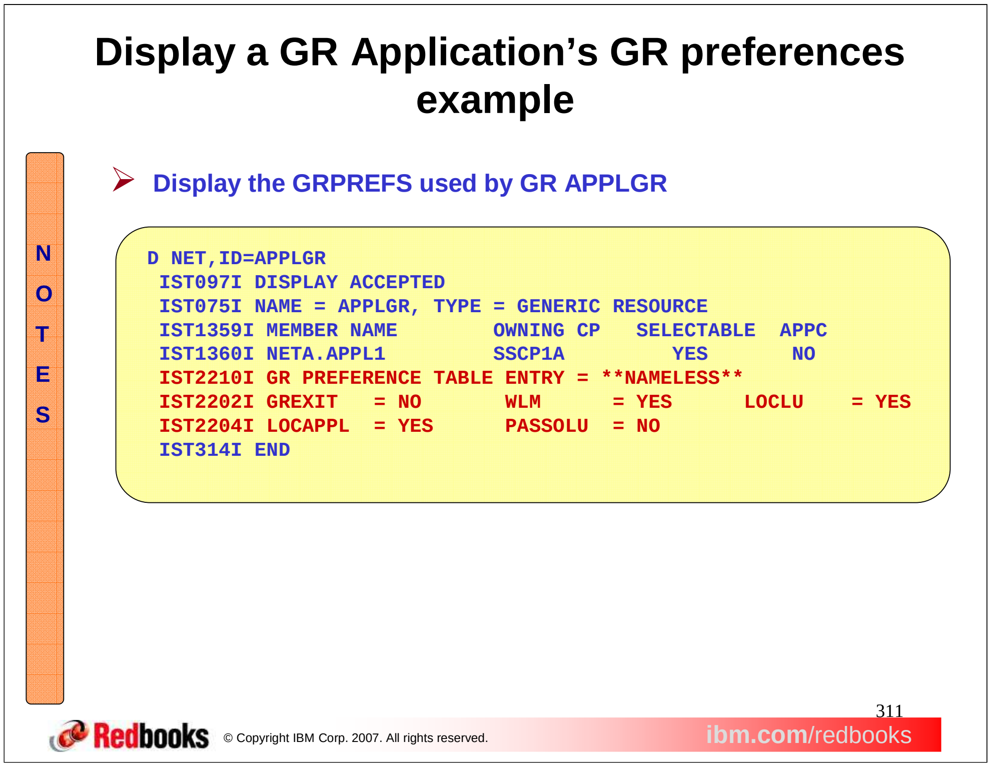
N O T E S

```
D NET,GRPREFS
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = GR PREFERENCES TABLE
IST075I NAME = GRHOST01, TYPE = GR PREFERENCES
IST924I ------------------------------------------------------------
IST2210I GR PREFERENCE TABLE ENTRY = **NAMELESS**
IST2202I GREXIT   = NO       WLM      = YES      LOCLU  = YES
IST2204I LOCAPPL  = YES      PASSOLU  = NO
IST924I ------------------------------------------------------------
IST2210I GR PREFERENCE TABLE ENTRY = GRCICS
IST2202I GREXIT   = NO       WLM      = NO       LOCLU  = YES
IST2204I LOCAPPL  = YES      PASSOLU  = YES
IST924I ------------------------------------------------------------
IST2210I GR PREFERENCE TABLE ENTRY = GRTSO
IST2202I GREXIT   = YES      WLM      = YES      LOCLU  = YES
IST2204I LOCAPPL  = YES      PASSOLU  = NO
IST314I END
```

Redbooks ibm.com/redbooks

You can display the entire GRPREFS table using the D NET,GRPREFS command. The generic resource preferences definition without a name is designated in the display by **NAMELESS**.

The **NAMELESS** entry identifies the defined default generic resource preferences. If no generic preference table is defined to VTAM the generic resource preferences displayed will indicate **DEFAULT**.

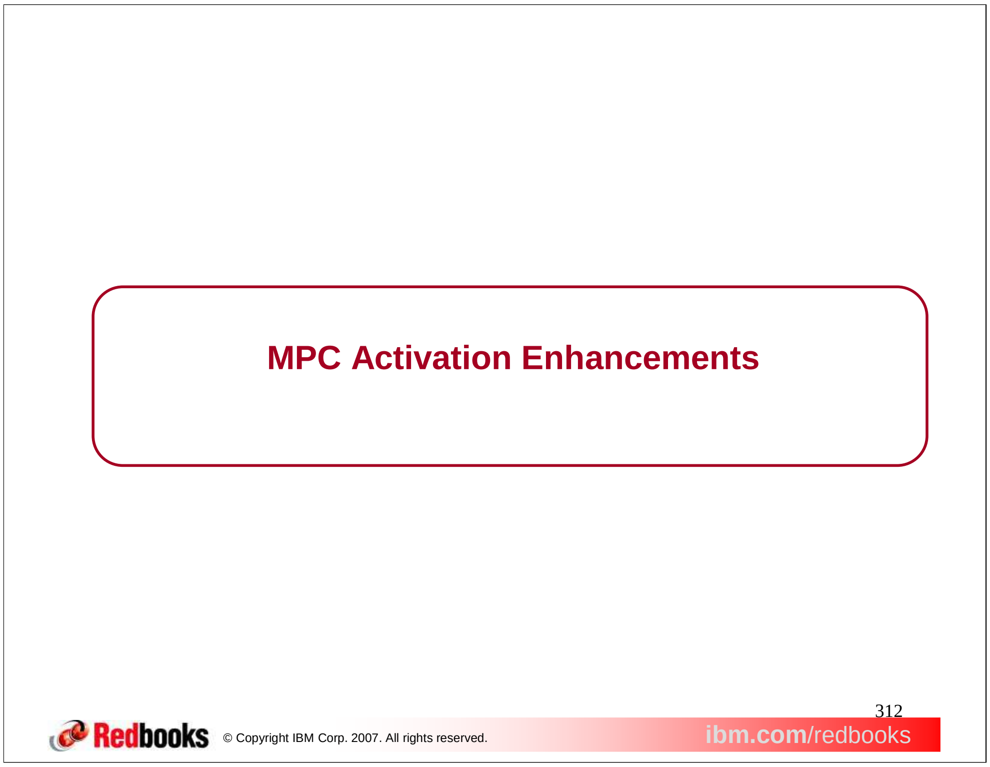


Display a GR Application’s GR preferences example

- Display the GRPREFS used by GR APPLGR

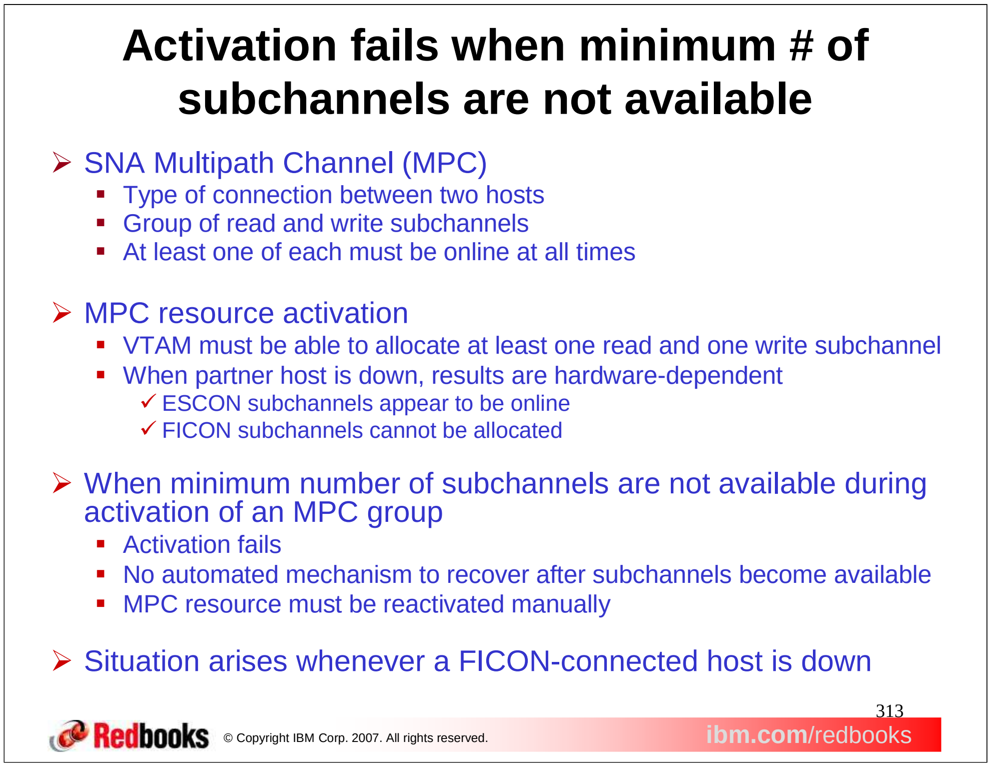
```
D NET,ID=APPLGR
 IST097I DISPLAY ACCEPTED
 IST075I NAME = APPLGR, TYPE = GENERIC RESOURCE
 IST1359I MEMBER NAME        OWNING CP   SELECTABLE  APPC
 IST1360I NETA.APPL1         SSCP1A         YES       NO
 IST2210I GR PREFERENCE TABLE ENTRY = **NAMELESS**
 IST2202I GREXIT   = NO       WLM      = YES      LOCLU    = YES
 IST2204I LOCAPPL  = YES      PASSOLU  = NO
 IST314I END
```

The existing D NET,ID=generic resource name output has been enhanced to include the generic resource preferences associated with the generic name. Messages IST2210I, IST2202I, and IST2204I have been added to the previously existing display.



MPC Activation Enhancements

This section describes the Multi Path Channel (MPC) Activation Enhancements.



Activation fails when minimum # of subchannels are not available

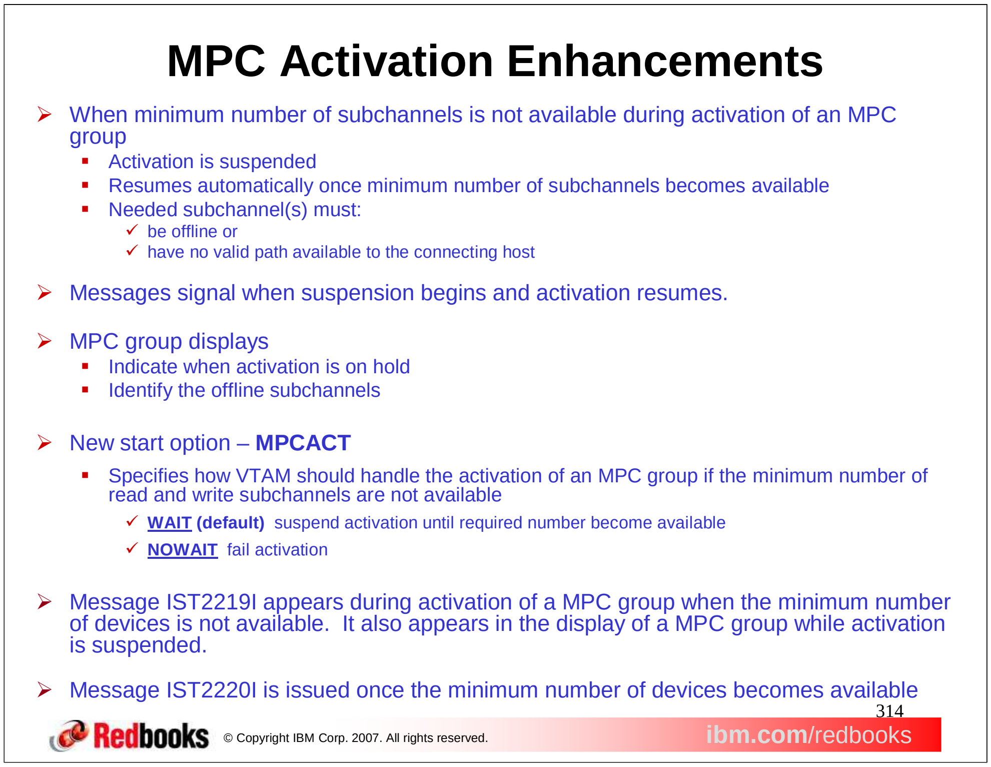
- SNA Multipath Channel (MPC)
  - Type of connection between two hosts
  - Group of read and write subchannels
  - At least one of each must be online at all times
- MPC resource activation
  - VTAM must be able to allocate at least one read and one write subchannel
  - When partner host is down, results are hardware-dependent
    - ✓ ESCON subchannels appear to be online
    - ✓ FICON subchannels cannot be allocated
- When minimum number of subchannels are not available during activation of an MPC group
  - Activation fails
  - No automated mechanism to recover after subchannels become available
  - MPC resource must be reactivated manually
- Situation arises whenever a FICON-connected host is down

VTAM supports multipath channel (MPC) connections between two hosts. An MPC resource consists of a group of read and write subchannels. At least one of each must be online at all times for the group to be functional. Thus, when the MPC resource is first activated (by a VARY ACT command of some sort), VTAM must be able to allocate at least one read and one write subchannel in the process.

When the partner host is down during MPC activation, the information VTAM receives about the subchannels when they are allocated depends on the hardware makeup of the connection. ESCON subchannels appear to be online in this case, so VTAM activates the device and waits for the other side to come up. For FICON subchannels, VTAM gets an indication that there is no valid path to the device, causing allocation of the subchannel to fail. This impacts the ability of VTAM to activate the MPC resource successfully.

Previously, once activation of an MPC group lacking one online read and one online write subchannel failed, there was no mechanism to recover automatically from this failure, even after the needed amount of subchannels became available. Reactivation of the MPC group had to be performed manually.

This situation is especially applicable when a FICON-connected host is down, due to the failures VTAM encounters when allocating the subchannels.



MPC Activation Enhancements

- When minimum number of subchannels is not available during activation of an MPC group
  - Activation is suspended
  - Resumes automatically once minimum number of subchannels becomes available
  - Needed subchannel(s) must:
    - ✓ be offline or
    - ✓ have no valid path available to the connecting host
- Messages signal when suspension begins and activation resumes.
- MPC group displays
  - Indicate when activation is on hold
  - Identify the offline subchannels
- New start option – **MPCACT**
  - Specifies how VTAM should handle the activation of an MPC group if the minimum number of read and write subchannels are not available
    - ✓ **WAIT (default)** suspend activation until required number become available
    - ✓ **NOWAIT** fail activation
- Message IST2219I appears during activation of a MPC group when the minimum number of devices is not available. It also appears in the display of a MPC group while activation is suspended.
- Message IST2220I is issued once the minimum number of devices becomes available

314

Redbooks ibm.com/redbooks

Now activations of MPC groups that fail to meet the one read/one write requirement are put on hold, provided any needed read and/or write subchannel is an offline CTC or one that has no valid path available to the connecting host. The suspension continues until the required minimum number of subchannels becomes available or the group is deactivated. New messages signal when the hold begins and when activation resumes.

The display of an MPC group indicates when its activation is on hold. Other existing output in that display identifies the offline subchannels, so appropriate action can be taken to bring enough of them online to cause activation of the MPC group to complete.
MPC Activation Enhancements is enabled by default. A new, modifiable VTAM connectivity start option (MPCACT) can be used to disable the function whenever manual retry is desired.

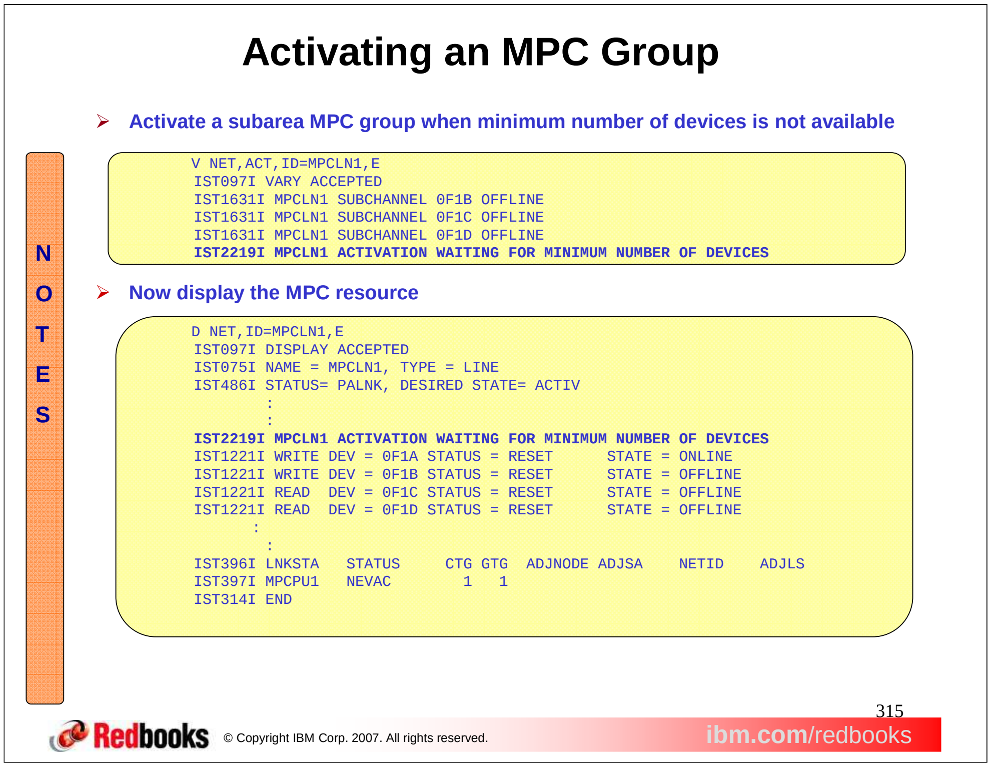
MPCACT=WAIT (default) specifies that activations of MPC subchannel groups are to be suspended if the minimum number of read and write subchannels are not available, either because they are offline or no valid path exists to the connecting host. VTAM will automatically resume activation once the minimum number becomes available.

MPCACT=NOWAIT specifies that VTAM is to fail activations of MPC subchannel groups if the minimum number of read and write subchannels are not available. The system operator must manually retry such activations after the minimum number becomes available.

Note: When modified, the option does not take effect for MPC groups that are in the process of being activated when command is issued until those MPC groups are deactivated and reactivated.

Message IST2219I indicates that activation of an MPC group is suspended waiting for the minimum number of read and write subchannels to become available.

Once the minimum number of devices becomes available, message IST2220I is issued to indicate that VTAM is ready to retry allocation of those subchannels it has detected as being available. If the minimum number still cannot be obtained for some odd reason, activation of the MPC group is suspended again, signaled by the reappearance of IST2219I along with any appropriate IST1631I messages. Otherwise, activation proceeds as in previous releases, ultimately resulting in message IST093I for the MPC line.



Activating an MPC Group

- Activate a subarea MPC group when minimum number of devices is not available

```
V NET,ACT,ID=MPCLN1,E
IST097I VARY ACCEPTED
IST1631I MPCLN1 SUBCHANNEL 0F1B OFFLINE
IST1631I MPCLN1 SUBCHANNEL 0F1C OFFLINE
IST1631I MPCLN1 SUBCHANNEL 0F1D OFFLINE
IST2219I MPCLN1 ACTIVATION WAITING FOR MINIMUM NUMBER OF DEVICES
```

- Now display the MPC resource

```
D NET,ID=MPCLN1,E
IST097I DISPLAY ACCEPTED
IST075I NAME = MPCLN1, TYPE = LINE
IST486I STATUS= PALNK, DESIRED STATE= ACTIV
        :
        :
IST2219I MPCLN1 ACTIVATION WAITING FOR MINIMUM NUMBER OF DEVICES
IST1221I WRITE DEV = 0F1A STATUS = RESET      STATE = ONLINE
IST1221I WRITE DEV = 0F1B STATUS = RESET      STATE = OFFLINE
IST1221I READ  DEV = 0F1C STATUS = RESET      STATE = OFFLINE
IST1221I READ  DEV = 0F1D STATUS = RESET      STATE = OFFLINE
       :
        :
IST396I LNKSTA   STATUS     CTG GTG  ADJNODE ADJSA    NETID    ADJLS
IST397I MPCPU1   NEVAC        1   1
IST314I END
```

N O T E S

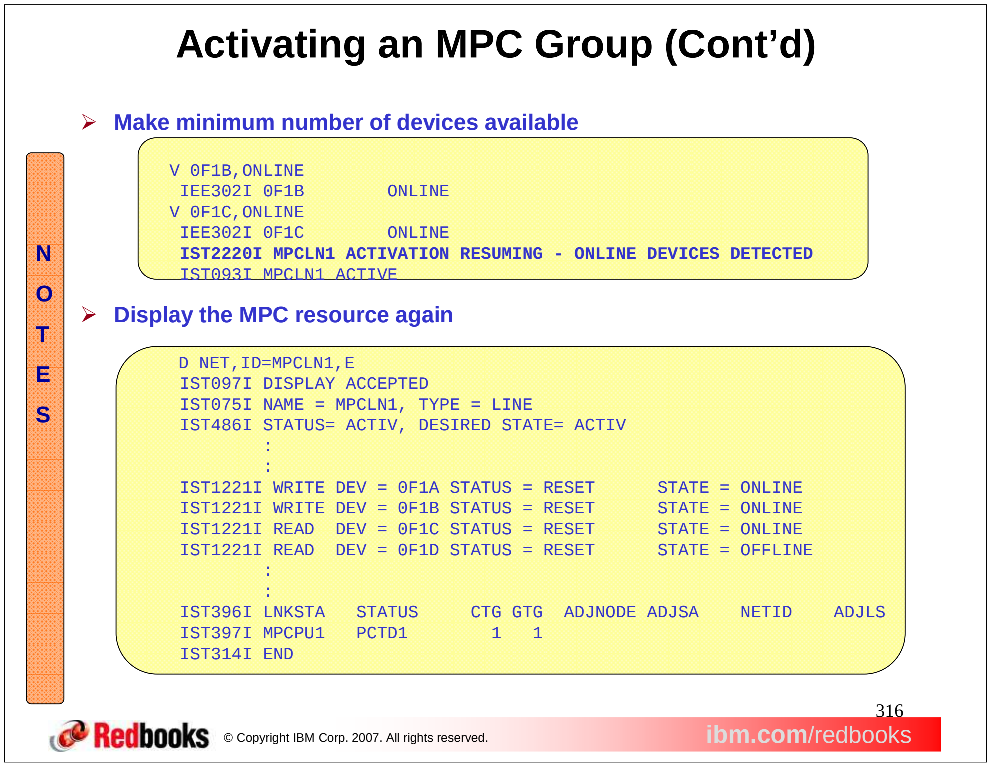
315

Redbooks ibm.com/redbooks

This example that shows how new and existing messages may be used to handle an MPC group that has insufficient subchannels available at activation time.

On this slide, the activation of a subarea MPC line (MPCLN1) is being suspended, as signaled by the IST2219I message. The set of IST1631I messages identifies which subchannels (0F1B-0F1D) are candidates to be brought online to allow activation to complete.

Displaying the subarea MPC line offers another method of identifying the subchannels that are candidates to be brought online to allow activation to complete. A write subchannel (0F1A) is already online, so only a read subchannel is needed in this case. The presence of IST2219I indicates the suspended activation condition, leaving the MPC line (MPCLN1) and PU (MPCPU1) states in **PALNK** (pending ACTLINK) and **NEVAC** (never active), respectively.



Activating an MPC Group (Cont’d)

- Make minimum number of devices available

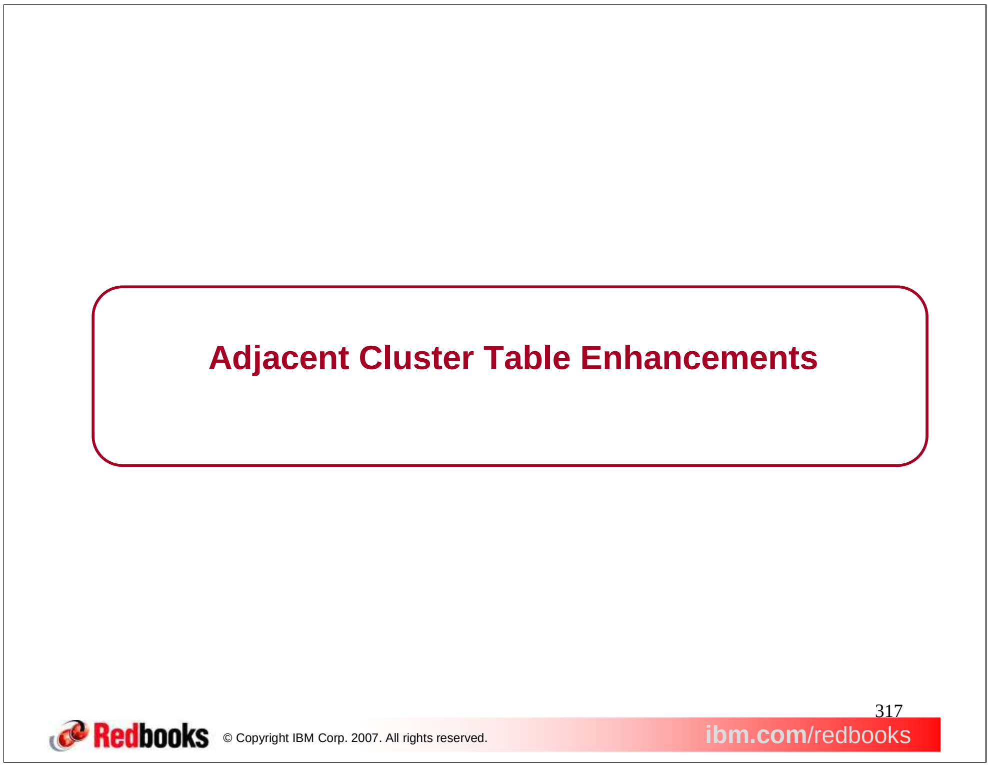
```
V 0F1B,ONLINE
 IEE302I 0F1B        ONLINE
V 0F1C,ONLINE
 IEE302I 0F1C        ONLINE
 IST2220I MPCLN1 ACTIVATION RESUMING - ONLINE DEVICES DETECTED
 IST093I MPCLN1 ACTIVE
```

- Display the MPC resource again

```
D NET,ID=MPCLN1,E
IST097I DISPLAY ACCEPTED
IST075I NAME = MPCLN1, TYPE = LINE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
        :
        :
IST1221I WRITE DEV = 0F1A STATUS = RESET      STATE = ONLINE
IST1221I WRITE DEV = 0F1B STATUS = RESET      STATE = ONLINE
IST1221I READ  DEV = 0F1C STATUS = RESET      STATE = ONLINE
IST1221I READ  DEV = 0F1D STATUS = RESET      STATE = OFFLINE
        :
        :
IST396I LNKSTA   STATUS     CTG GTG  ADJNODE ADJSA    NETID    ADJLS
IST397I MPCPU1   PCTD1        1   1
IST314I END
```

This slide shows activation of the subarea MPC line resuming after a needed read subchannel (0F1C) has been brought online. This is evidenced by the IST2220I message. Note that bringing 0F1B online had no effect, as it is a write subchannel, and one of those (0F1A) was already online.

Displaying the subarea MPC line at this point shows that enough subchannels are online now. The MPC line (MPCLN1) state is now **ACTIV** and the PU (MPCPU1) state becomes **PCTD1** (pending contacted 1), awaiting the partner to come up. Note that IST2219I no longer appears in the display.



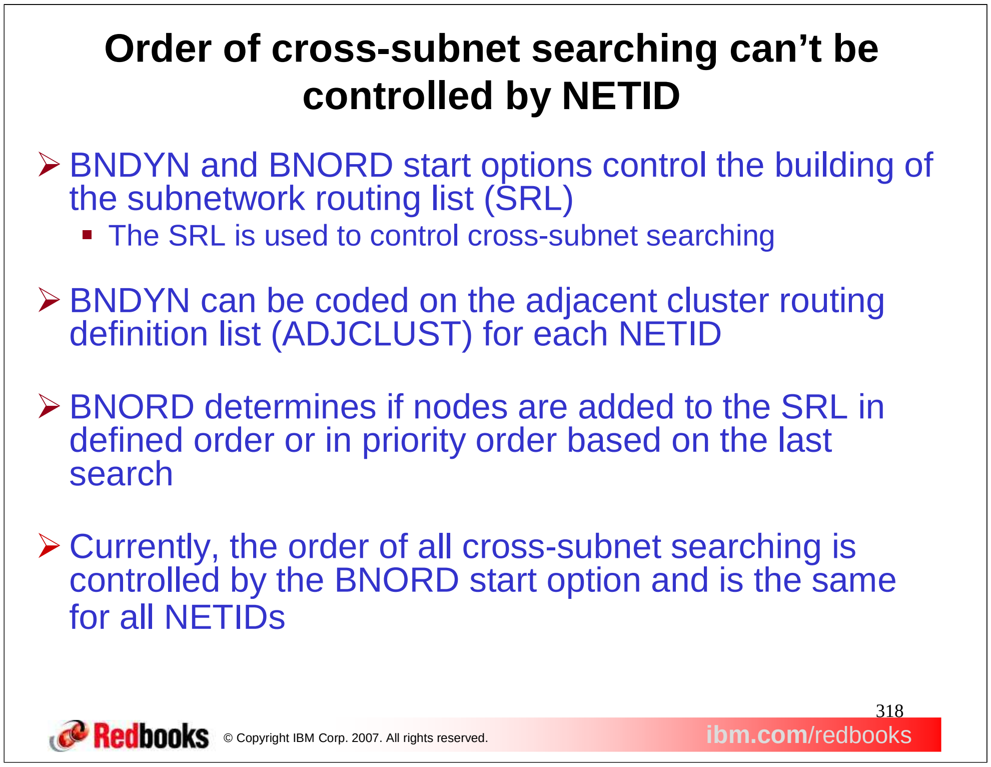
Adjacent Cluster Table Enhancements

317

Redbooks

ibm.com/redbooks

This section describes enhancements to the Adjacent Cluster table definitions.

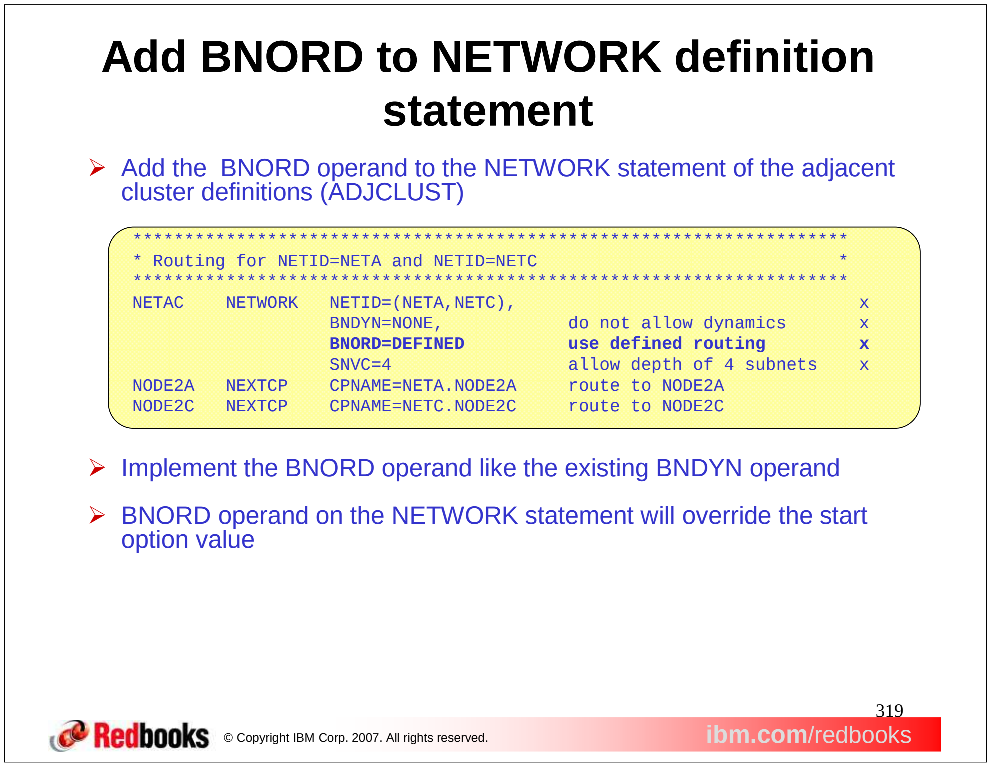


Order of cross-subnet searching can't be controlled by NETID

- ➢ BNDYN and BNORD start options control the building of the subnetwork routing list (SRL)
  - ▪ The SRL is used to control cross-subnet searching
- ➢ BNDYN can be coded on the adjacent cluster routing definition list (ADJCLUST) for each NETID
- ➢ BNORD determines if nodes are added to the SRL in defined order or in priority order based on the last search
- ➢ Currently, the order of all cross-subnet searching is controlled by the BNORD start option and is the same for all NETIDs

The BNDYN and BNORD start options control how the subnetwork routing list (SRL) is built. The SRL is used by a border node to control cross-subnet searching and is built for each search. The adjacent cluster routing definitions allow the user to customize the building of the SRL. The BNDYN start option controls the amount of dynamics used in building the SRL and can be coded on the adjacent cluster routing definition list (ADJCLUST) to customize routing between subnetworks for each NETID. The BNORD start option determines if nodes are added to the SRL in defined order from the adjacent cluster definitions or in priority order based on the last search for a NETID.

Currently, the order of all cross-subnet searching is controlled by the BNORD start option and is the same for all NETIDs. This does not allow the user to control the order of cross-subnet searching by NETID.



Add BNORD to NETWORK definition statement

- Add the BNORD operand to the NETWORK statement of the adjacent cluster definitions (ADJCLUST)

```
**********************************************************************
* Routing for NETID=NETA and NETID=NETC                              *
**********************************************************************
NETAC     NETWORK   NETID=(NETA,NETC),                                 x
                    BNDYN=NONE,             do not allow dynamics      x
                    BNORD=DEFINED           use defined routing        x
                    SNVC=4                  allow depth of 4 subnets   x
NODE2A    NEXTCP    CPNAME=NETA.NODE2A      route to NODE2A
NODE2C    NEXTCP    CPNAME=NETC.NODE2C      route to NODE2C
```

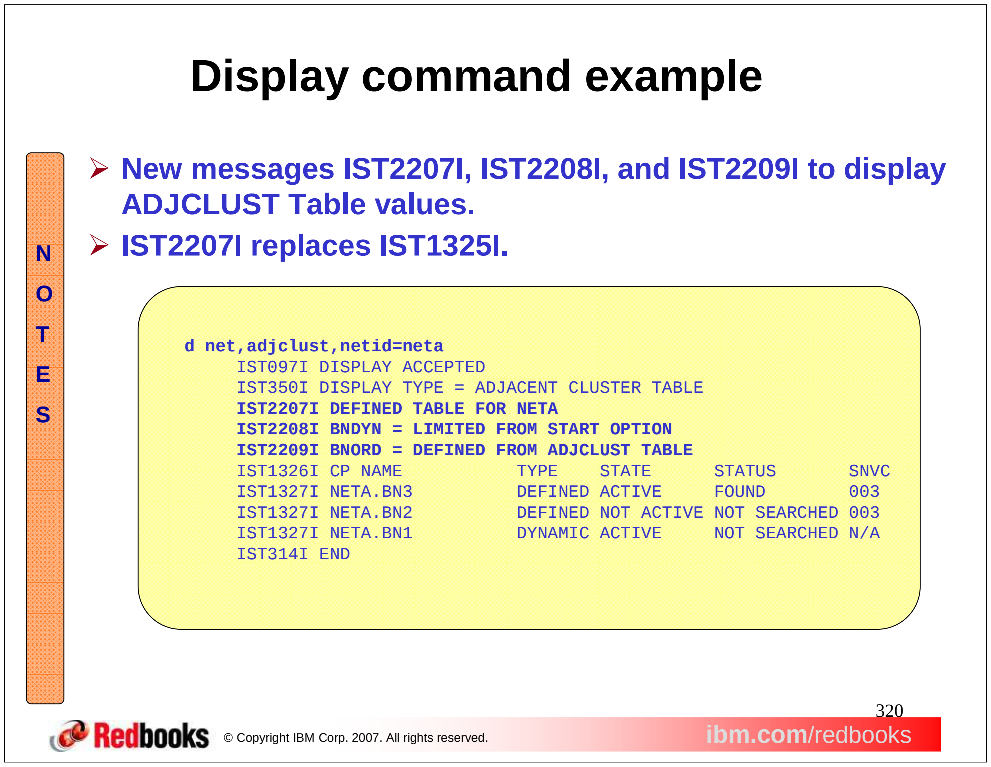
- Implement the BNORD operand like the existing BNDYN operand
- BNORD operand on the NETWORK statement will override the start option value

319

Redbooks ibm.com/redbooks

In z/OS V1R9 we have added the BNORD operand to the NETWORK statement of the adjacent cluster definitions. The BNORD operand is implemented like the existing BNDYN operand on the NETWORK statement. If the BNORD operand is coded on the NETWORK statement, it will override the start option value when building a subnet routing list. Adding the BNORD operand to the NETWORK statement allows the user to specify the order of cross-subnet searching for each NETID coded in the adjacent cluster routing definitions.

The values for BNORD are PRIORITY and DEFINED with PRIORITY being the default start option value. Priority routing indicates that preference is given to nodes for which the most recent search was successful. Defined routing indicates that searches are done in the order specified in the adjacent cluster definition list.



Display command example

N O T E S

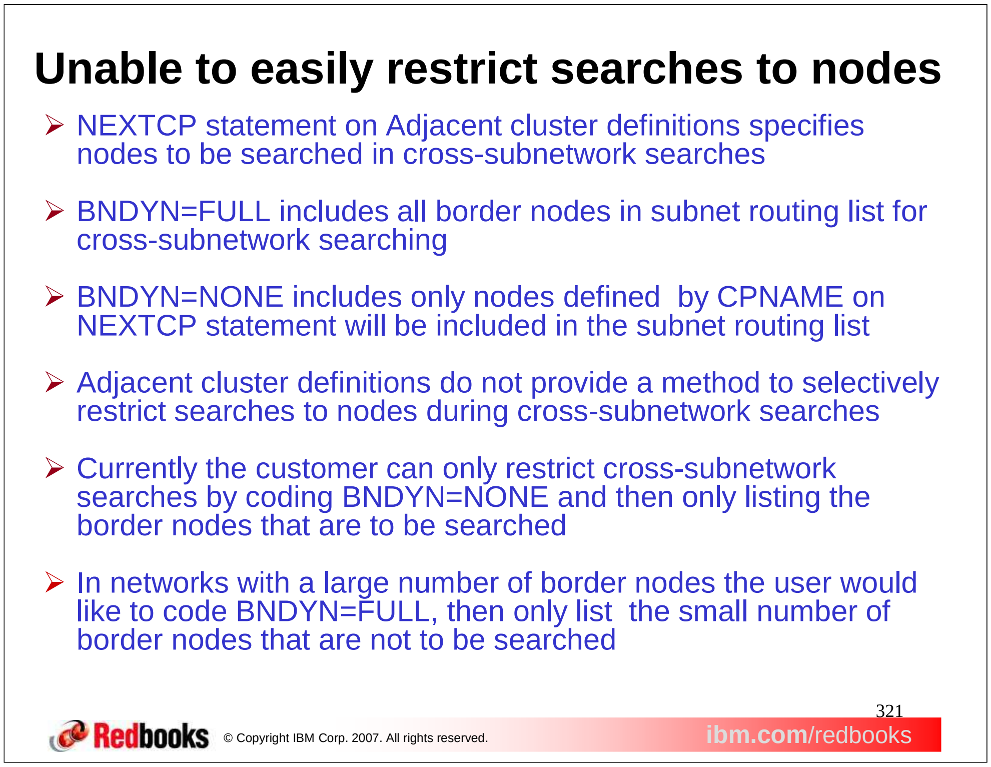
- New messages IST2207I, IST2208I, and IST2209I to display ADJCLUST Table values.
- IST2207I replaces IST1325I.

```
d net,adjclust,netid=neta
     IST097I DISPLAY ACCEPTED
     IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE
     IST2207I DEFINED TABLE FOR NETA
     IST2208I BNDYN = LIMITED FROM START OPTION
     IST2209I BNORD = DEFINED FROM ADJCLUST TABLE
     IST1326I CP NAME            TYPE    STATE      STATUS       SNVC
     IST1327I NETA.BN3           DEFINED ACTIVE     FOUND        003
     IST1327I NETA.BN2           DEFINED NOT ACTIVE NOT SEARCHED 003
     IST1327I NETA.BN1           DYNAMIC ACTIVE     NOT SEARCHED N/A
     IST314I END
```

320

Redbooks ibm.com/redbooks

New messages IST2208I and IST2209I are added to the DISPLAY ADJCLUST to display the BNORD and BNDYN values. The new messages are modeled after the IST1704I and IST1705I messages for SORDER and SSCPORD for the ADJSSCP TABLE. Either IST2208I or IST2209I will be issued to display the value of both the BNDYN and BNORD search control options. IST2208I is issued when the value is obtained from the START OPTION. IST2209I is issued when the value is obtained from the adjacent cluster definition table.
Message IST2207I replaces message IST1325I because the border node dynamics information was expanded and moved to IST2208I and IST2209I.
New messages IST2208I and IST2209I are also issued when detailed locate search failure information is displayed to aid in problem diagnosis. The new messages will indicate the values and origin of BNDYN and BNORD that were used for the search. Detailed locate search information can be displayed by setting the LSIRFMSG and FSIRFMSG START OPTIONS.

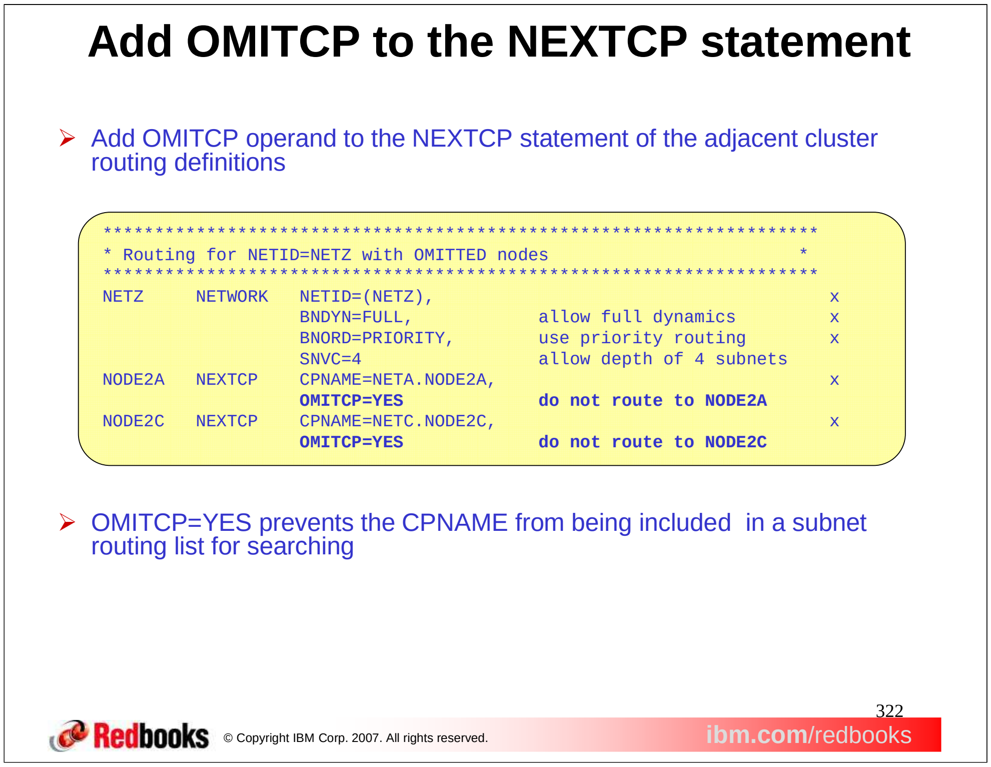


Unable to easily restrict searches to nodes

- NEXTCP statement on Adjacent cluster definitions specifies nodes to be searched in cross-subnetwork searches
- BNDYN=FULL includes all border nodes in subnet routing list for cross-subnetwork searching
- BNDYN=NONE includes only nodes defined by CPNAME on NEXTCP statement will be included in the subnet routing list
- Adjacent cluster definitions do not provide a method to selectively restrict searches to nodes during cross-subnetwork searches
- Currently the customer can only restrict cross-subnetwork searches by coding BNDYN=NONE and then only listing the border nodes that are to be searched
- In networks with a large number of border nodes the user would like to code BNDYN=FULL, then only list the small number of border nodes that are not to be searched

321

The NEXTCP statement on the adjacent cluster routing definitions specifies nodes to be searched during cross-subnet searching. Border node dynamics determines if additional nodes are added to the subnetwork routing list for searching. BNDYN=FULL specifies that all nodes valid for cross-subnet searching are included in the subnetwork routing list for cross-subnetwork searching. BNDYN=NONE specifies that only nodes defined on the NEXTCP statement will be included in the subnetwork routing list for cross-subnetwork searching. Adjacent cluster definitions do not provide a method to selectively restrict searches to individual nodes during cross-subnetwork searching. Currently the user can only restrict cross-subnetwork searches by coding BNDYN=NONE and then only listing the border nodes that are to be searched. In networks with a large number of border nodes the user would like to code BNDYN=FULL, so that all possible border nodes are included for cross-subnet searching, then only list the small number of border nodes that are not to be searched. Another use for this option would be during planned outages to easily restrict searches to individual border nodes where a desirable path is not available.



Add OMITCP to the NEXTCP statement

- Add OMITCP operand to the NEXTCP statement of the adjacent cluster routing definitions

```
*********************************************************************
* Routing for NETID=NETZ with OMITTED nodes                        *
*********************************************************************
NETZ     NETWORK   NETID=(NETZ),                                     x
                   BNDYN=FULL,            allow full dynamics        x
                   BNORD=PRIORITY,        use priority routing       x
                   SNVC=4                 allow depth of 4 subnets
NODE2A   NEXTCP    CPNAME=NETA.NODE2A,                               x
                   OMITCP=YES             do not route to NODE2A
NODE2C   NEXTCP    CPNAME=NETC.NODE2C,                               x
                   OMITCP=YES             do not route to NODE2C
```

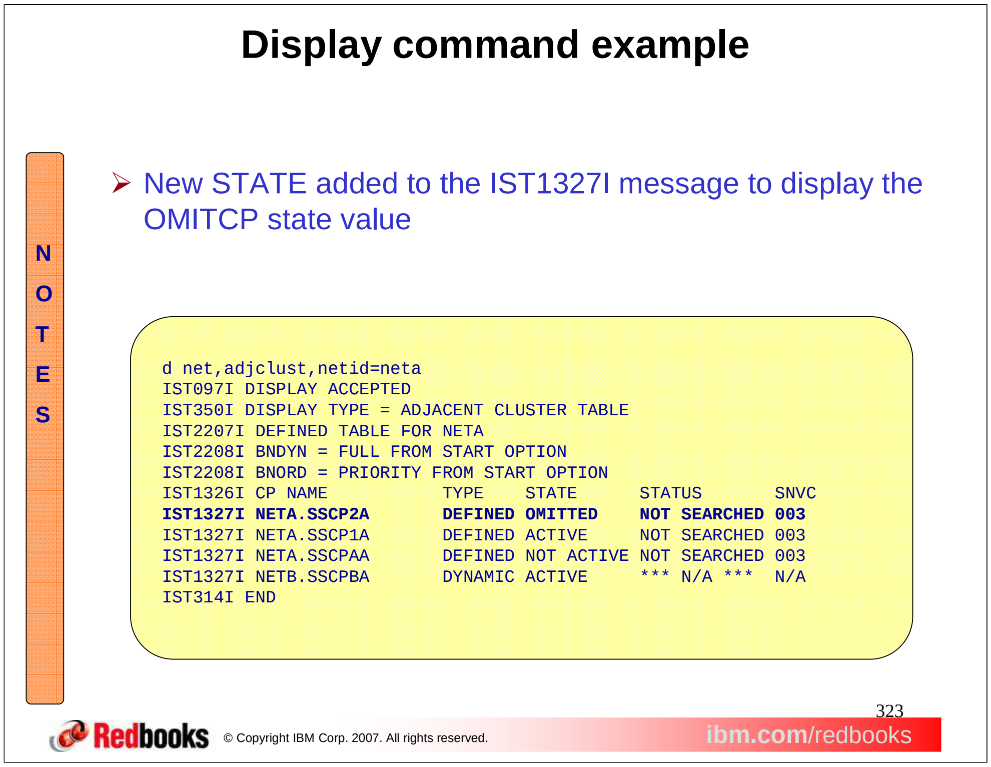
- OMITCP=YES prevents the CPNAME from being included in a subnet routing list for searching

322

Redbooks ibm.com/redbooks

The solution is to add a new OMITCP operand to the NEXTCP statement of the adjacent cluster routing definitions. The OMITCP operand will have a value of YES or NO with NO being the default. If OMITCP=YES is coded, the node specified on the CPNAME operand would not be included in a subnet routing list built for the NETID specified on NETWORK statement. The node would not be added as a defined or dynamic entry to the SRL. This will allow the user to selectively restrict searches to a specific node. The operand will work with all levels of dynamics. With BNDYN=NONE it is the same as not adding the node to the list of NEXTCPs.

The OMITCP operand could be used during planned outages to restrict searches to border nodes where a desirable path is not available. The border node selection function of the DSME allows more control cross-subnetwork searching. The DSME can customize cross-subnet searching based on search information other than the NETID, such as the OLU or DLU names.

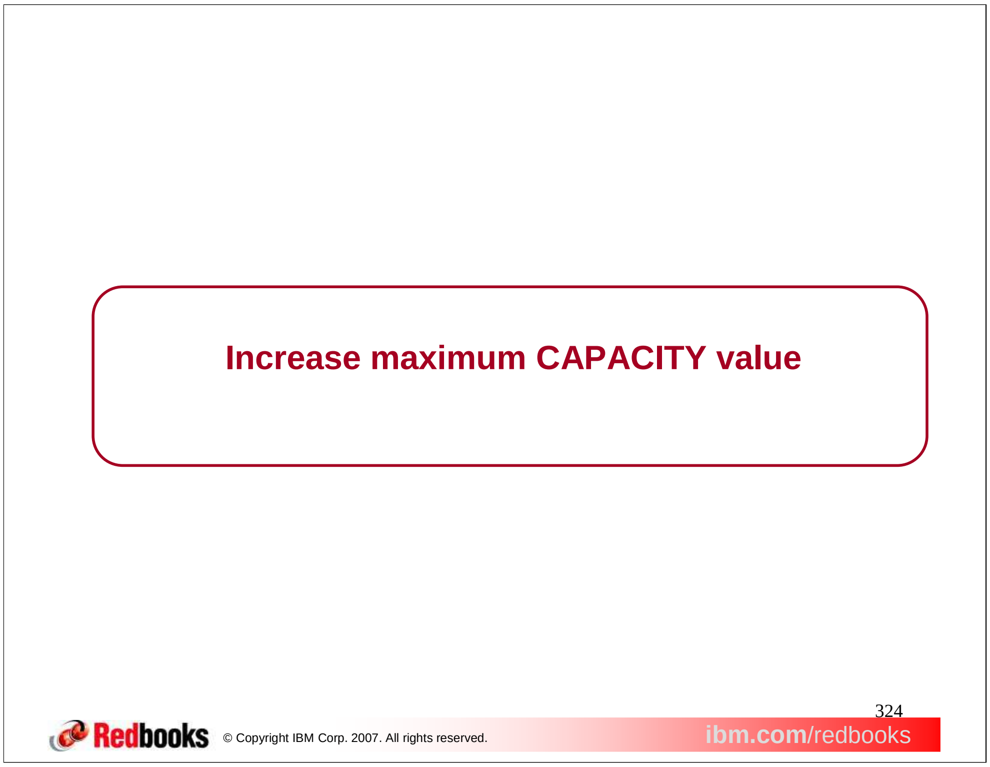


Display command example

- New STATE added to the IST1327I message to display the OMITCP state value

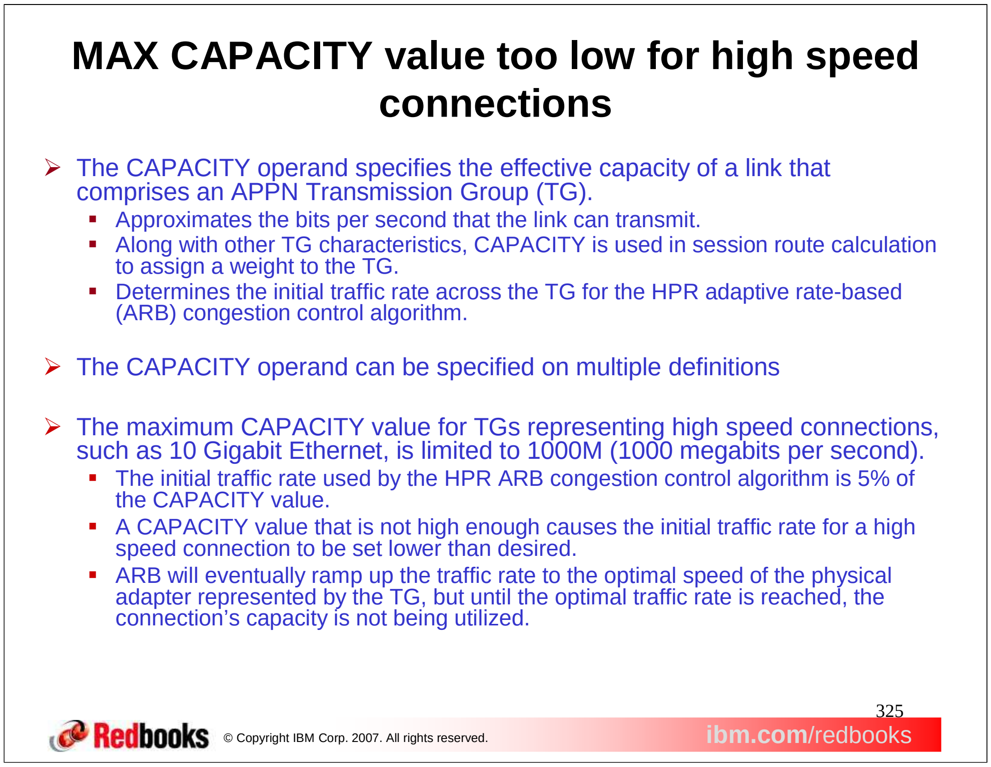
```
d net,adjclust,netid=neta
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE
IST2207I DEFINED TABLE FOR NETA
IST2208I BNDYN = FULL FROM START OPTION
IST2208I BNORD = PRIORITY FROM START OPTION
IST1326I CP NAME            TYPE     STATE      STATUS       SNVC
IST1327I NETA.SSCP2A        DEFINED OMITTED     NOT SEARCHED 003
IST1327I NETA.SSCP1A        DEFINED ACTIVE      NOT SEARCHED 003
IST1327I NETA.SSCPAA        DEFINED NOT ACTIVE NOT SEARCHED 003
IST1327I NETB.SSCPBA        DYNAMIC ACTIVE      *** N/A ***  N/A
IST314I END
```

To display the new OMITCP operand value, a new state variable was added for message IST1327I. If OMITCP=YES is coded the state of OMITTED is displayed. If OMITCP=YES is not coded the existing states of ACTIVE or NOT ACTIVE are displayed. This display was done from NETB.SSCPBA. SSCPBA was not defined in the adjacent cluster definitions but was added because border node dynamics is set to full.



Increase maximum CAPACITY value

The maximum limit of 1000M for CAPACITY has been increased in z/OS V1R9 Communications Server.



MAX CAPACITY value too low for high speed connections

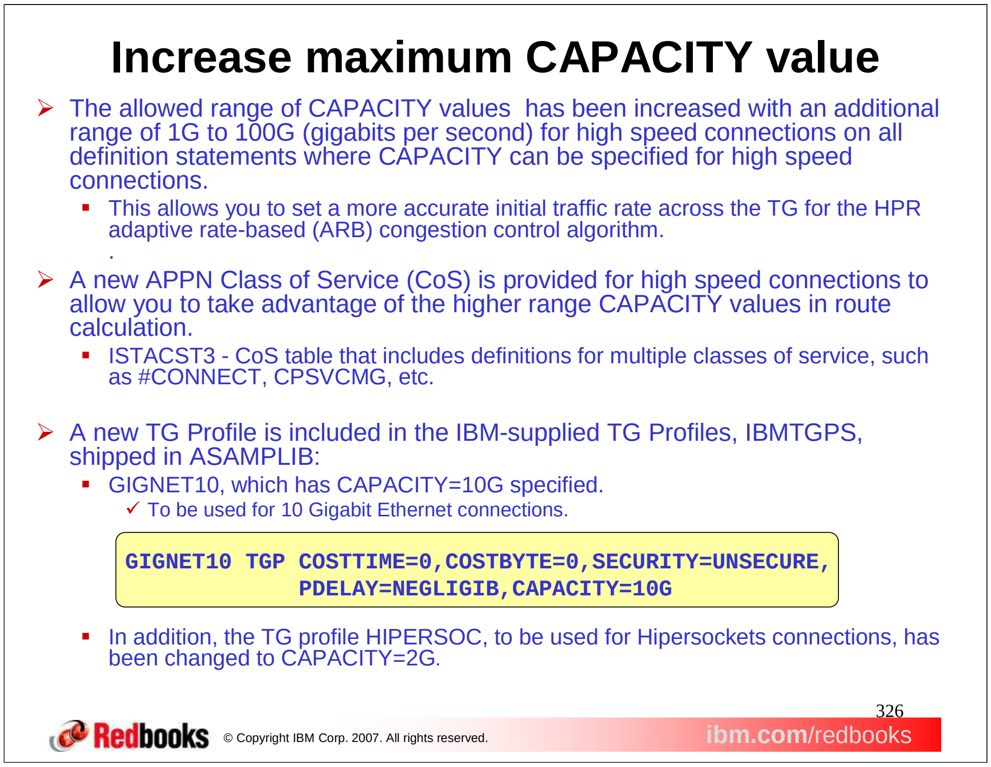
- The CAPACITY operand specifies the effective capacity of a link that comprises an APPN Transmission Group (TG).
  - Approximates the bits per second that the link can transmit.
  - Along with other TG characteristics, CAPACITY is used in session route calculation to assign a weight to the TG.
  - Determines the initial traffic rate across the TG for the HPR adaptive rate-based (ARB) congestion control algorithm.
- The CAPACITY operand can be specified on multiple definitions
- The maximum CAPACITY value for TGs representing high speed connections, such as 10 Gigabit Ethernet, is limited to 1000M (1000 megabits per second).
  - The initial traffic rate used by the HPR ARB congestion control algorithm is 5% of the CAPACITY value.
  - A CAPACITY value that is not high enough causes the initial traffic rate for a high speed connection to be set lower than desired.
  - ARB will eventually ramp up the traffic rate to the optimal speed of the physical adapter represented by the TG, but until the optimal traffic rate is reached, the connection’s capacity is not being utilized.

The CAPACITY operand can be specified in VTAM major nodes where the definition statement defines an APPN TG, as well as on APPN TG Profiles and APPN CoS definitions. The value specifies the effective capacity of a link, approximating the bits per second that the link can transmit. Along with other TG characteristics assigned to the TG, CAPACITY is used in session route calculation to assign a weight to the TG to determine the optimal route through an APPN network. CAPACITY is also used to determine the initial traffic rate across the TG for the HPR ARB congestion control algorithm.

The maximum limit of 1000M for the CAPACITY value of a TG representing a high speed connection is not high enough to set the optimal initial traffic rate for that connection.

The CAPACITY operand can be specified on the Cross Domain Resource Manager (CDRM) major node, the External Communications Adapter (XCA) major node, the Local SNA major node, the Model major node, the Network Control Program (NCP) major node, the Switched major node, the APPN TG Profile (TGP) definitions, and the APPN Class of Service (CoS) definitions.

The initial data rate across a TG for the HPR ARB congestion control algorithm is 5% of the CAPACITY value. This is the rate at which date is initially sent across the physical adapter represented by the TG. If the physical adapter can handle larger amounts of data, ARB ramps up the value gradually until it reaches the optimal traffic rate for the adapter. However, if the data rate is initially too low, the connection’s capacity is not being utilized until ARB increases the data rate to the connection’s optimal speed. Some physical connection types that can currently benefit from a higher CAPACITY are 10 Gigabit Ethernet, FICON Express, and Hipersockets. These connection types are only supported by Enterprise Extender.



Increase maximum CAPACITY value

- The allowed range of CAPACITY values has been increased with an additional range of 1G to 100G (gigabits per second) for high speed connections on all definition statements where CAPACITY can be specified for high speed connections.
  - This allows you to set a more accurate initial traffic rate across the TG for the HPR adaptive rate-based (ARB) congestion control algorithm.
    .
- A new APPN Class of Service (CoS) is provided for high speed connections to allow you to take advantage of the higher range CAPACITY values in route calculation.
  - ISTACST3 - CoS table that includes definitions for multiple classes of service, such as #CONNECT, CPSVCMG, etc.
- A new TG Profile is included in the IBM-supplied TG Profiles, IBMTGPS, shipped in ASAMPLIB:
  - GIGNET10, which has CAPACITY=10G specified.
    - ✓ To be used for 10 Gigabit Ethernet connections.

```
GIGNET10 TGP COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,
             PDELAY=NEGLIGIB,CAPACITY=10G
```

  - In addition, the TG profile HIPERSOC, to be used for Hipersockets connections, has been changed to CAPACITY=2G.

326

ibm.com/redbooks

An additional range of 1G to 100G is now allowed on the CAPACITY operand on all definition statements where CAPACITY can be specified. This range allows the user to specify a higher CAPACITY value for a TG than was previously available. The primary advantage of the higher CAPACITY value is to set a more accurate initial traffic rate across a high speed connection for the HPR adaptive rate-based (ARB) congestion control algorithm. ARB increases the traffic rate from the initial rate set by the CAPACITY value, so a higher initial traffic rate allows the algorithm to ramp up to an optimal traffic rate faster.

Also, for session route calculation, a new APPN Class of Service is provided to take advantage of the higher range CAPACITY values for high speed connections. ISTACST3, a new set of 12-row APPN CoS definitions will be shipped in z/OS V1R9 Communications Server in ASAMPLIB. This is a table that includes definitions for multiple classes of service, such as #CONNECT, CPSVCMG, etc. Unlike the other IBM-supplied Cos tables, COSAPPN and ISTACST2, these definitions utilize the new higher CAPACITY values on the LINEROW statements. These CoS definitions are designed to enable z/OS Communications Server to select an optimal route for a session when connections used in the network include those with high speed link characteristics. Some of these high speed connections are FICON, Gigabit Ethernet, and HiperSockets.

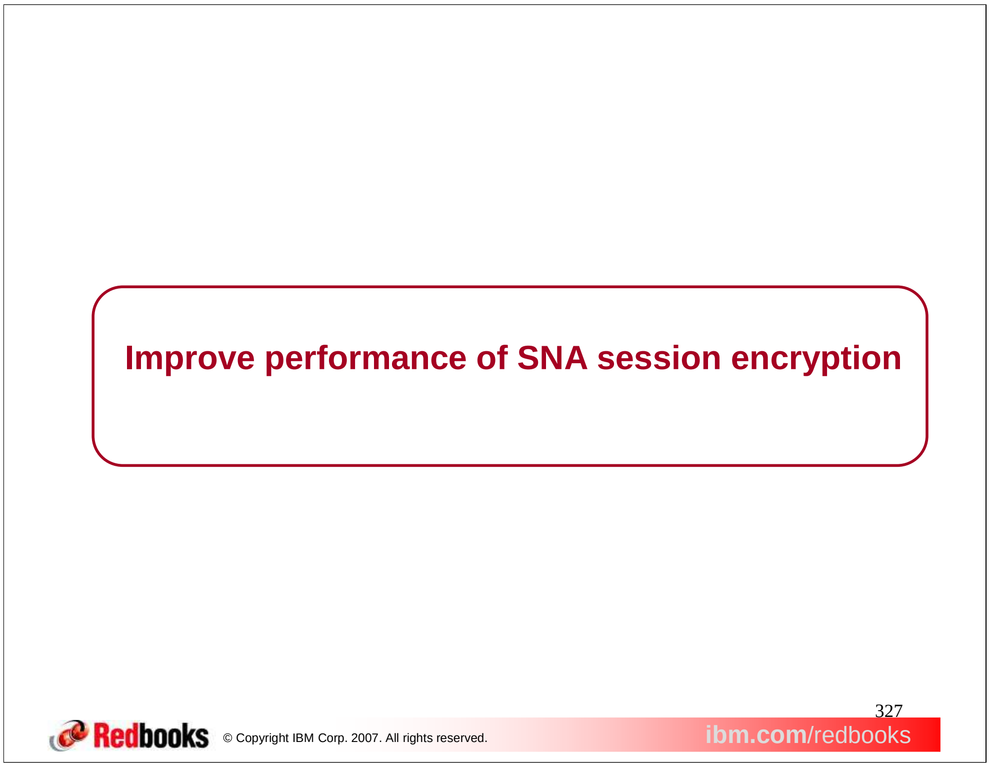
To use ISTACST3, you must copy the CoS definitions into SYS1.VTAMLST and then activate the member in which the definitions reside. You can have only one set of CoS definitions active at any time. COSAPPN is automatically activated when z/OS Communications Server is initialized. If you choose to use ISTACST3 you can activate it in one of the following ways: You can use the VARY ACT command to activate it, you can place the ISTACST3 member in the configuration list to automatically activate it at z/OS Communications Server initialization or you can rename the ISTACST3 member to COSAPPN and rename COSAPPN to something else. It will then be automatically activated at z/OS Communications Server initialization. When any Class of Service (CoS) is used in a network, whether COSAPPN, ISTACST2, or ISTACST3, it is important to use the same CoS on all network nodes in the network. If you do not use the same CoS, different session routes can be selected, depending on the TG characteristics specified in the different Classes of Service. This is true even across network boundaries when you are using border node configurations.

IBM-supplied TG Profiles are shipped in member IBMTGPS in ASAMPLIB. A new TGP for 10 Gigabit Ethernet connections, GIGNET10 is now included in IBMTGPS. This TGP sets the initial traffic rate across the TG for the HPR adaptive rate-based (ARB) congestion control algorithm to 5% of CAPACITY. GIGNET10 specifies CAPACITY=10G and will result in an initial data rate of 500M. In addition, the HIPERSOC TGP has been changed to CAPACITY=2G.

The IBMTGPS TG profiles are automatically activated at z/OS Communications Server initialization. All that is needed to activate them is to copy the definitions from ASAMPLIB into a member in VTAMLST. After IBMTGPS is activated, you can then assign the group of TG characteristics defined in a specific TG profile (for example GIGNET10) to a TG using one of the following methods:

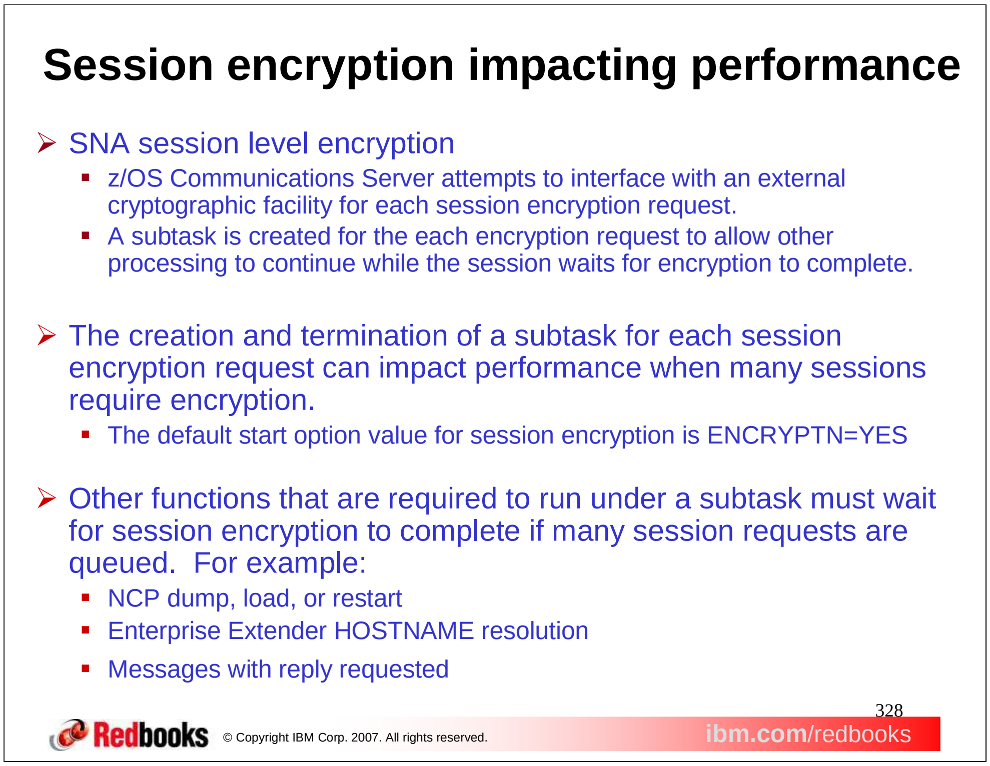
1. Specify the TGP=GIGNET10 operand on the PU definition statement
2. Assign the TGP to an already existing APPN TG with the MODIFY TGP,TGPNAME=GIGNET10,ID=adjacent_node,TGN=tg_number command.

The initial data rate across a TG for the HPR ARB congestion control algorithm is 5% of the CAPACITY value. This is the rate at which date is initially sent across the physical adapter represented by the TG. If the physical adapter can handle larger amounts of data, ARB ramps up the value gradually until it reaches the optimal traffic rate for the adapter. Therefore, CAPACITY values in the range of 1G-10G should be specified only for TGs across physical adapters that can handle the initial data rate. Assigning a CAPACITY value to a TG that is much higher than the physical adapter can handle can cause the adapter to be overrun. The data rate will eventually be reduced by ARB, but assigning an initial traffic rate that is too high can cause performance to suffer until an accurate traffic rate is reached. Some physical connection types that can currently benefit from a CAPACITY in the range of 1G-10G are 10 Gigabit Ethernet, FICON Express, and Hipersockets. These connection types are only supported by Enterprise Extender.



Improve performance of SNA session encryption

“Improve performance of SNA session encryption” is a functional enhancement introduced in z/OS V1R9 Communications Server.



Session encryption impacting performance

- SNA session level encryption
  - z/OS Communications Server attempts to interface with an external cryptographic facility for each session encryption request.
  - A subtask is created for the each encryption request to allow other processing to continue while the session waits for encryption to complete.
- The creation and termination of a subtask for each session encryption request can impact performance when many sessions require encryption.
  - The default start option value for session encryption is ENCRYPTN=YES
- Other functions that are required to run under a subtask must wait for session encryption to complete if many session requests are queued. For example:
  - NCP dump, load, or restart
  - Enterprise Extender HOSTNAME resolution
  - Messages with reply requested

Redbooks ibm.com/redbooks

SNA session level encryption requires a subtask be created for each session encryption request. This results in an attempt to interface with an external cryptographic facility.

The number of subtasks (TCB structures) allowed concurrently is controlled by the DLRTCB and MAXHNRES start options. These structures utilize below the line storage, so limits are necessary. For example, if DLRTCB=32 and MAXHNRES=20 are specified, then a total of 52 subtasks can be attached at one time. However, once requests are received by ISTINCDP, the subtasks attached for HOSTNAME resolution are not limited to the value specified for MAXHNRES, nor are the subtasks attached for other functions limited to the value specified for DLRTCB. MAXHNRES does limit the number of DISPLAY EE and DISPLAY EEDIAG commands requiring HOSTNAME resolution that VTAM will accept at once, however. This restriction is policed in the NOS component and is meant to prevent a user CLIST from overwhelming ISTINCDP.

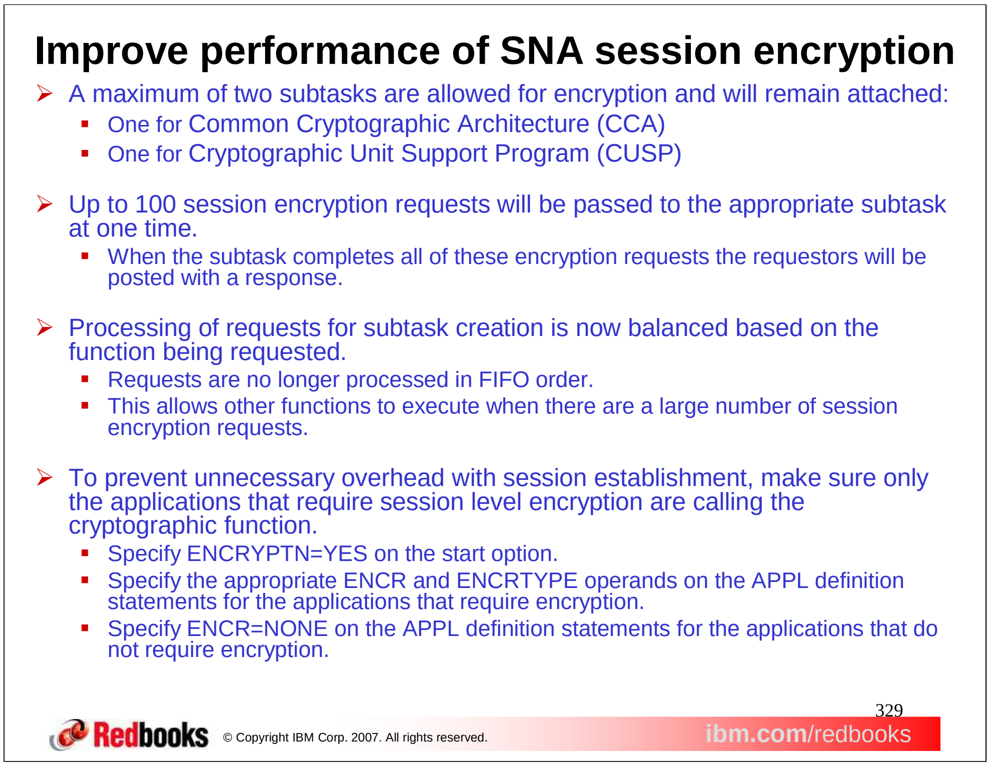
There are two symptoms of this problem that are commonly seen:

1. Thousands of sessions appear to be hung because they are queued up waiting for subtask resources needed for encryption requests. This often happens because the default for the ENCRYPTN start option is YES, causing ALL session requests to be sent to ISTINCDP even when encryption is not needed.

2. Because ISTINCDP processes requests in a first in first out order, functions other than encryption can't proceed if many encryption requests were queued first. There is a limit to the number of subtasks that can be attached concurrently. The functions, such as NCP loads, appear to be hung.

The overhead for creating and subsequently deleting the control block structure for each subtask can impact performance, causing many sessions to wait pending encryption. This impact can be severe, especially when the ENCRYPTN start option is allowed to default and ENCR=OPT is allowed to default on all APPL definitions. The default start option value of ENCRYPTN=YES causes ALL sessions with an application to request encryption processing, unless ENCR=NONE is specified on the APPL definition statement for that application.

Since there are a limited number of subtasks allowed to run concurrently (that number is the total of the values specified for the DLRTCB and MAXHNRES start options), many session encryption requests can cause other functions to wait until resources are available. This is because all requests for subtask creation are processed in the order that they are received.



Improve performance of SNA session encryption

- A maximum of two subtasks are allowed for encryption and will remain attached:
  - One for Common Cryptographic Architecture (CCA)
  - One for Cryptographic Unit Support Program (CUSP)
- Up to 100 session encryption requests will be passed to the appropriate subtask at one time.
  - When the subtask completes all of these encryption requests the requestors will be posted with a response.
- Processing of requests for subtask creation is now balanced based on the function being requested.
  - Requests are no longer processed in FIFO order.
  - This allows other functions to execute when there are a large number of session encryption requests.
- To prevent unnecessary overhead with session establishment, make sure only the applications that require session level encryption are calling the cryptographic function.
  - Specify ENCRYPTN=YES on the start option.
  - Specify the appropriate ENCR and ENCRTYPE operands on the APPL definition statements for the applications that require encryption.
  - Specify ENCR=NONE on the APPL definition statements for the applications that do not require encryption.

329

Redbooks ibm.com/redbooks

Instead of attaching and detaching a subtask for each session encryption request (represented by a DLRPL control block), one subtask will be created when the first CCA request is received and one subtask will be created when the first CUSP request is received.   Each of these subtasks will remain attached and waiting for work until VTAM termination. The subtask will interface with the appropriate cryptographic facility for each request. When all of the requests passed to the subtask (up to 100 at a time) have been completed, the DLRPL control blocks will be returned to the requestors so that session establishment can continue.

This should improve the performance of session level encryption because it eliminates the overhead of the ATTACH/DETACH processing for each encryption request and substitutes WAIT/POST processing, which has far less performance impact. Since only up to two subtasks are dedicated to encryption, an additional benefit is that it frees system resources for other functions supported by ISTINCDP.

Rather than creating subtasks in the order in which the request are received, requests for each functional area are load balanced to allow all functions a chance to used the limited resources needed to attach a subtask. This prevents these functions from having to wait for a large number of session encryption requests to be completed.

In addition, the concurrent number of subtasks allowed for HOSTNAME resolution is limited to the value specified for the MAXHNRES start option.

There are four functions that can result in HOSTNAME resolution: EE line activation with the HOSTNAME operand on the GROUP, Dialing an EE switched PU with HOSTNAME operand, DISPLAY EE command with HOSTNAME operand, and DISPLAY EEDIAG command with HOSTNAME operand.

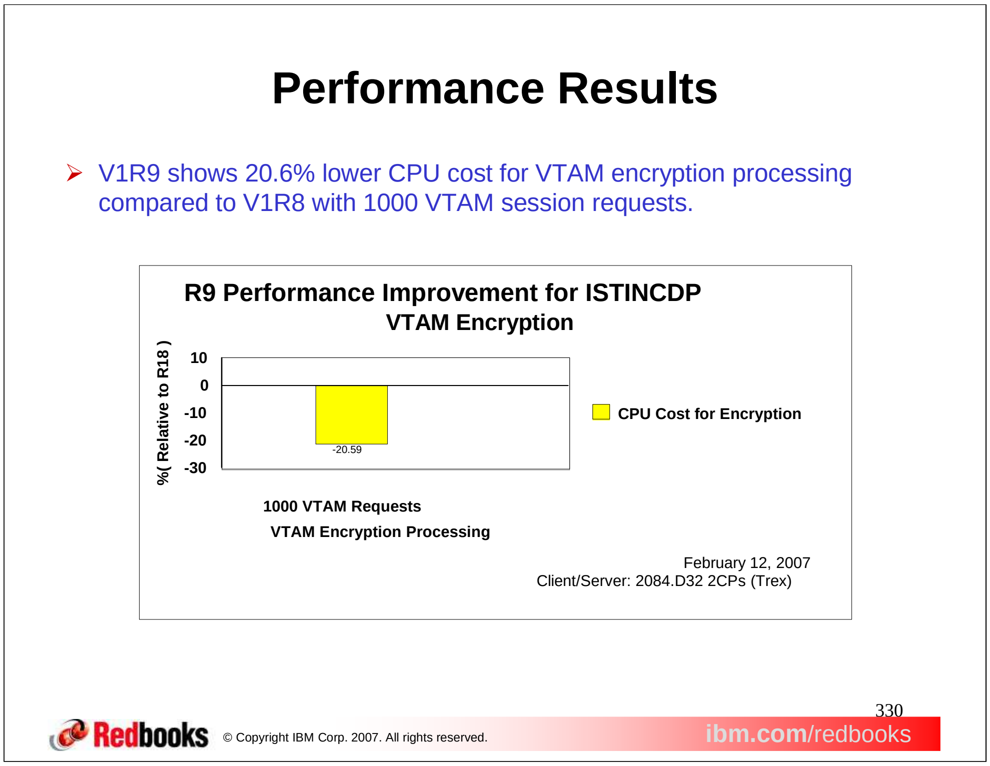
Prior to this, HOSTNAME resolution could utilize all subtask resources, including those reserved by the DLRTCB start option. MAXHNRES subtask resources concurrently being used for DISPLAY EE and DISPLAY EEDIAG command processing is further limited to 80% of MAXHNRES. This will allow EE line activation with HOSTNAME resolution to proceed even if a large number of the DISPLAY commands have been entered. However, Subtask requests for functions other than HOSTNAME resolution will be allowed to use more resources than those reserved by the DLRTCB start option if all of the resources reserved by the MAXHNRES start option are not in use.

Often only a few applications require session encryption. If the ENCRYPTN start option and the ENCR operand on all the APPL definition statements are allowed to default, requests for encryption will be made for all sessions, even when it is not necessary. If none of your applications require session level encryption, specify ENCRYPTN=NO on the start option. However, this cannot be overridden at the application level. If only a few applications require encryption, do the following:

Specify or allow the ENCRYPTN start option to default to YES. If you specify ENCRYPTN=NO on the start option, you cannot use session level encryption for any applications.

Specify the appropriate ENCR and ENCRTYPE operands on the APPL definition statements for only those applications that require session level encryption.

Specify ENCR=NONE on the APPL definition statements for all applications that do not require encryption.



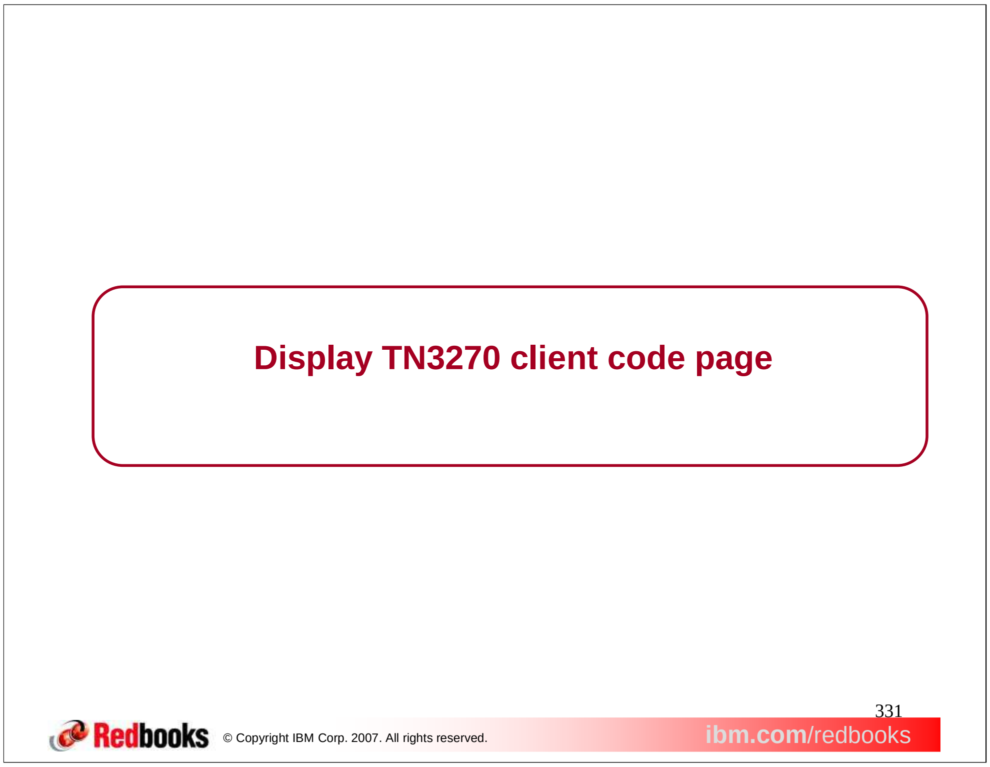
Performance Results

- V1R9 shows 20.6% lower CPU cost for VTAM encryption processing compared to V1R8 with 1000 VTAM session requests.

R9 Performance Improvement for ISTINCDP
VTAM Encryption

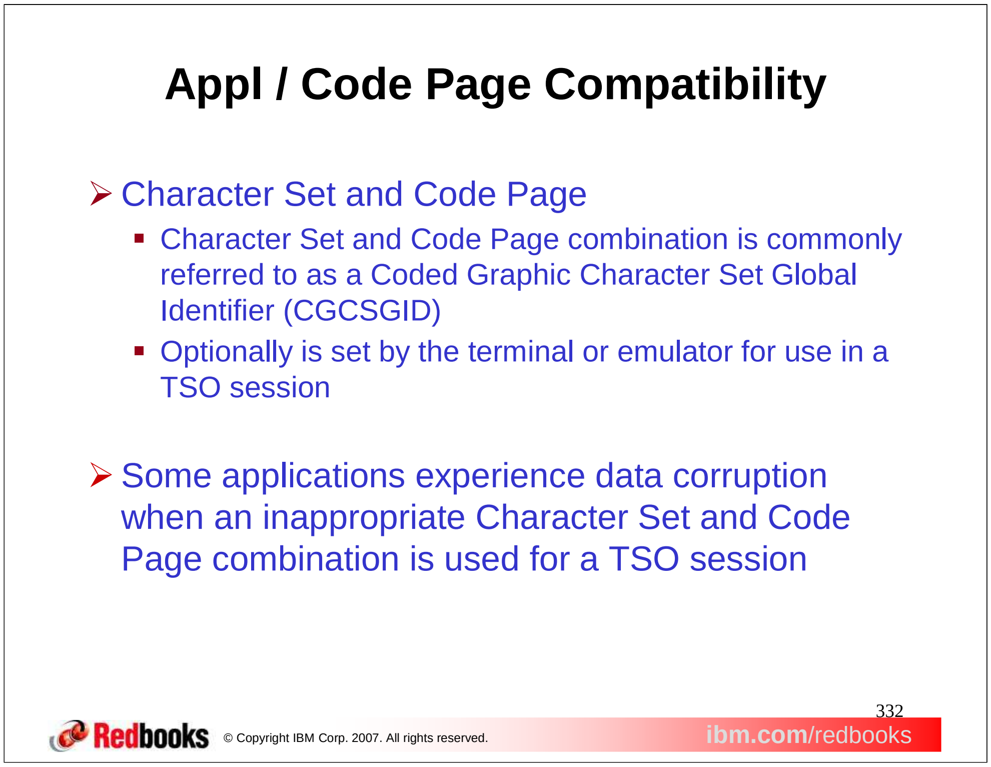
February 12, 2007
Client/Server: 2084.D32 2CPs (Trex)

In performance testing, CPU cost was lowered by 20.6% for VTAM encryption processing with 1000 session requests from z/OS Communications Server V1R8 to V1R9.



Display TN3270 client code page

This section describes the enhancement to display TN3270 Client Code Page information.



Appl / Code Page Compatibility

- Character Set and Code Page
  - Character Set and Code Page combination is commonly referred to as a Coded Graphic Character Set Global Identifier (CGCSGID)
  - Optionally is set by the terminal or emulator for use in a TSO session

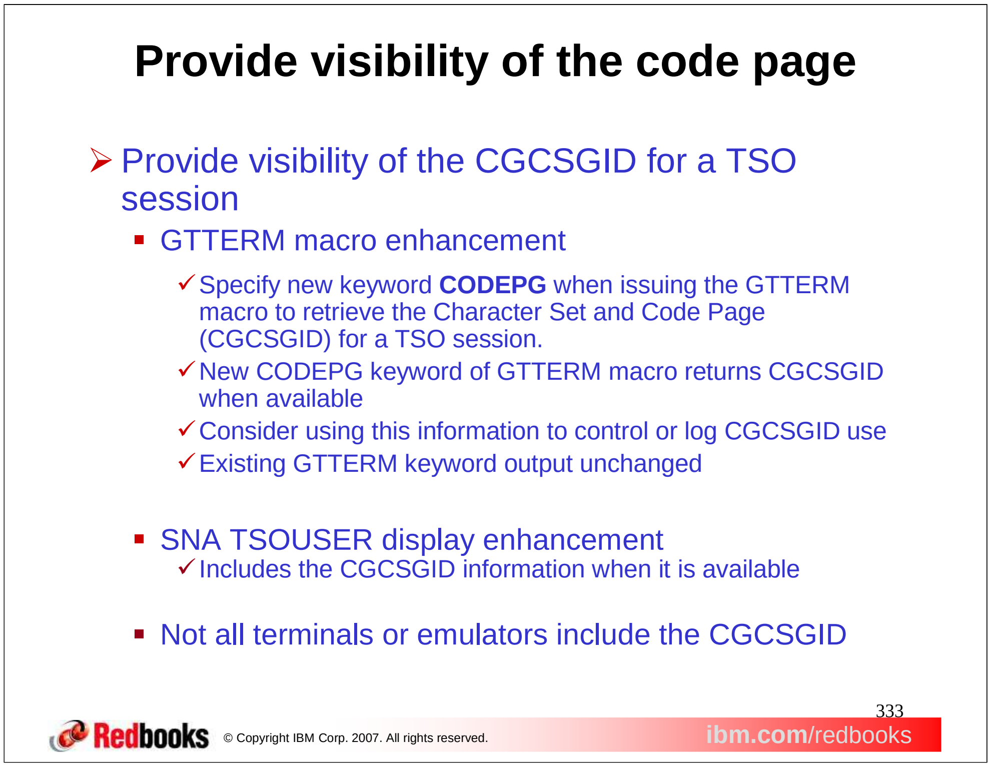
- Some applications experience data corruption when an inappropriate Character Set and Code Page combination is used for a TSO session

332

Redbooks ibm.com/redbooks

A Character Set and Code Page combination is commonly referred to as a Coded Graphic Character Set Global Identifier (CGCSGID) . The CGCSGID values are set by the terminal or emulator and used for a TSO session. However, not all terminals or emulators include the CGCSGID information.   Consult the documentation of the applicable terminal or emulator to see if CGCSGID information is supported.

The Character Set and Code Page combination in use for a TSO session may be inappropriate for some applications and cause data corruption.
This has been identified as a problem in some user DB2 environments where a TSO user with an incompatible client code page performs some processing that updates a DB2 database. As a result of incompatible code page, the data appears to be corrupted when stored back into DB2.



Provide visibility of the code page

- Provide visibility of the CGCSGID for a TSO session
  - GTTERM macro enhancement
    - ✓ Specify new keyword **CODEPG** when issuing the GTTERM macro to retrieve the Character Set and Code Page (CGCSGID) for a TSO session.
    - ✓ New CODEPG keyword of GTTERM macro returns CGCSGID when available
    - ✓ Consider using this information to control or log CGCSGID use
    - ✓ Existing GTTERM keyword output unchanged
  - SNA TSOUSER display enhancement
    - ✓ Includes the CGCSGID information when it is available
  - Not all terminals or emulators include the CGCSGID

333

Redbooks ibm.com/redbooks

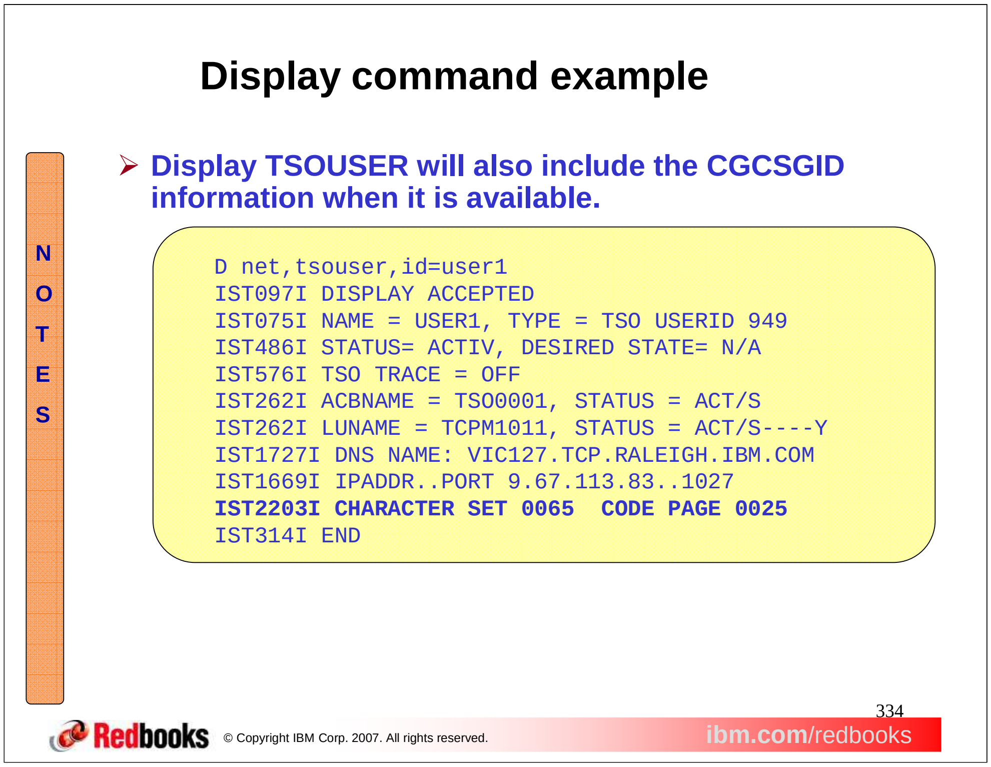
This new function provides visibility of the CGCSGID for a TSO session.   Users may want to use this information as a criteria to permit or deny access to an application through use of a logon exit.

TSO/VTAM supports a GTTERM macro that the user can use to acquire information about a terminal. A new keyword, CODEPG, has been added to the GTTERM processing to allow the user to retrieve the CGCSGID information for a TSO session.

When the GTTERM macro is issued with the CODEPG keyword, the following information is returned to the issuer: Terminal name, Network ID, IP address, Port number, Character Set, if available, and Code Page, if available.   Note that the Character Set and Code Page (CGCSGID) information is only available when the terminal or emulator includes it when the session is established. Consider updating your logon exit to log via a message what Code Page a client is using or enforce a certain set of Code Pages that users can use for a specific application. The existing GTTERM keyword output is unchanged. See the TSO/E Programming Services publication for information on the GTTERM macro.

The SNA TSOUSER display has been enhanced to report the CGCSGID in use for a TSO user. The TSOUSER Display will show Code Page and Character Set information when it is available.

Not all terminals or emulators include the CGCSGID information. Consult the applicable terminal or emulator documentation to see if CGCSGID information is supported.



Display command example

- Display TSOUSER will also include the CGCSGID information when it is available.

```
D net,tsouser,id=user1
IST097I DISPLAY ACCEPTED
IST075I NAME = USER1, TYPE = TSO USERID 949
IST486I STATUS= ACTIV, DESIRED STATE= N/A
IST576I TSO TRACE = OFF
IST262I ACBNAME = TSO0001, STATUS = ACT/S
IST262I LUNAME = TCPM1011, STATUS = ACT/S----Y
IST1727I DNS NAME: VIC127.TCP.RALEIGH.IBM.COM
IST1669I IPADDR..PORT 9.67.113.83..1027
IST2203I CHARACTER SET 0065  CODE PAGE 0025
IST314I END
```

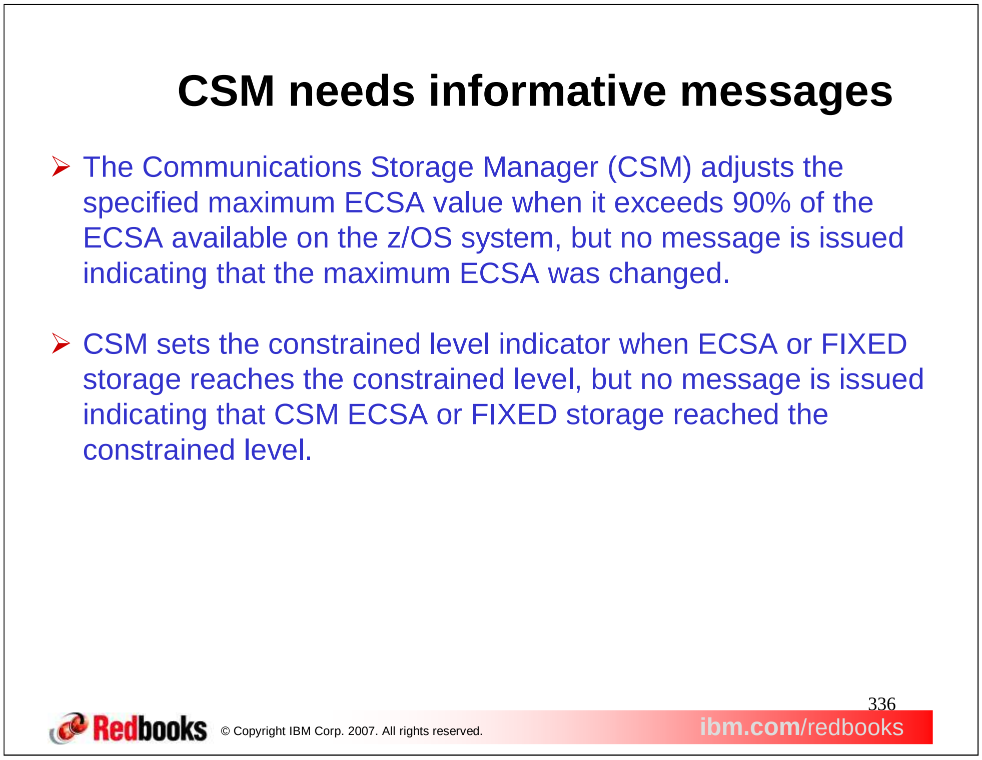
The output of the Display TSOUSER command will also include the CGCSGID information when it is available.

CSM Enhancements

335

ibm.com/redbooks

This section describes the enhancement made to CSM for z/OS V1R9.



CSM needs informative messages

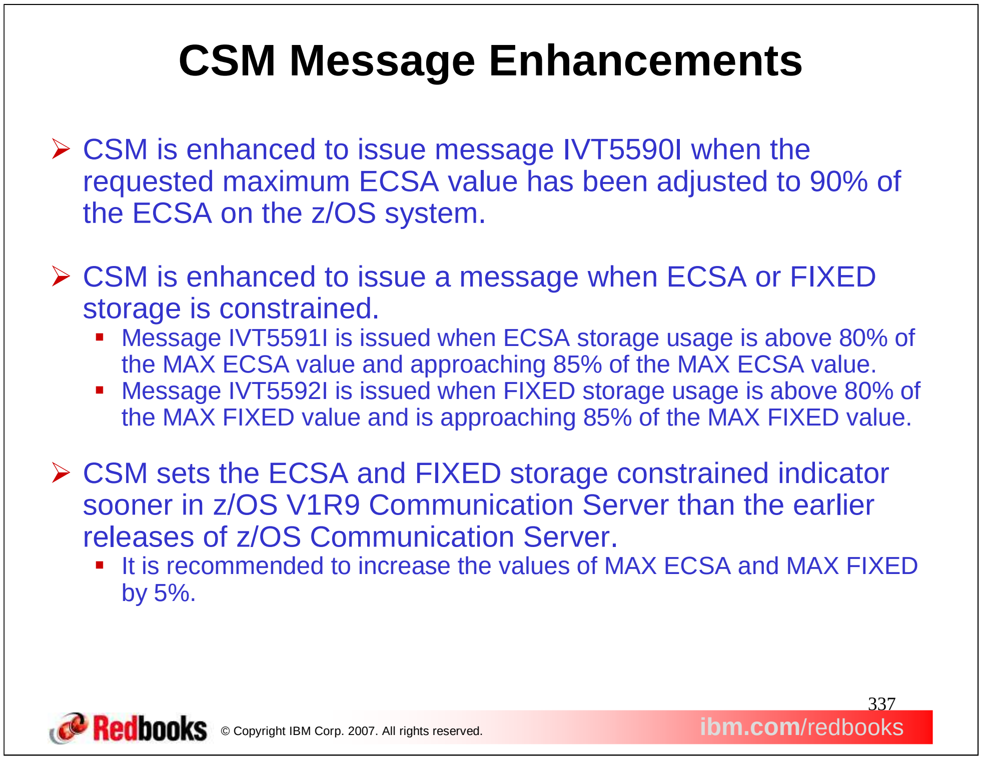
- The Communications Storage Manager (CSM) adjusts the specified maximum ECSA value when it exceeds 90% of the ECSA available on the z/OS system, but no message is issued indicating that the maximum ECSA was changed.
- CSM sets the constrained level indicator when ECSA or FIXED storage reaches the constrained level, but no message is issued indicating that CSM ECSA or FIXED storage reached the constrained level.

336

ibm.com/redbooks

CSM needs to issue a message when it adjusts MAX ECSA value. It will clarify the new value to the user.

CSM needs to issue a message when it sets the constrained level indicator for CSM ECSA or fixed storage. This will allow the operator to take some actions to relieve the situation.



CSM Message Enhancements

- CSM is enhanced to issue message IVT5590I when the requested maximum ECSA value has been adjusted to 90% of the ECSA on the z/OS system.
- CSM is enhanced to issue a message when ECSA or FIXED storage is constrained.
  - Message IVT5591I is issued when ECSA storage usage is above 80% of the MAX ECSA value and approaching 85% of the MAX ECSA value.
  - Message IVT5592I is issued when FIXED storage usage is above 80% of the MAX FIXED value and is approaching 85% of the MAX FIXED value.
- CSM sets the ECSA and FIXED storage constrained indicator sooner in z/OS V1R9 Communication Server than the earlier releases of z/OS Communication Server.
  - It is recommended to increase the values of MAX ECSA and MAX FIXED by 5%.

337

Redbooks ibm.com/redbooks

Message IVT5590I can be issued during: CSM initialization when the ECSA MAX value specified on the IVTPRM00 parmlib member is larger than 90% of the ECSA on the system. DISPLAY CSM command processing when the maximum ECSA value in effect has been adjusted by CSM. And MODIFY CSM command processing when the maximum ECSA requested is larger than 90% of the ECSA on the system.

CSM changed the definition of the ECSA and FIXED storage constrained level. CSM now sets the ECSA storage at the constrained level When ECSA storage usage is above 80% of the MAX ECSA value and approaching 85% of the MAX ECSA value. CSM sets the fixed storage at the constrained level When fixed storage usage is above 80% of the MAX FIXED value and approaching 85% of the MAX FIXED value.

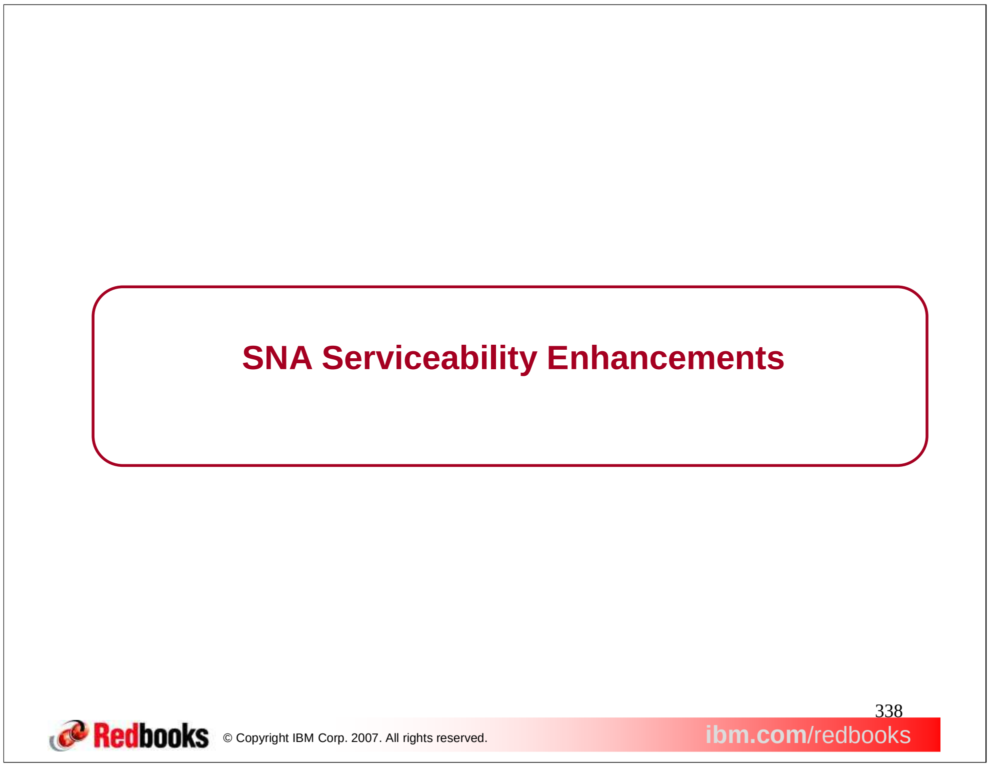
CSM changed the definition of the ECSA and FIXED storage normal level. CSM sets the ECSA storage at the normal level when ECSA storage usage goes below 80% of the MAX ECSA value. CSM sets the fixed storage at the normal level when fixed storage usage goes below 80% of the MAX FIXED value.

CSM issues the message IVT5564I ECSA storage shortage relieved when the current ECSA storage usage goes below 80% of the MAX ECSA value. CSM issues the message IVT5565I fixed storage shortage relieved when the current fixed storage usage goes below 80% of the MAX FIXED value.

The CSM Monitor function is available to monitor CSM buffers between many components of z/OS for Communication Server. This function can be controlled using the Modify CSM command with the MONITOR operand. The valid options are MONITOR=ON, MONITOR=OFF and MONITOR=DYNAMIC. The default value of the MONITOR option is DYNAMIC. If the user choose the option MONITOR=DYNAMIC, CSM Buffer Monitoring will be dynamically activated and inactivated.

In prior releases, CSM activated dynamically CSM buffer Monitoring when CSM storage usage reached 85% or higher of the MAX ECSA value or the current fixed storage usage reached 85% or higher of the MAX FIXED value. CSM inactivated the Dynamic CSM Monitor function when the current ECSA storage usage went below 80% of the MAX ECSA value and the current fixed storage usage went below 80% of the MAX FIXED value.

In z/OS V1R9, the threshold for activating the Dynamic CSM Monitor function is when the storage usage is 80% or higher. The threshold for inactivating the Dynamic CSM Monitor function is when the storage usage goes below 75%.

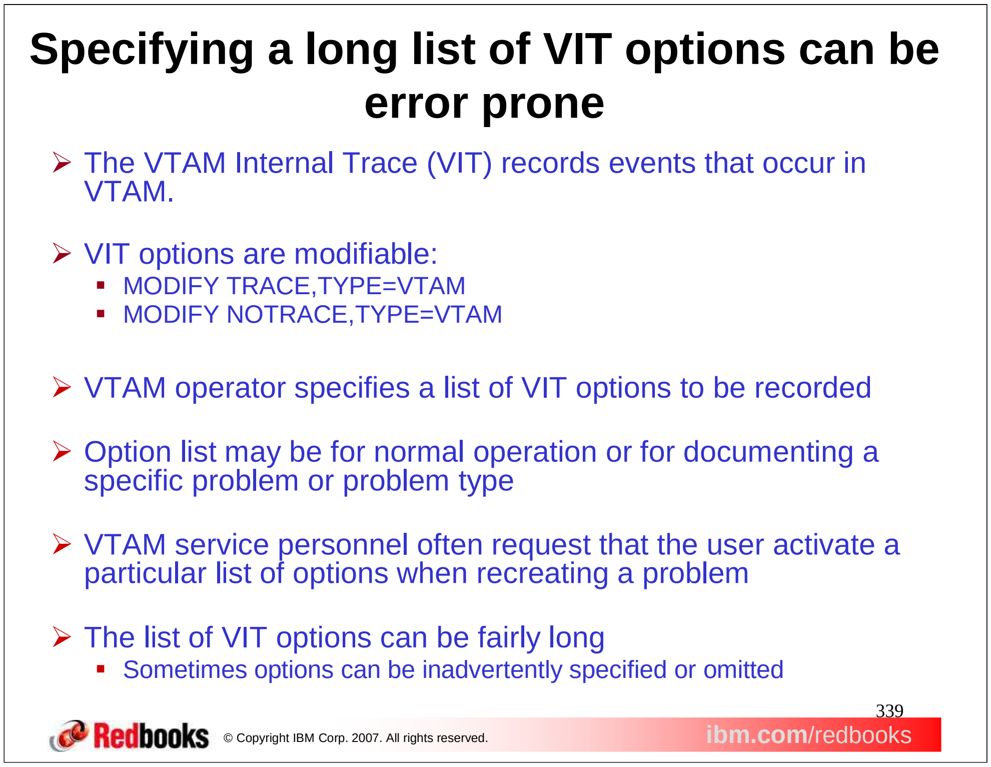


SNA Serviceability Enhancements

338

ibm.com/redbooks

This section describes some serviceability enhancements made for VTAM in z/OS Communications Server for V1R9.



Specifying a long list of VIT options can be error prone

- The VTAM Internal Trace (VIT) records events that occur in VTAM.
- VIT options are modifiable:
  - MODIFY TRACE,TYPE=VTAM
  - MODIFY NOTRACE,TYPE=VTAM
- VTAM operator specifies a list of VIT options to be recorded
- Option list may be for normal operation or for documenting a specific problem or problem type
- VTAM service personnel often request that the user activate a particular list of options when recreating a problem
- The list of VIT options can be fairly long
  - Sometimes options can be inadvertently specified or omitted

Redbooks

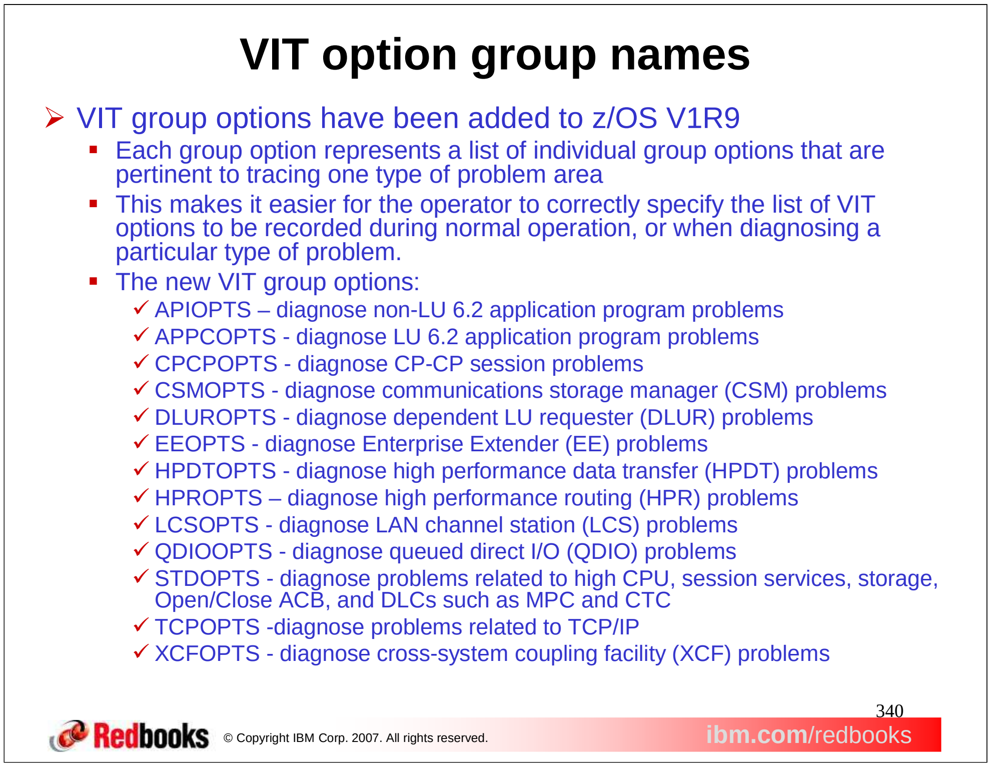
There are two wraparound tables in storage for internal VIT recording: The ECSA table is from 100 to 999 pages in size and is used to record the most recent events. The optional data space table (in ‘net’.ISTITDS1) is from 10 to 50 megabytes in size. Entries are copied to it from the ECSA table periodically to preserve older event records.

The external trace can be much larger and therefore is recommended for documenting problems where a substantial amount of history is needed. GTF must be active for VTAM external tracing. External VIT tracing will only occur if explicitly requested. No VIT options are traced by default fore external tracing. Any combination of VIT options can be turned on or off.

On the other hand, VTAM always records certain VIT entries to an internal trace table. The user can expand this trace table, use the optional data space table, and specify that many more options be traced. But the user cannot completely turn off internal tracing.

Events traced fall into categories called VIT options. Each option is comprised of one or more individual VIT entry types. The SNA Resource Definition Reference describes how to code the TRACE,TYPE=VTAM start option. The VIT options and entries are described in detail in SNA Diagnosis Volume 2. SNA Operation describes how to modify the VIT options by turning individual options on or off. SNA Diagnosis Volume 1 has a detailed treatment of VIT option modification.

Specifying a long list of VIT options can be error prone. It may be difficult to remember the right VIT options to specify to document a particular type of problem. Forgetting an option might lead to another recreate request! So the choice for the user was to specify ALL to be sure everything needed was traced, or to ask for, look up, divine, or remember the best list of VIT options for the situation at hand. However, specifying ALL when not required fills the internal and external VIT tables quicker, making lost entries due to wrapping more likely.



VIT option group names

- VIT group options have been added to z/OS V1R9
  - Each group option represents a list of individual group options that are pertinent to tracing one type of problem area
  - This makes it easier for the operator to correctly specify the list of VIT options to be recorded during normal operation, or when diagnosing a particular type of problem.
  - The new VIT group options:
    - ✓ APIOPTS – diagnose non-LU 6.2 application program problems
    - ✓ APPCOPTS - diagnose LU 6.2 application program problems
    - ✓ CPCPOPTS - diagnose CP-CP session problems
    - ✓ CSMOPTS - diagnose communications storage manager (CSM) problems
    - ✓ DLUROPTS - diagnose dependent LU requester (DLUR) problems
    - ✓ EEOPTS - diagnose Enterprise Extender (EE) problems
    - ✓ HPDTOPTS - diagnose high performance data transfer (HPDT) problems
    - ✓ HPROPTS – diagnose high performance routing (HPR) problems
    - ✓ LCSOPTS - diagnose LAN channel station (LCS) problems
    - ✓ QDIOOPTS - diagnose queued direct I/O (QDIO) problems
    - ✓ STDOPTS - diagnose problems related to high CPU, session services, storage, Open/Close ACB, and DLCs such as MPC and CTC
    - ✓ TCPOPTS -diagnose problems related to TCP/IP
    - ✓ XCFOPTS - diagnose cross-system coupling facility (XCF) problems

340

Redbooks ibm.com/redbooks

Prior to z/OS V1R9, the only group option available was ALL. It could be used to turn all of the VIT options on or off. z/OS V1R9 provides 13 new VIT group options that will make it easier to get exactly the right set of VIT options activated. The name of each group option is intended to convey its meaning. Each option is applicable to tracing a particular type of problem.

The MODIFY TRACE command will add the OPTIONs specified to the currently active list of options for the specified MODE (internal or external). It doesn't replace the currently active list of options with the ones specified. The MODIFY NOTRACE command will subtract the OPTIONs specified from the currently active list of options for the specified MODE (internal or external). Just as multiple options can be specified on for TRACE,TYPE=VTAM, multiple group options can be specified, even though they overlap. And a mixture of group options and individual options can be specified as well. VTAM will sort it out!

Two of the VIT options, HPR and SSCP, have associated subtrace options. The subtrace options are inactive by default. The HPR option has an ARBP subtrace option. The SSCP option has two subtrace options: TGVC and TREE.

Subtrace options can be turned on or off with a MODIFY TRACE or MODIFY NOTRACE command, respectively. The associated VIT option must be included in the command for this to be accepted. For example:

•F net,TRACE,TYPE=VTAM,OPTION=SSCP,SUBTRACE=TGVC is valid.

•F net,TRACE,TYPE=VTAM,OPTION=CIO,SUBTRACE=TGVC is not valid.

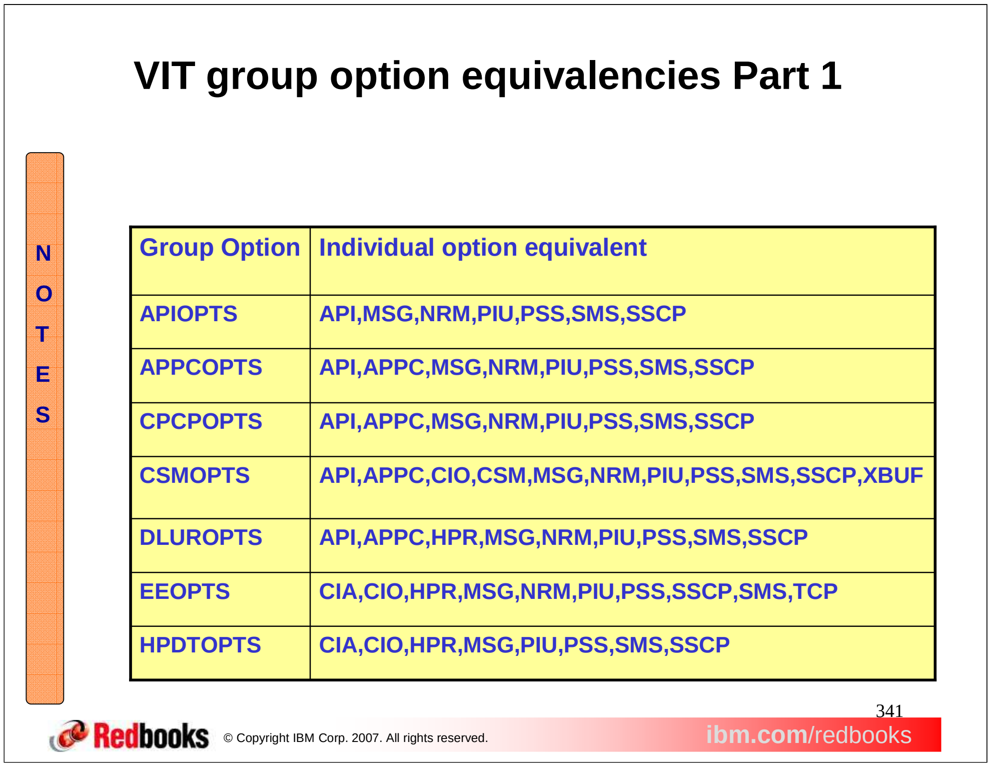
With this new function, any VTAM group option containing HPR as a component option can be used to activate or inactivate HPR subtrace option ARBP.

For example, F net,TRACE,TYPE=HPROPTS,SUBTRACE=ARBP will activate HPR subtrace ARBP in addition to the HPR option and the other component options of HPROPTS.

And F net,NOTRACE,TYPE=QDIOOPTS,SUBTRACE=ARBP will inactivate subtrace option ARBP and all component options of QDIOOPTS **except for HPR**! That is because F net,NOTRACE,TYPE=HPR,SUBTRACE=ARBP inactivates subtrace option ARBP but not option HPR.

All the group options contain SSCP as a component option, so any group option can be used to activate or inactivate SSCP subtraces TGVC and TREE. But such an inactivation will leave the SSCP option itself active.

It's simpler, and recommended, to use the appropriate individual VIT option to turn off subtraces.



VIT group option equivalencies Part 1

Group Option	Individual option equivalent
APIOPTS	API,MSG,NRM,PIU,PSS,SMS,SSCP
APPCOPTS	API,APPC,MSG,NRM,PIU,PSS,SMS,SSCP
CPCPOPTS	API,APPC,MSG,NRM,PIU,PSS,SMS,SSCP
CSMOPTS	API,APPC,CIO,CSM,MSG,NRM,PIU,PSS,SMS,SSCP,XBUF
DLUROPTS	API,APPC,HPR,MSG,NRM,PIU,PSS,SMS,SSCP
EEOPTS	CIA,CIO,HPR,MSG,NRM,PIU,PSS,SSCP,SMS,TCP
HPDTOPTS	CIA,CIO,HPR,MSG,PIU,PSS,SMS,SSCP

Specifying **APIOPTS** is equivalent to specifying all of the following VIT options: API, MSG, NRM, PIU, PSS, SMS and SSCP.

Specifying **APPCOPTS** is equivalent to specifying all of the following VIT options: API, APPC, MSG, NRM, PIU, PSS, SMS and SSCP.

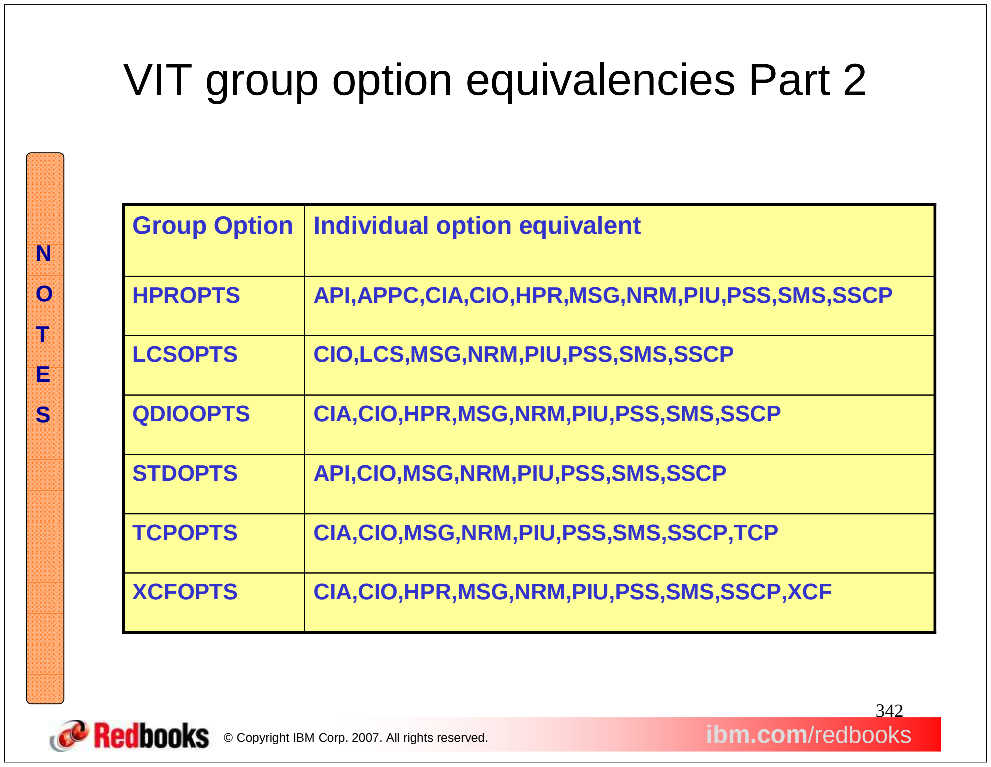
Specifying **CPCPOPTS** is equivalent to specifying all of the following VIT options: API, APPC, MSG, NRM, PIU, PSS, SMS and SSCP.

Specifying **CSMOPTS** is equivalent to specifying all of the following VIT options: API, APPC, CIO, CSM, MSG, NRM, PIU, PSS, SMS, SSCP and XBUF.

Specifying **DLUROPTS** is equivalent to specifying all of the following VIT options: API, APPC, HPR, MSG, NRM, PIU, PSS, SMS and SSCP.

Specifying **EEOPTS** is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP and TCP.

Specifying **HPDTOPTS** is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, PIU, PSS, SMS and SSCP.



VIT group option equivalencies Part 2

Group Option	Individual option equivalent
HPROPTS	API,APPC,CIA,CIO,HPR,MSG,NRM,PIU,PSS,SMS,SSCP
LCSOPTS	CIO,LCS,MSG,NRM,PIU,PSS,SMS,SSCP
QDIOOPTS	CIA,CIO,HPR,MSG,NRM,PIU,PSS,SMS,SSCP
STDOPTS	API,CIO,MSG,NRM,PIU,PSS,SMS,SSCP
TCPOPTS	CIA,CIO,MSG,NRM,PIU,PSS,SMS,SSCP,TCP
XCFOPTS	CIA,CIO,HPR,MSG,NRM,PIU,PSS,SMS,SSCP,XCF

Specifying **HPROPTS** is equivalent to specifying all of the following VIT options: API, APPC, CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS and SSCP.

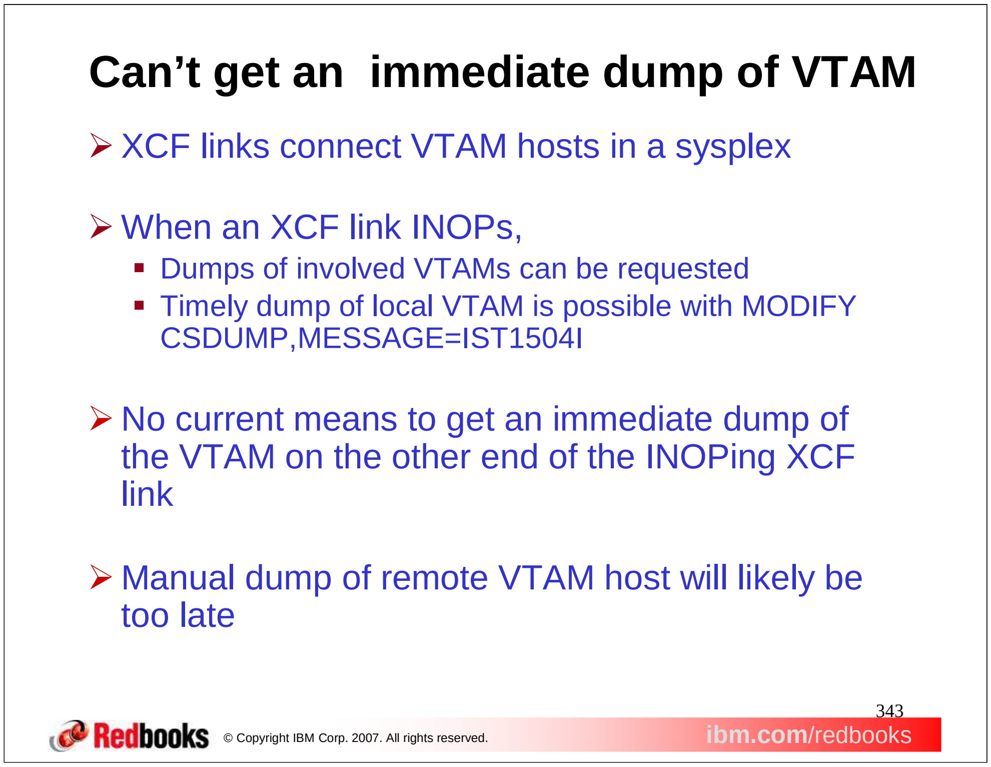
Specifying **LCSOPTS** is equivalent to specifying all of the following VIT options: CIO, LCS, MSG, NRM, PIU, PSS, SMS and SSCP.

Specifying **QDIOOPTS** is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS and SSCP.

Specifying **STDOPTS** is equivalent to specifying all of the following VIT options: API, CIO, MSG, NRM, PIU, PSS, SMS and SSCP. These are the options traced internally by default.

Specifying **TCPOPTS** is equivalent to specifying all of the following VIT options: CIA, CIO, MSG, NRM, PIU, PSS, SMS, SSCP, and TCP.

Specifying **XCFOPTS** is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP and XCF.



Can’t get an immediate dump of VTAM

- XCF links connect VTAM hosts in a sysplex
- When an XCF link INOPs,
  - Dumps of involved VTAMs can be requested
  - Timely dump of local VTAM is possible with MODIFY CSDUMP,MESSAGE=IST1504I
- No current means to get an immediate dump of the VTAM on the other end of the INOPing XCF link
- Manual dump of remote VTAM host will likely be too late

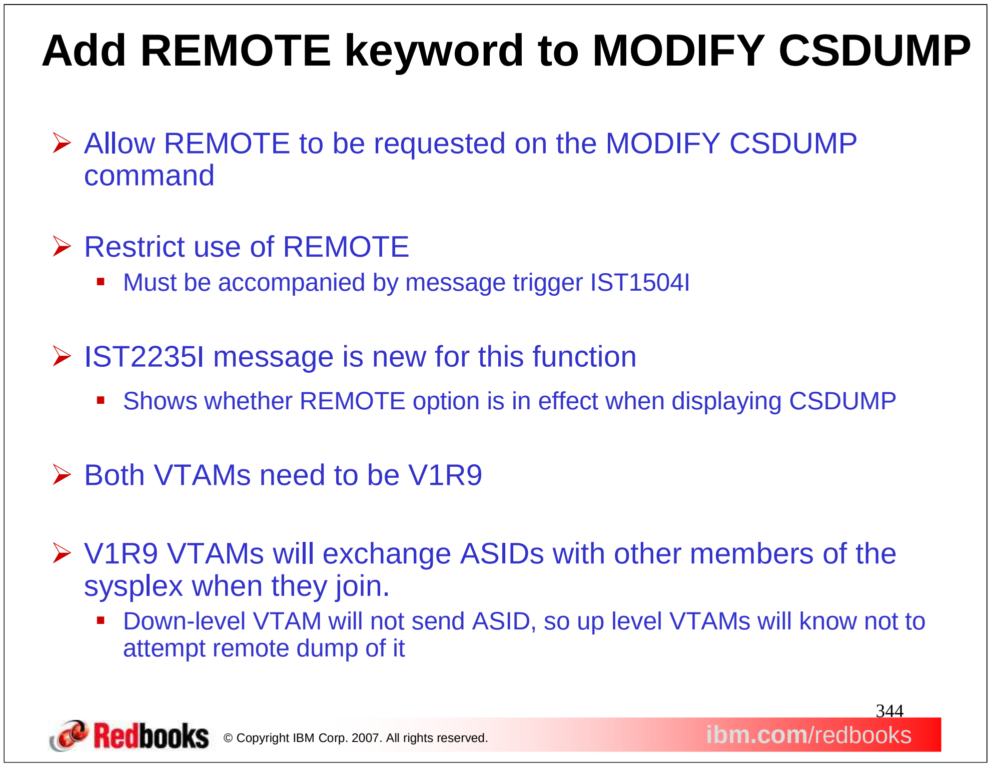
343

Redbooks

ibm.com/redbooks

XCF links are used to connect VTAM hosts in a sysplex. When an XCF link INOPs, existing VTAM facilities can be used to obtain a timely dump of the local VTAM for problem diagnosis. The operator can do this by setting a trigger on message IST1504I, which is only issued at the time of an XCF link INOP.

However, there is no current means to get an immediate dump of the VTAM on the other end of the XCF link. By the time a dump of the other VTAM is requested by the operator, it may be far too late to determine anything useful from it.



Add REMOTE keyword to MODIFY CSDUMP

- Allow REMOTE to be requested on the MODIFY CSDUMP command
- Restrict use of REMOTE
  - Must be accompanied by message trigger IST1504I
- IST2235I message is new for this function
  - Shows whether REMOTE option is in effect when displaying CSDUMP
- Both VTAMs need to be V1R9
- V1R9 VTAMs will exchange ASIDs with other members of the sysplex when they join.
  - Down-level VTAM will not send ASID, so up level VTAMs will know not to attempt remote dump of it

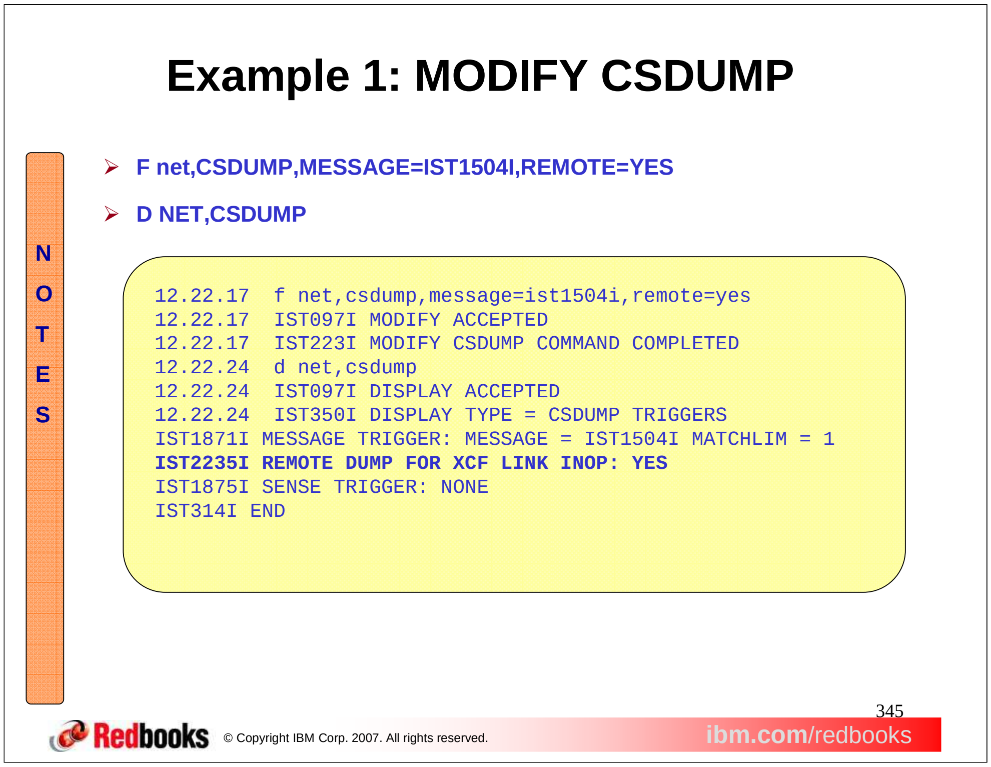
344

For MODIFY CSDUMP, VTAM issues an SDUMPX request to the system. If VTAM is connected to other VTAMs in a sysplex, the SDUMPX request by VTAM can include the existing REMOTE parameter to dump another VTAM in the sysplex. VTAM will only attempt this when message IST1504I is issued, and only when the operator has specifically requested it using the new REMOTE parameter on MODIFY CSDUMP.

New message IST2235I will show whether REMOTE is in effect for CSDUMP. It is added to message group IST1871I and only displayed if MESSAGE=IST1504I trigger is set.

On the SDUMPX request, VTAM needs to specify the ASID of the VTAM on the remote host. Otherwise, VTAM can't be dumped in the remote host. If REMOTE is active for CSDUMP and an XCF link INOPs, VTAM will check for the ASID of the partner VTAM. If it was not received, no remote dump attempt will be made.

It should only be necessary to set the REMOTE parameter on in one system per sysplex.



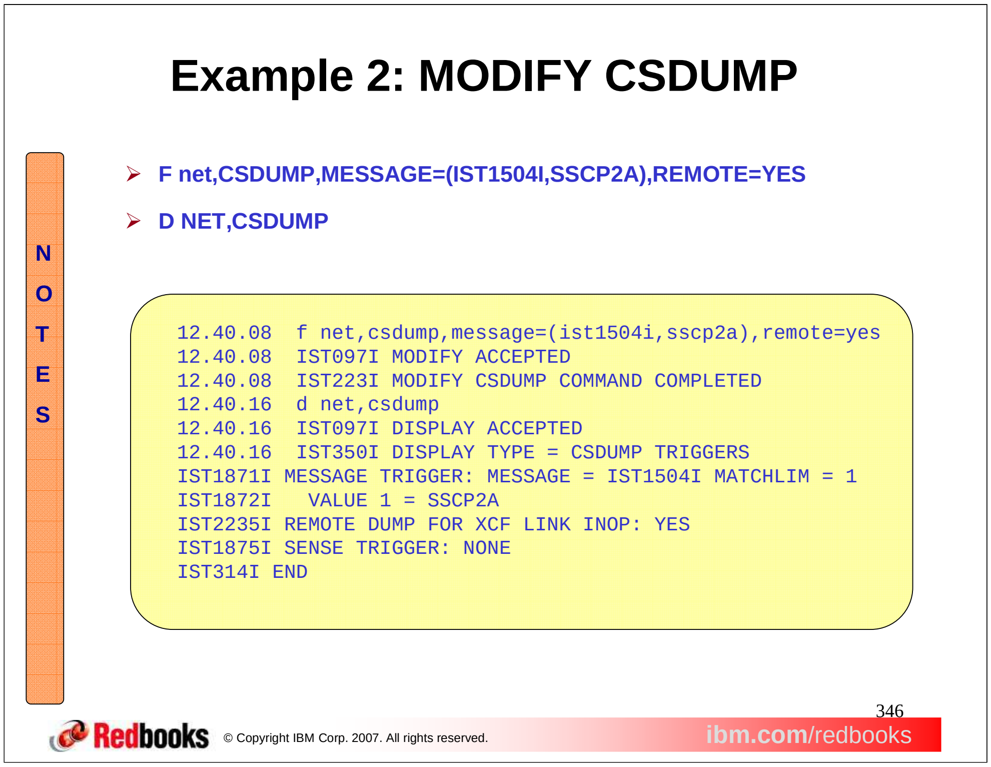
Example 1: MODIFY CSDUMP

NOTES

- F net,CSDUMP,MESSAGE=IST1504I,REMOTE=YES
- D NET,CSDUMP

```
12.22.17  f net,csdump,message=ist1504i,remote=yes
12.22.17  IST097I MODIFY ACCEPTED
12.22.17  IST223I MODIFY CSDUMP COMMAND COMPLETED
12.22.24  d net,csdump
12.22.24  IST097I DISPLAY ACCEPTED
12.22.24  IST350I DISPLAY TYPE = CSDUMP TRIGGERS
IST1871I MESSAGE TRIGGER: MESSAGE = IST1504I MATCHLIM = 1
IST2235I REMOTE DUMP FOR XCF LINK INOP: YES
IST1875I SENSE TRIGGER: NONE
IST314I END
```

In this example, a local dump and a dump of the remote host will be attempted if the XCF link to any other VTAM connected in the sysplex should INOP. The output from the D NET,CSDUMP command tells us that the REMOTE=YES was specified on the MODIFY CSDUMP command.



NOTES

Example 2: MODIFY CSDUMP

- F net,CSDUMP,MESSAGE=(IST1504I,SSCP2A),REMOTE=YES
- D NET,CSDUMP

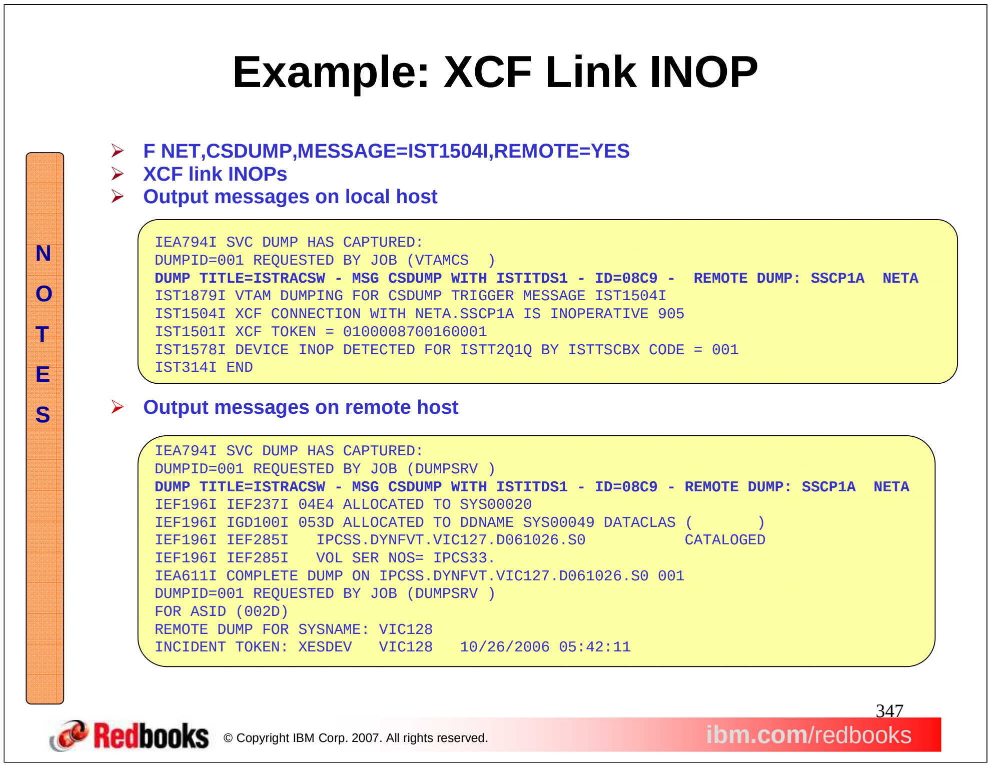
```
12.40.08  f net,csdump,message=(ist1504i,sscp2a),remote=yes
12.40.08  IST097I MODIFY ACCEPTED
12.40.08  IST223I MODIFY CSDUMP COMMAND COMPLETED
12.40.16  d net,csdump
12.40.16  IST097I DISPLAY ACCEPTED
12.40.16  IST350I DISPLAY TYPE = CSDUMP TRIGGERS
IST1871I MESSAGE TRIGGER: MESSAGE = IST1504I MATCHLIM = 1
IST1872I   VALUE 1 = SSCP2A
IST2235I REMOTE DUMP FOR XCF LINK INOP: YES
IST1875I SENSE TRIGGER: NONE
IST314I END
```

346

Redbooks ibm.com/redbooks

In this second example, the message trigger includes variable text to restrict it to an IST1504I message identifying a specific system (by CP Name).

The trigger will match, and a local dump and a remote dump will be requested, only if the XCF link to the named system INOPs. If an XCF link to another system INOPs, no local or remote dump will be attempted.



Example: XCF Link INOP

- F NET,CSDUMP,MESSAGE=IST1504I,REMOTE=YES
- XCF link INOPs
- Output messages on local host

```
IEA794I SVC DUMP HAS CAPTURED:
DUMPID=001 REQUESTED BY JOB (VTAMCS  )
DUMP TITLE=ISTRACSW - MSG CSDUMP WITH ISTITDS1 - ID=08C9 -  REMOTE DUMP: SSCP1A  NETA
IST1879I VTAM DUMPING FOR CSDUMP TRIGGER MESSAGE IST1504I
IST1504I XCF CONNECTION WITH NETA.SSCP1A IS INOPERATIVE 905
IST1501I XCF TOKEN = 0100008700160001
IST1578I DEVICE INOP DETECTED FOR ISTT2Q1Q BY ISTTSCBX CODE = 001
IST314I END
```

- Output messages on remote host

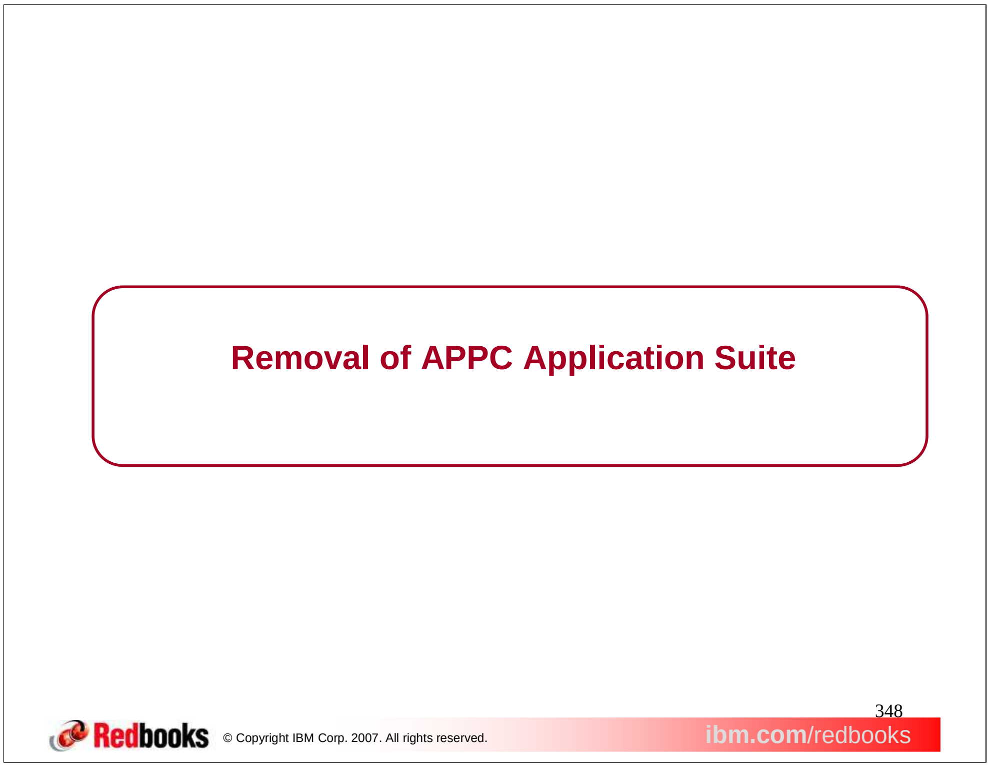
```
IEA794I SVC DUMP HAS CAPTURED:
DUMPID=001 REQUESTED BY JOB (DUMPSRV )
DUMP TITLE=ISTRACSW - MSG CSDUMP WITH ISTITDS1 - ID=08C9 - REMOTE DUMP: SSCP1A  NETA
IEF196I IEF237I 04E4 ALLOCATED TO SYS00020
IEF196I IGD100I 053D ALLOCATED TO DDNAME SYS00049 DATACLAS (        )
IEF196I IEF285I   IPCSS.DYNFVT.VIC127.D061026.S0           CATALOGED
IEF196I IEF285I   VOL SER NOS= IPCS33.
IEA611I COMPLETE DUMP ON IPCSS.DYNFVT.VIC127.D061026.S0 001
DUMPID=001 REQUESTED BY JOB (DUMPSRV )
FOR ASID (002D)
REMOTE DUMP FOR SYSNAME: VIC128
INCIDENT TOKEN: XESDEV   VIC128   10/26/2006 05:42:11
```

347

Redbooks ibm.com/redbooks

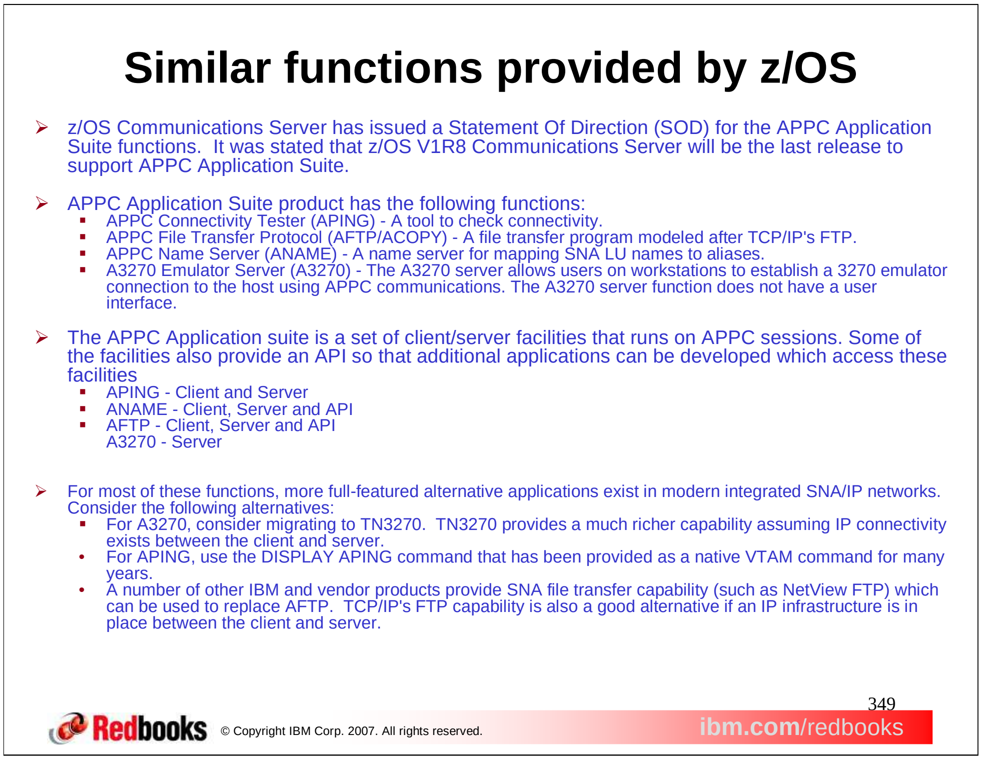
The top set of messages are seen on the VTAM host where the F net,CSDUMP,MESSAGE=IST1504I,REMOTE=YES command has been issued, and the XCF link INOPs. Note that the dump title includes the name of the remote host on which a dump is also requested.

The bottom set of messages are seen on the remote VTAM host when the XCF link INOPs. Note that the dump title is the same on both hosts. The IEA611 message identifies the host that requested that this remote dump be taken.



Removal of APPC Application Suite

This section covers the z/OS Communications Server Removal of APPC Application Suite for V1R9.



Similar functions provided by z/OS

- z/OS Communications Server has issued a Statement Of Direction (SOD) for the APPC Application Suite functions. It was stated that z/OS V1R8 Communications Server will be the last release to support APPC Application Suite.

- APPC Application Suite product has the following functions:
  - APPC Connectivity Tester (APING) - A tool to check connectivity.
  - APPC File Transfer Protocol (AFTP/ACOPY) - A file transfer program modeled after TCP/IP's FTP.
  - APPC Name Server (ANAME) - A name server for mapping SNA LU names to aliases.
  - A3270 Emulator Server (A3270) - The A3270 server allows users on workstations to establish a 3270 emulator connection to the host using APPC communications. The A3270 server function does not have a user interface.

- The APPC Application suite is a set of client/server facilities that runs on APPC sessions. Some of the facilities also provide an API so that additional applications can be developed which access these facilities
  - APING - Client and Server
  - ANAME - Client, Server and API
  - AFTP - Client, Server and API

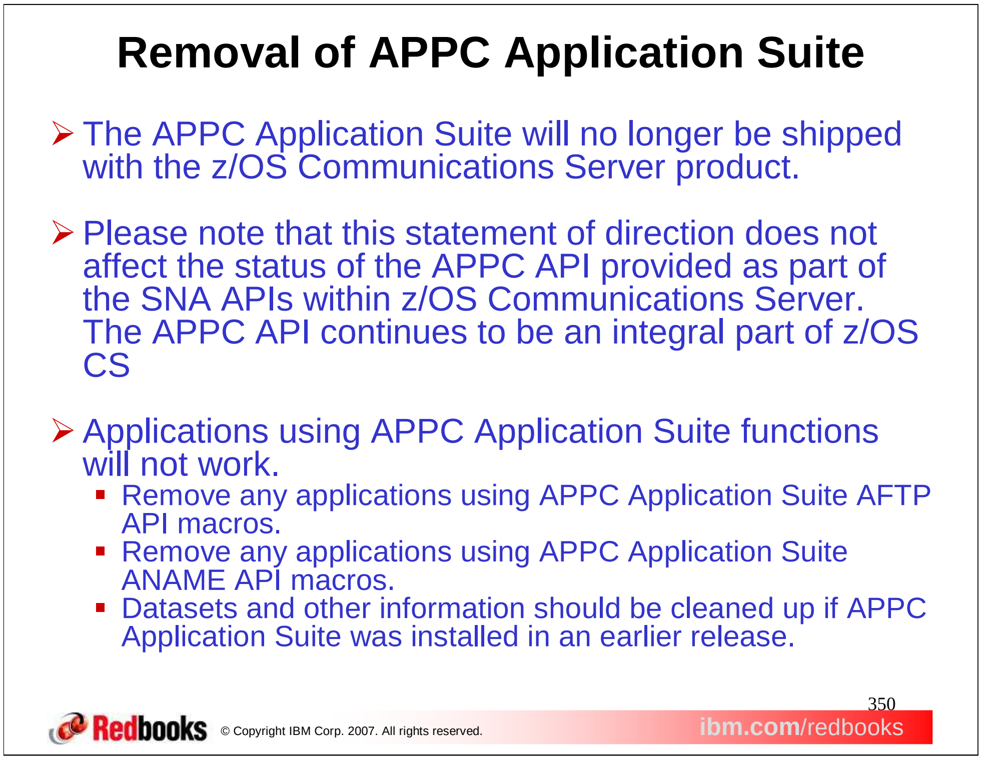
    A3270 - Server

- For most of these functions, more full-featured alternative applications exist in modern integrated SNA/IP networks. Consider the following alternatives:
  - For A3270, consider migrating to TN3270. TN3270 provides a much richer capability assuming IP connectivity exists between the client and server.
  - For APING, use the DISPLAY APING command that has been provided as a native VTAM command for many years.
  - A number of other IBM and vendor products provide SNA file transfer capability (such as NetView FTP) which can be used to replace AFTP. TCP/IP's FTP capability is also a good alternative if an IP infrastructure is in place between the client and server.

349

Redbooks ibm.com/redbooks

The APPC Application Suite was introduced in the early 1990s to provide some simple peer-to-peer based utilities for the then-emerging APPN (Advanced Peer-to-Peer Networking) world, and to help generate interest in APPC-based application development. The APPC Application Suite are LU 6.2 programs. They provide the common end-user tasks such as file transfer, terminal emulation, connectivity testing, and name registration. They also provide API supports.
SNA/IP provides similar functions to APPC Application Suite. There is no need to have the similar functions provided by the same product. So, the APPC Application Suite function will be removed.



Removal of APPC Application Suite

- The APPC Application Suite will no longer be shipped with the z/OS Communications Server product.
- Please note that this statement of direction does not affect the status of the APPC API provided as part of the SNA APIs within z/OS Communications Server. The APPC API continues to be an integral part of z/OS CS
- Applications using APPC Application Suite functions will not work.
  - Remove any applications using APPC Application Suite AFTP API macros.
  - Remove any applications using APPC Application Suite ANAME API macros.
  - Datasets and other information should be cleaned up if APPC Application Suite was installed in an earlier release.

Alternatives, now exist for the more useful utilities, and there is no longer any need to generate new interest in the mature APPC APIs. Therefore, z/OS V1R9 Communications Server discontinues support of the APPC Application Suite.
These are the steps to cleanup datasets and other information of the APPC Application Suite if you installed it in an earlier z/OS release:
•Delete AFTP client REXX exec.
•Delete ACOPY client REXX exec.
•Delete AFTP server REXX exec.
•Remove the transaction program (TP) profile of the AFTP server (AFTPD). This is the profile that defines AFTPD to MVS/APPC.
•Delete APING client REXX exec.
•Remove the transaction program (TP) profile of the APING server (APINGD). This is the profile that defines APINGD to MVS/APPC.
•Delete ANAME client REXX exec.
•Remove the transaction program (TP) profile of the ANAME server (ANAMED). This is the profile that defines ANAMED to MVS/APPC.
•Remove the A3270 application major node definition from VTAMLST.
•Delete initialization files of AFTP programs (AFTP, AFTPD, and ACOPY), the ANAME server, and the A3270 server.
•Delete VSAM dataset with the APPC Application Suite messages.

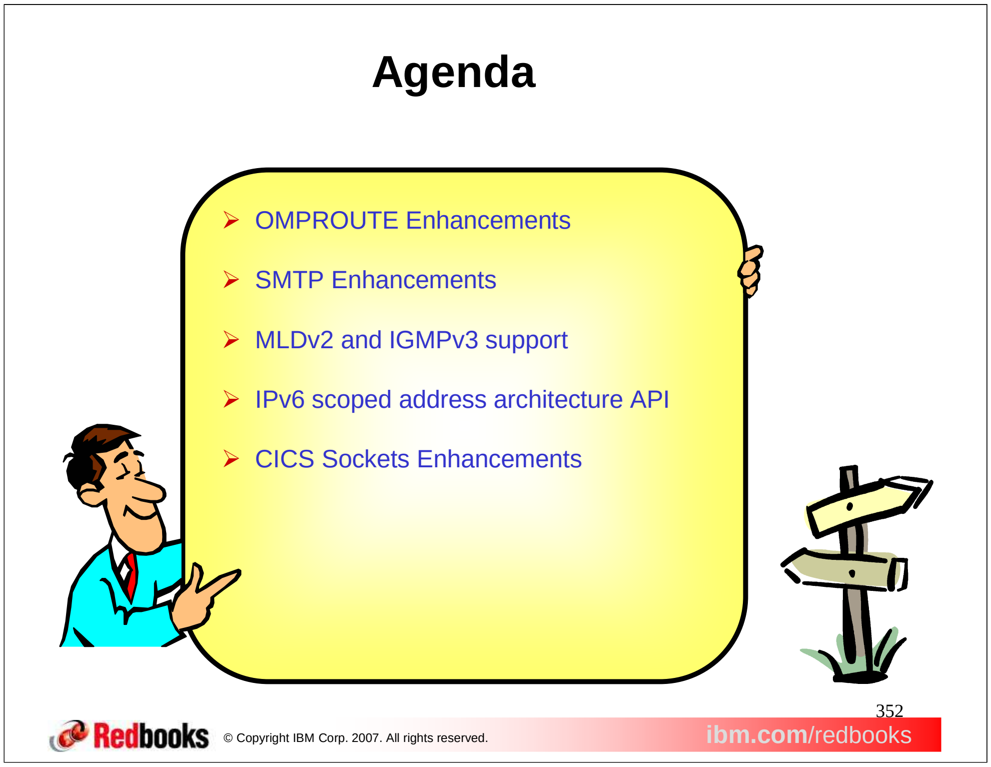
Refer to the z/OS New Functions Summary and the z/OS Migration books for additional information.

ibm.com

Miscellaneous

International Technical Support Organization

This presentation describes miscellaneous enhancements to z/OS V1R9 Communications Server.



Agenda

- OMPROUTE Enhancements
- SMTP Enhancements
- MLDv2 and IGMPv3 support
- IPv6 scoped address architecture API
- CICS Sockets Enhancements

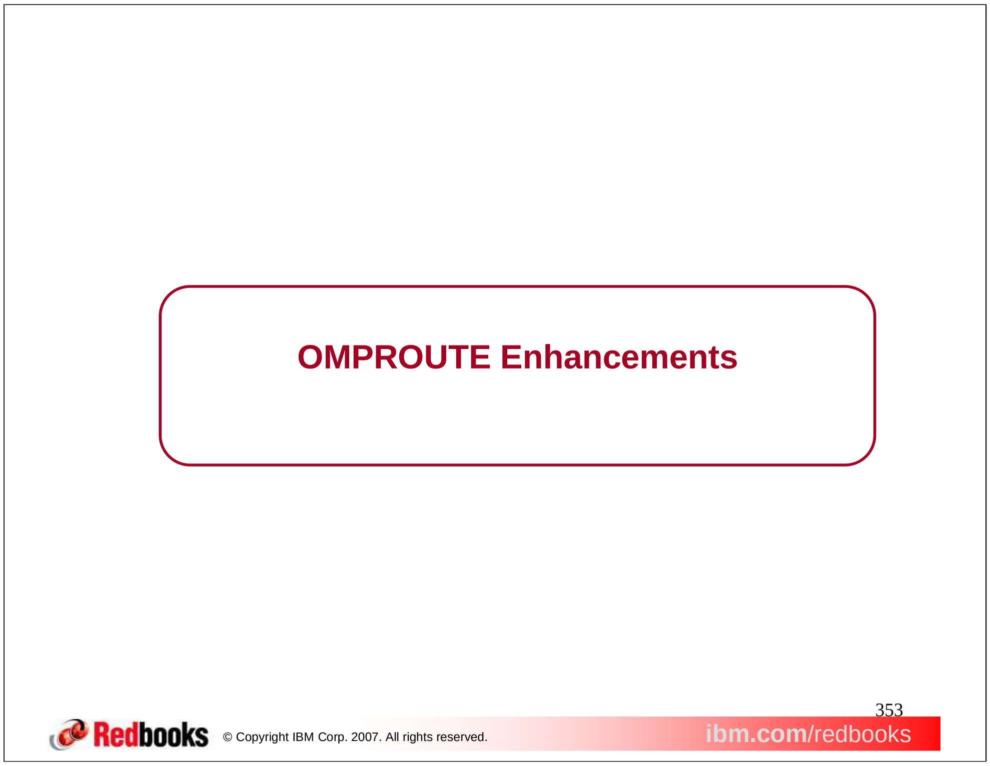
There are several OMPROUTE enhancements in V1R9. These enhancements stem from requirements that were made.

SMTP Enhancements consist of a new SMTP configuration statement (REMOTEPORT) and new operator commands.

The new versions of the Internet Group Management Protocol for IPv4 , IGMPv3, and Multicast Listening Discovery for IPv6, MLDv2 were implemented in z/OS V1R9 Communications Server.

IPv6 scoped address architecture API added support for the IPv6 scoped address API changes, primarily to the Resolver Getaddrinfo and Getnameinfo functions.

There are also some enhancements to CICS Sockets for improved availability.

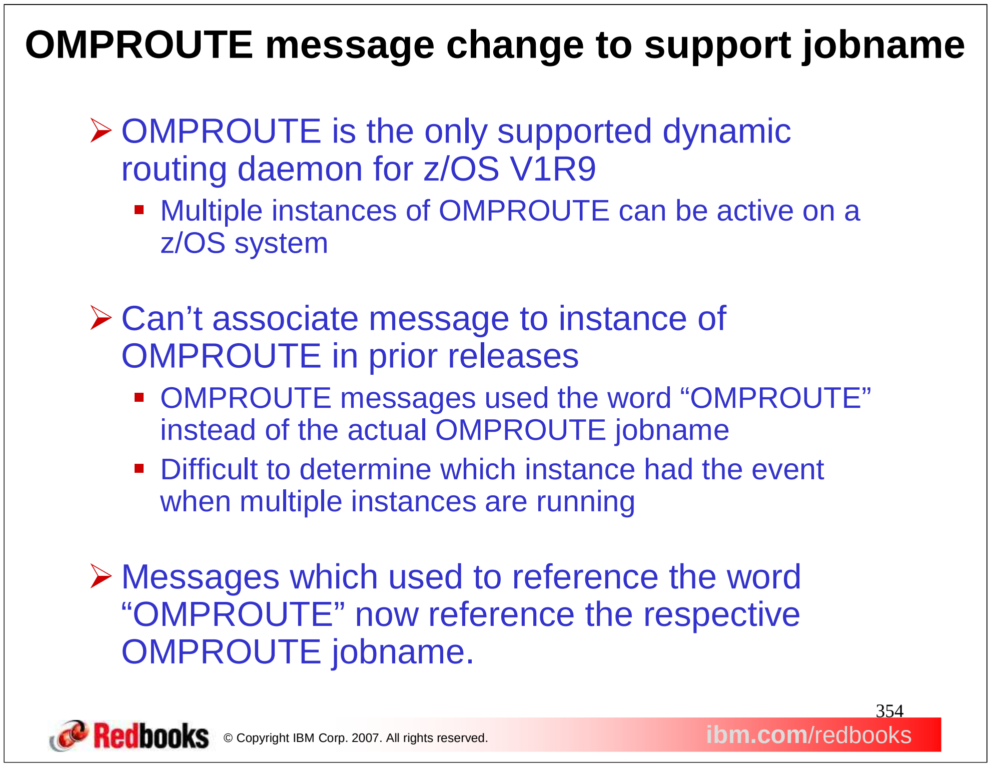


OMPROUTE Enhancements

353

ibm.com/redbooks

This section covers the enhancements made to OMPROUTE in z/OS V1R9.



OMPROUTE message change to support jobname

- OMPROUTE is the only supported dynamic routing daemon for z/OS V1R9
  - Multiple instances of OMPROUTE can be active on a z/OS system
- Can’t associate message to instance of OMPROUTE in prior releases
  - OMPROUTE messages used the word “OMPROUTE” instead of the actual OMPROUTE jobname
  - Difficult to determine which instance had the event when multiple instances are running
- Messages which used to reference the word “OMPROUTE” now reference the respective OMPROUTE jobname.

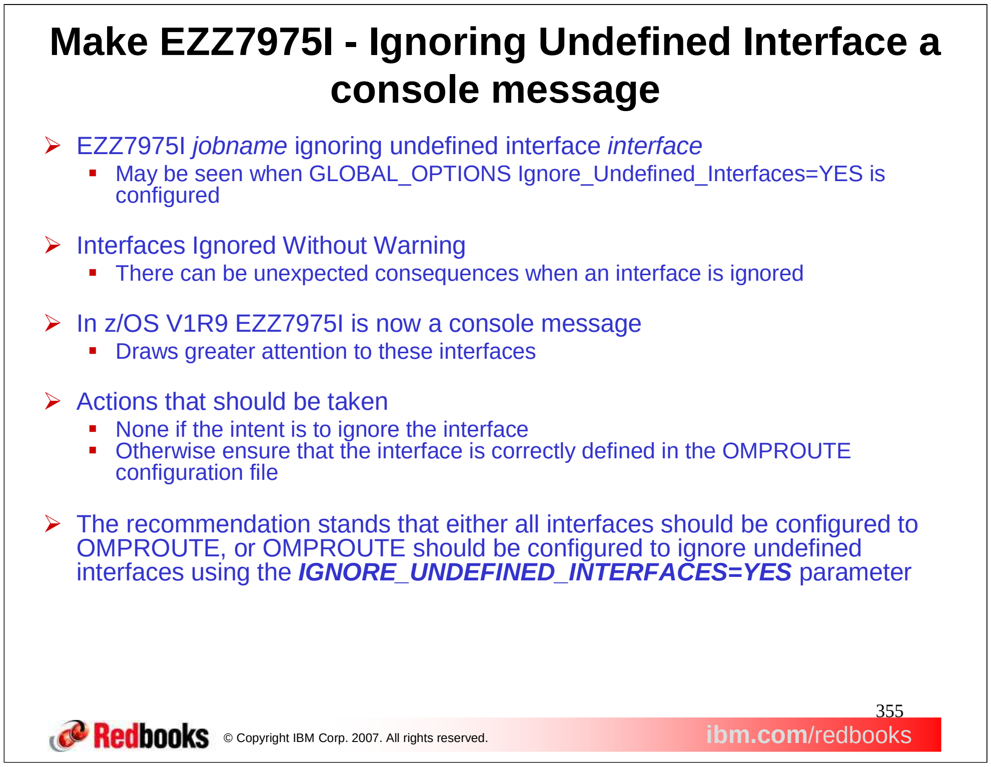
354

ibm.com/redbooks

OMPROUTE is the only supported dynamic routing daemon in z/OS V1R9. It supports RIP, OSPF, IPv6 RIP and IPv6 OSPF routing protocols. In a Common INET environment a copy of OMPROUTE must be started for each stack that requires OMPROUTE services. Therefore multiple instances of OMPROUTE can be active on a z/OS system.

Prior to z/OS V1R9, some OMPROUTE messages used the word “OMPROUTE” instead of the actual OMPROUTE jobname to indicate which instance of OMPROUTE triggered the message. If more than one OMPROUTE instance is running, and a message indicated an event by “OMPROUTE”, there was potential for error to determine which instance of OMPROUTE was associated with the event.

Messages which previously referenced “OMPROUTE”, now reference the jobname for the instance of OMPROUTE which triggered the message.



Make EZZ7975I - Ignoring Undefined Interface a console message

- EZZ7975I *jobname* ignoring undefined interface *interface*
  - May be seen when GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES is configured
- Interfaces Ignored Without Warning
  - There can be unexpected consequences when an interface is ignored
- In z/OS V1R9 EZZ7975I is now a console message
  - Draws greater attention to these interfaces
- Actions that should be taken
  - None if the intent is to ignore the interface
  - Otherwise ensure that the interface is correctly defined in the OMPROUTE configuration file
- The recommendation stands that either all interfaces should be configured to OMPROUTE, or OMPROUTE should be configured to ignore undefined interfaces using the ***IGNORE_UNDEFINED_INTERFACES=YES*** parameter

355

Redbooks

ibm.com/redbooks

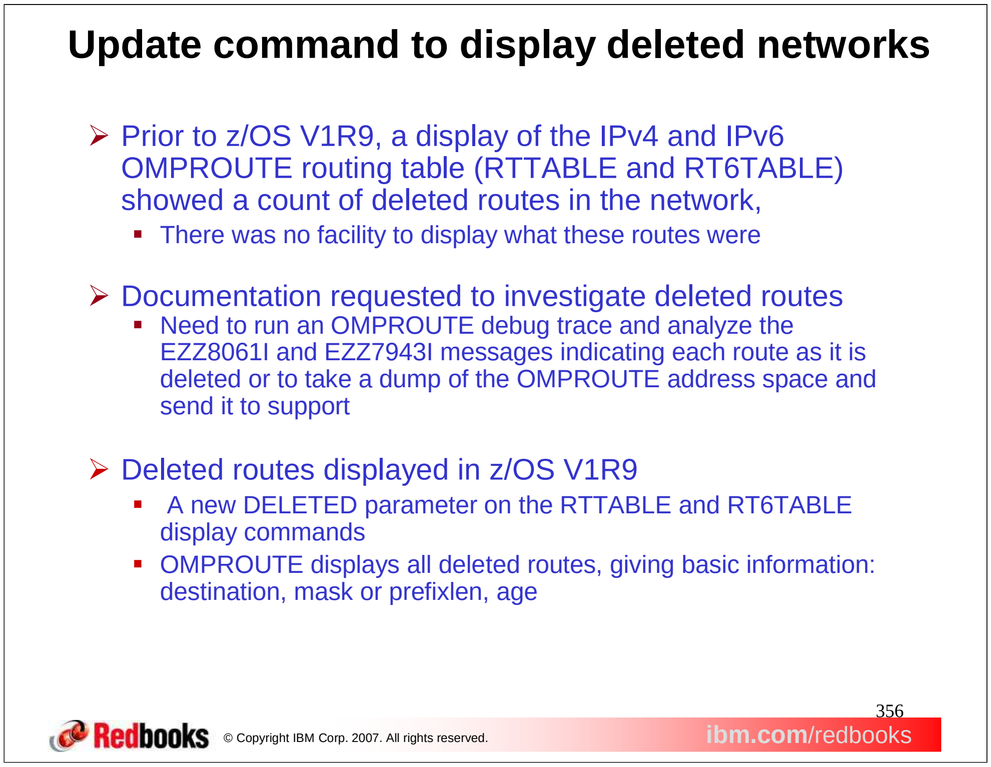
The EZZ7975I *jobname* ignoring undefined interface *interface* message is seen when the GLOBAL_OPTIONS Ignored_Undefined_Interfaces=YES parameter is configured in the OMPROUTE configuration file, and the interface *interface* learned from the TCPIP stack is not defined (or not properly defined) as an IPv4 or IPv6 Interface, RIP_Interface, or OSPF_Interface (as appropriate) in the OMPROUTE configuration file. As a result, OMPROUTE will not update the BSDROUTINGPARMs for the interface in the stack; rather the default MTU of 576 and class mask will be used for IPv4 interfaces which are ignored. (This is not an issue for IPv6 Interfaces). Also, neither the home address (IPv4 and IPv6) or the subnet (IPv4 only) will be advertised via routing protocol, OMPROUTE will not add a direct route to the interface subnet (IPv4 only), and static routes which use this interface will not be accepted from TCP/IP and therefore will not be advertised (both IPv4 and IPv6).

Service has seen instances where routing problems are caused because users fail to define their interfaces to OMPROUTE.   Many routing problems seen in the support stream are caused because of either improperly defined interfaces, or ignored interfaces. These include incorrectly or inadvertently advertised routes or the appearance that expected routes are not being advertised. The problems can be difficult to diagnose without gathering documentation.

Because so many problems have been reported because of improperly defined or undefined interfaces, the EZZ7975I warning message will be output to the console starting in z/OS V1R9. This message will be seen for any interfaces which are defined to TCPIP that do not have a properly coded matching OMPROUTE interface definition if GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES is coded. This will draw greater attention to these interfaces so that userss who have ignored interfaces that they did not intend to ignore will recognize their error more quickly.

If you are intentionally using GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES to not advertise routing information about particular interfaces, and the interface cited in the message text is one of those interfaces you do not want to advertise, then this message does not indicate a problem. However, if you do want the IP address of the interface cited in the EZZ7975I message to be advertised via routing protocol to other routers, you will need to either add or correct the definition for this interface in the OMPROUTE configuration file.

The recommendation stands that either all interfaces should be defined to OMPROUTE OR if GLOBAL_OPTIONS Ignore_Undefined_Interface=YES, care must be taken to ensure that any interfaces that are ignored should be ignored and vice versa.



Update command to display deleted networks

- Prior to z/OS V1R9, a display of the IPv4 and IPv6 OMPROUTE routing table (RTTABLE and RT6TABLE) showed a count of deleted routes in the network,
  - There was no facility to display what these routes were
- Documentation requested to investigate deleted routes
  - Need to run an OMPROUTE debug trace and analyze the EZZ8061I and EZZ7943I messages indicating each route as it is deleted or to take a dump of the OMPROUTE address space and send it to support
- Deleted routes displayed in z/OS V1R9
  - A new DELETED parameter on the RTTABLE and RT6TABLE display commands
  - OMPROUTE displays all deleted routes, giving basic information: destination, mask or prefixlen, age

356

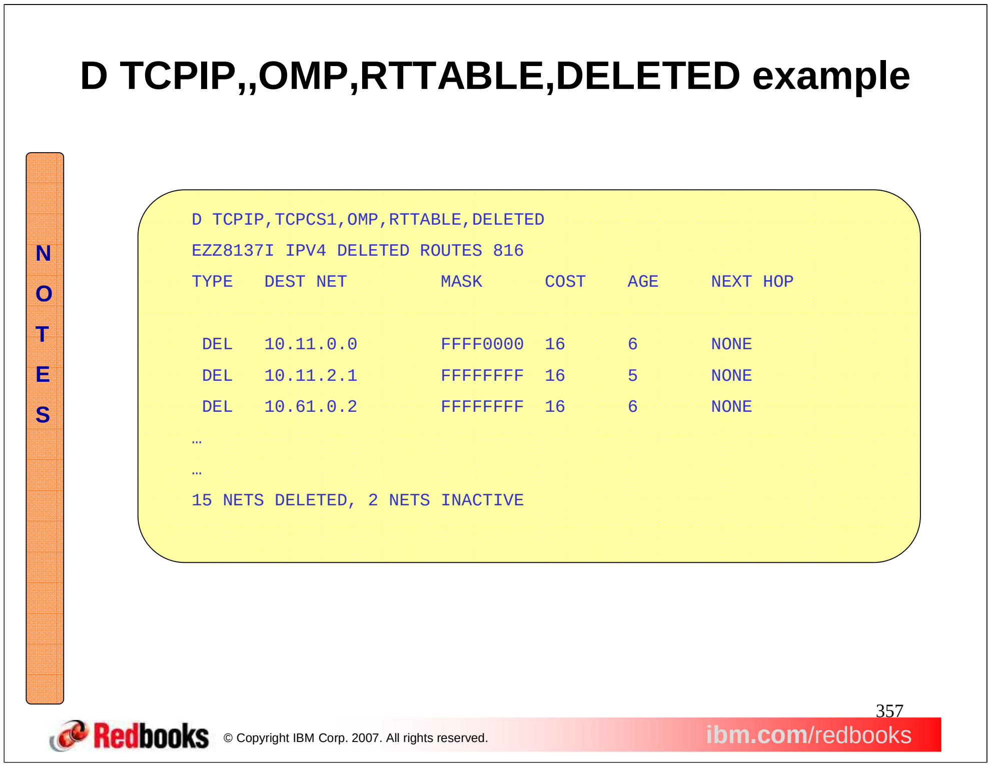
Redbooks

ibm.com/redbooks

Prior to z/OS V1R9, the OMPROUTE RTTABLE and RT6TABLE displays showed a count of deleted routes, but there was no way to see what these routes actually were.

Because there was no way to display which routes were deleted, you needed to either run an OMPROUTE debug trace and individually find the entries indicating that a route had been deleted, or provide level 2 with a dump of the OMPROUTE address space to extract these routes.

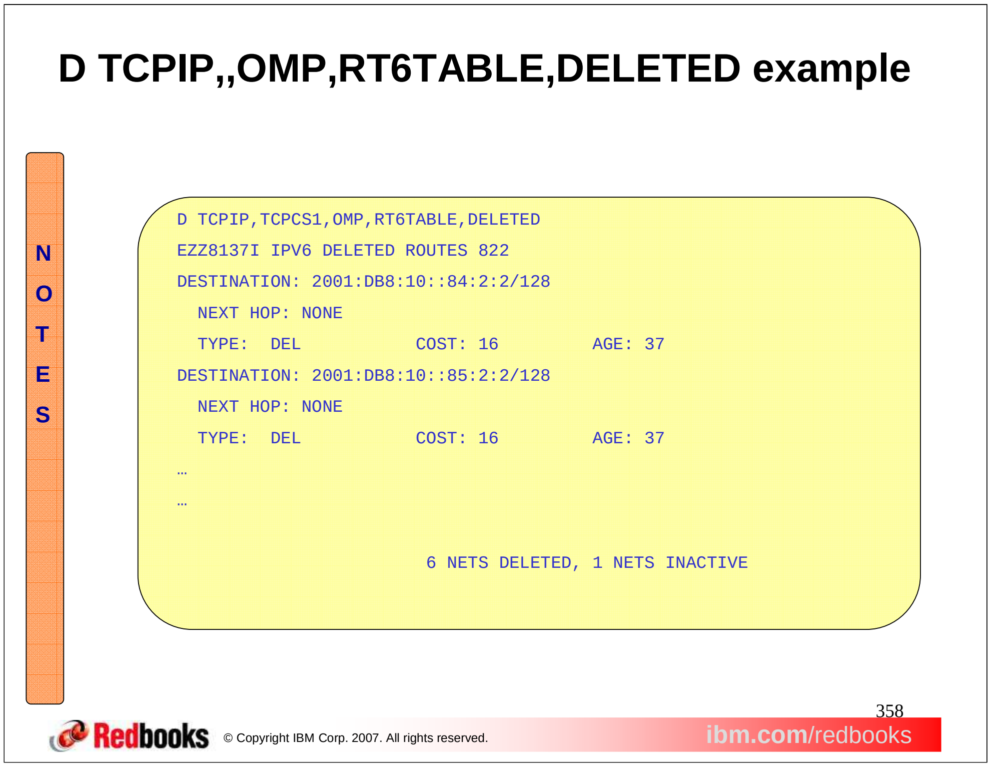
To ease the ability to see the deleted routes, the capability to display deleted routes has been added to the existing RTTABLE and RT6TABLE displays. In z/OS V1R9, a new DELETED parameter has been added to the RTTABLE and RT6TABLE display and modify commands. This parameter can also be used with the new policy based routing OMPROUTE RTTABLE displays for IPv4.



D TCPIP,,OMP,RTTABLE,DELETED example

```
D TCPIP,TCPCS1,OMP,RTTABLE,DELETED
EZZ8137I IPV4 DELETED ROUTES 816
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP

 DEL   10.11.0.0        FFFF0000  16      6       NONE
 DEL   10.11.2.1        FFFFFFFF  16      5       NONE
 DEL   10.61.0.2        FFFFFFFF  16      6       NONE
…
…
15 NETS DELETED, 2 NETS INACTIVE
```

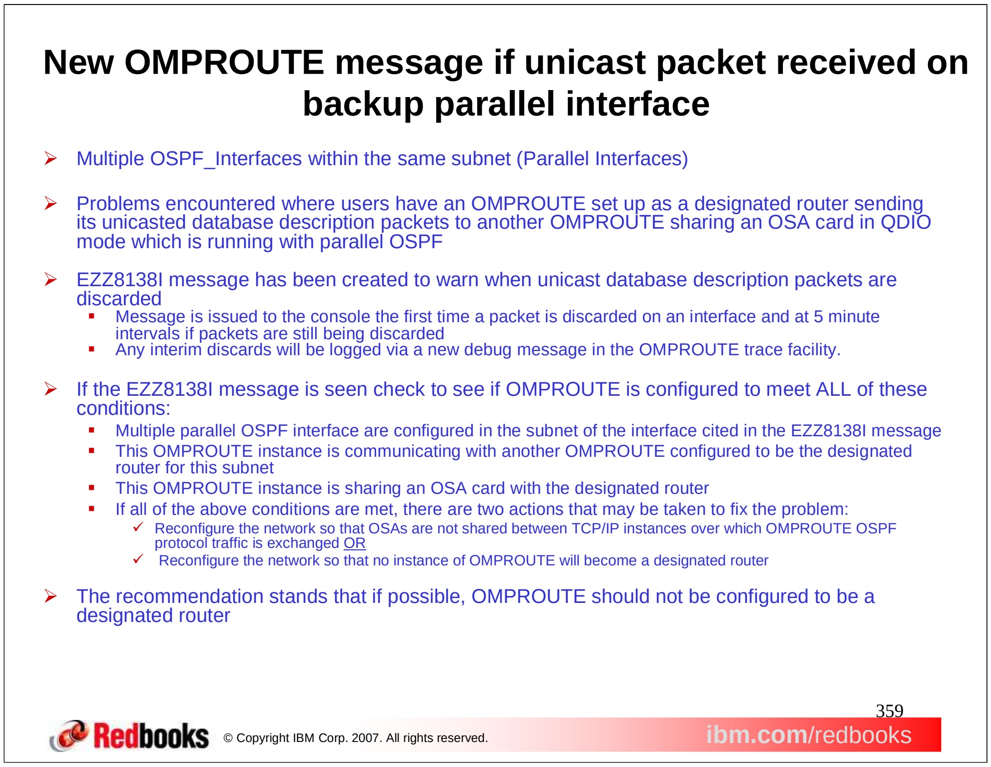
This is an example of the output of the D TCPIP,OMP,RTTABLE,DELETED command. This command can be issued to see all deleted routes in OMPROUTE’s main IPv4 routing table. The same information can also be seen in the F OMP,RTTABLE,DELETED display.



D TCPIP,,OMP,RT6TABLE,DELETED example

```
D TCPIP,TCPCS1,OMP,RT6TABLE,DELETED
EZZ8137I IPV6 DELETED ROUTES 822
DESTINATION: 2001:DB8:10::84:2:2/128
  NEXT HOP: NONE
  TYPE:  DEL           COST: 16         AGE: 37
DESTINATION: 2001:DB8:10::85:2:2/128
  NEXT HOP: NONE
  TYPE:  DEL           COST: 16         AGE: 37
…
…

                        6 NETS DELETED, 1 NETS INACTIVE
```

This is an example of the output of the D TCPIP,OMP,RT6TABLE,DELETED command. This command can be issued to see all deleted routes in OMPROUTE’s main IPv6 routing table. The same output can be seen in the F OMPROUTE,RT6TABLE,DELETED display.



New OMPROUTE message if unicast packet received on backup parallel interface

- Multiple OSPF_Interfaces within the same subnet (Parallel Interfaces)
- Problems encountered where users have an OMPROUTE set up as a designated router sending its unicasted database description packets to another OMPROUTE sharing an OSA card in QDIO mode which is running with parallel OSPF
- EZZ8138I message has been created to warn when unicast database description packets are discarded
  - Message is issued to the console the first time a packet is discarded on an interface and at 5 minute intervals if packets are still being discarded
  - Any interim discards will be logged via a new debug message in the OMPROUTE trace facility.
- If the EZZ8138I message is seen check to see if OMPROUTE is configured to meet ALL of these conditions:
  - Multiple parallel OSPF interface are configured in the subnet of the interface cited in the EZZ8138I message
  - This OMPROUTE instance is communicating with another OMPROUTE configured to be the designated router for this subnet
  - This OMPROUTE instance is sharing an OSA card with the designated router
  - If all of the above conditions are met, there are two actions that may be taken to fix the problem:
    - ✓ Reconfigure the network so that OSAs are not shared between TCP/IP instances over which OMPROUTE OSPF protocol traffic is exchanged OR
    - ✓ Reconfigure the network so that no instance of OMPROUTE will become a designated router
- The recommendation stands that if possible, OMPROUTE should not be configured to be a designated router

Redbooks

ibm.com/redbooks

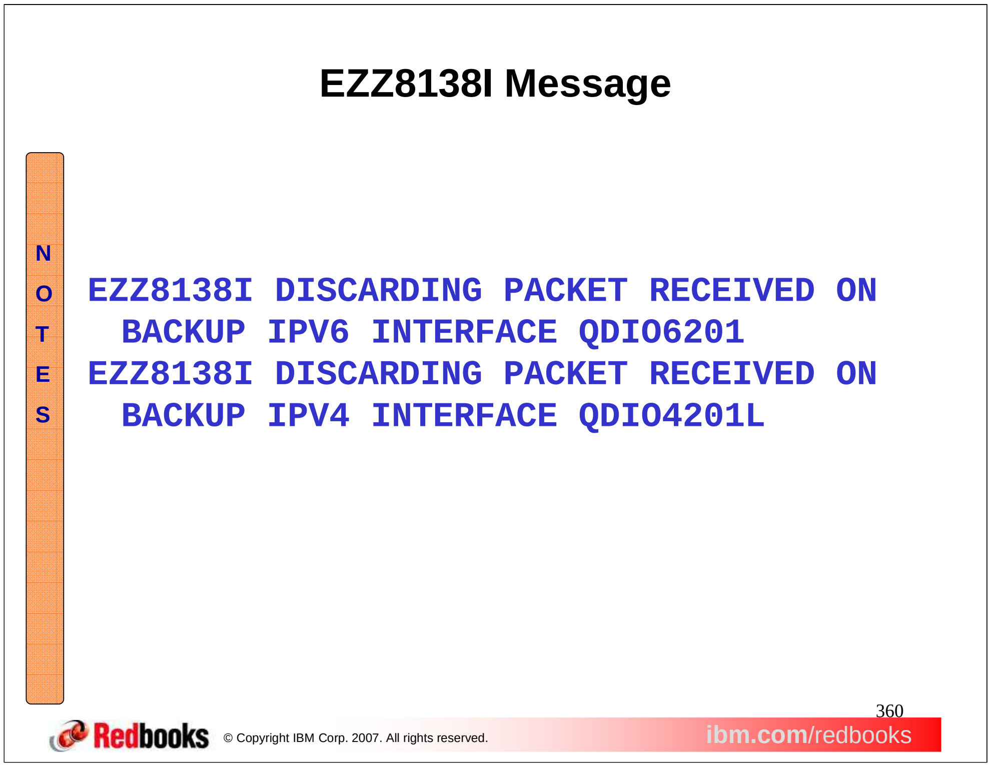
When OMPROUTE is started, if two or more OSPF_Interfaces (or IPv6_OSPF_Interfaces) are in the same subnet (or on the same link, for IPv6), then one of those interfaces will be chosen as the primary OSPF interface. A primary interface can be assigned at the start of OMPROUTE via the Parallel_OSPF=Primary parameter on the OSPF_Interface and IPv6_OSPF_Interface statements, otherwise one will be chosen by OMPROUTE as the primary. The primary OSPF interface is the interface over which the actual OSPF protocol traffic will flow. All other interfaces in that subnet (or on that link) will be given backup status, which means if the primary interface is taken down, one of the backups will take over the responsibility of sending and receiving OSPF protocol traffic. There is a phase of the process of forming an adjacency where OSPF unicast packets called database description packets are sent and received between OMPROUTE and the designated router, and the OSPF protocol specification requires that OMPROUTE receive these packets on the primary OSPF interface. Any packets received on a backup OSPF interface will be discarded.

Service has seen a number of reported problems where an OMPROUTE with parallel OSPF interfaces is exchanging database description packets with another OMPROUTE which is the designated router, and these two OMPROUTEs share an OSA card in QDIO mode between them. Shared QDIO is wonderful because there is no need to send a packet out into the network if you can just send it to its destination over the OSA card. When the packet is received over the card it is given a “destination address” and then passed up to TCPIP and OMPROUTE. The problem is that this “destination IP address” may or may not be the actual IP address to which the sender sent the packet originally. Most applications do not care what interface a packet is received on, but OMPROUTE does, and if the wrong destination address is assigned by the receiver of a packet learned over the OSA card, and that wrong destination address is for one of the backup parallel ospf interfaces, the packet will be discarded. This can cause neighbor state regressions and loops, or if an adjacency had previously formed, adjacency failures can occur, resulting in the loss of routes.

A new message, EZZ8138I, has been created to output to the console the first time a packet is received on a backup parallel ospf interface and discarded. After the first time this message is seen, it will be suppressed and only be issued to the console every 5 minutes to prevent flooding. Any interim discards are logged in an OMPROUTE trace if one is running. This message will aid in diagnosis of the problem cited above You can only be experiencing this particular problem if you meet all the criteria; 1) multiple parallel OSPF interfaces in the same subnet (or link for IPv6), 2) the designated router for this subnet is another OMPROUTE instance, and 3) this OMPROUTE is sharing an OSA card in QDIO mode with the designated router OMPROUTE. If this is your problem, there are two ways to fix it: 1) reconfigure the network so that the OSAs are not shared between the designated router OMPROUTE’s TCPIP stack and any other OMPROUTE’s TCPIP stacks, or 2) reconfigure the network so that no instance of OMPROUTE will be the designated router (if possible).

If you do not meet the criteria for being in the shared QDIO problem, but you are still seeing EZZ8138I messages, then it would appear that some other problem is occurring or some other router is sending unsolicited OSPF protocol traffic to one or more of your backup parallel interfaces. In that case, you may want to check to see if packets are being leaked across VLANs, if those are in use. If not, the documentation that needs to be gathered is a –t2 –d1 (or –6t2 -6d3 for IPv6 OSPF) trace and a dump of both OMPROUTE and TCPIP’s address spaces.

If possible, do not let OMPROUTE become the designated router for the subnet or link. You can configure the Router_Priority=0 parameter on the OSPF_Interface or IPv6_OSPF_Interface statement to keep an instance of OMPROUTE from becoming a designated router. Be careful though – if you ONLY have OMPROUTE instances in your subnet, one of them then HAS to be a designated router. In that case, chose that designated router so that you can avoid sharing an OSA between it and any other OMPROUTE in that subnet.

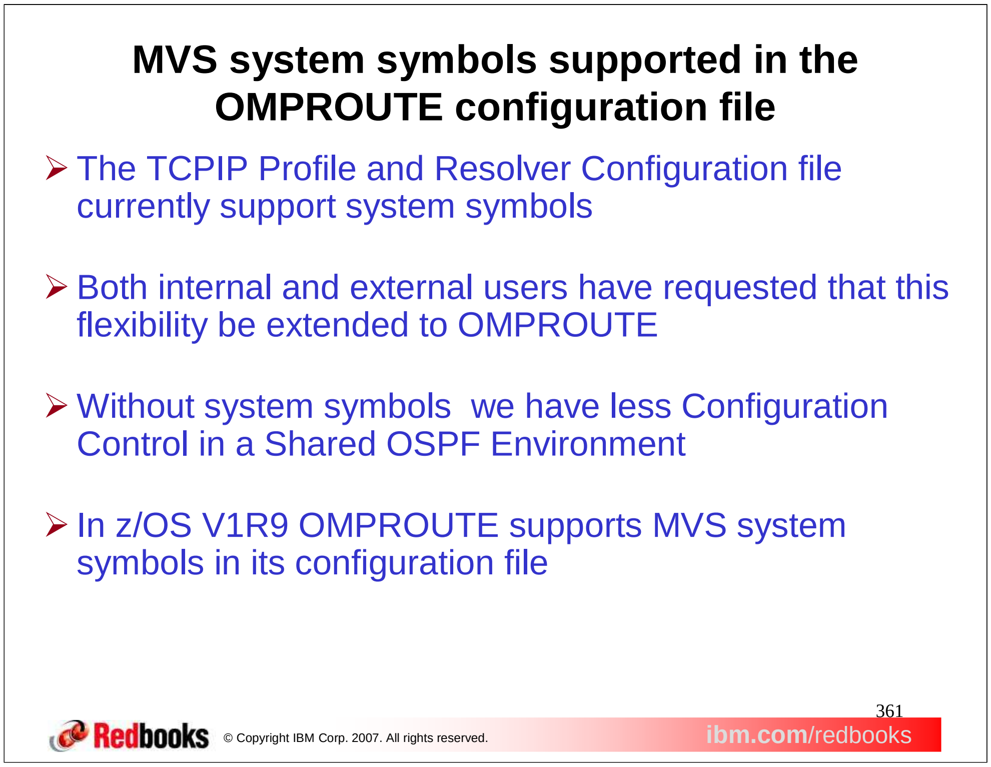


EZZ8138I Message

N O T E S

```
EZZ8138I DISCARDING PACKET RECEIVED ON
  BACKUP IPV6 INTERFACE QDIO6201
EZZ8138I DISCARDING PACKET RECEIVED ON
  BACKUP IPV4 INTERFACE QDIO4201L
```

Here are two examples of what the new EZZ8138I message looks like for an IPv4 and IPv6 OSPF interface.



MVS system symbols supported in the OMPROUTE configuration file

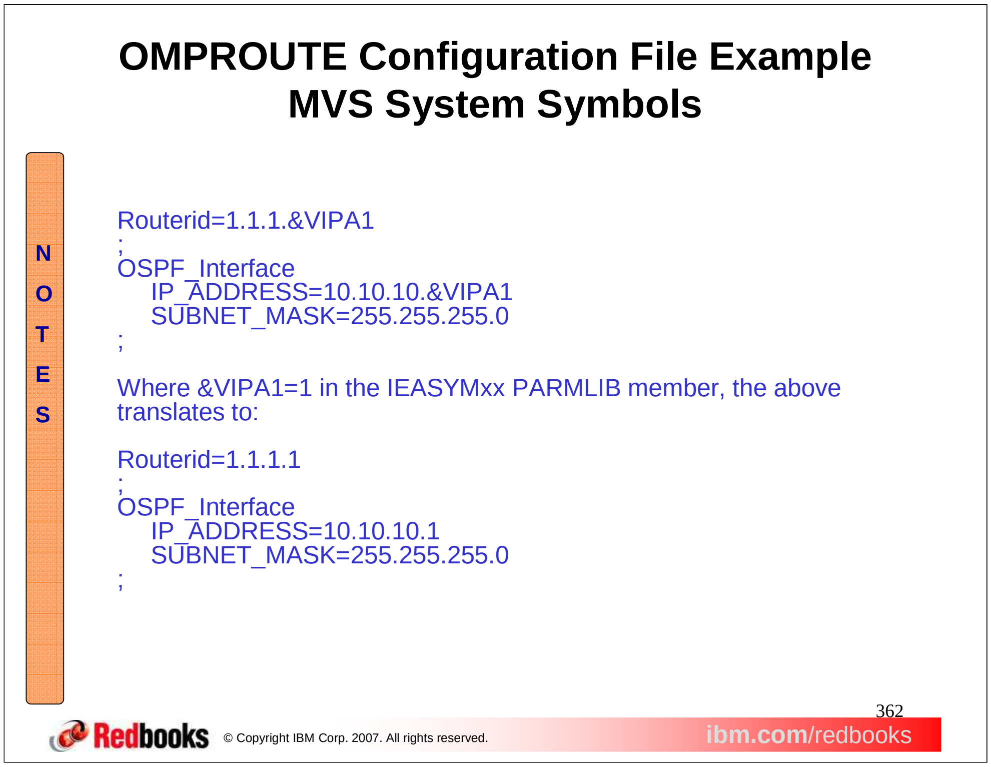
- The TCPIP Profile and Resolver Configuration file currently support system symbols
- Both internal and external users have requested that this flexibility be extended to OMPROUTE
- Without system symbols we have less Configuration Control in a Shared OSPF Environment
- In z/OS V1R9 OMPROUTE supports MVS system symbols in its configuration file

Prior to V1R9, the TCPIP Profile and Resolver configuration files supported MVS system symbols, however OMPROUTE did not. Both internal and external users have been interested in adding this functionality.

The ability to use the MVS system symbols in the OMPROUTE configuration file is nice in and of itself because now OMPROUTE configuration files can be shared between OMPROUTE instances. It was possible to share configuration files between OMPROUTE instances prior to V1R9 by using wildcarding; however in an OSPF environment there was no way to wildcard the Routerid, so if you did share configuration files, there was no way to specify a unique routerid for each OMPROUTE instance.

OMPROUTE now supports MVS system symbols in its configuration files.

If you need to see how a symbol was translated, turn on –t2 –d1 OMPROUTE trace and look for the text “Translated to”. For each line that contained an MVS system symbol there will be a line in the trace file which shows to what the symbol was translated.



OMPROUTE Configuration File Example MVS System Symbols

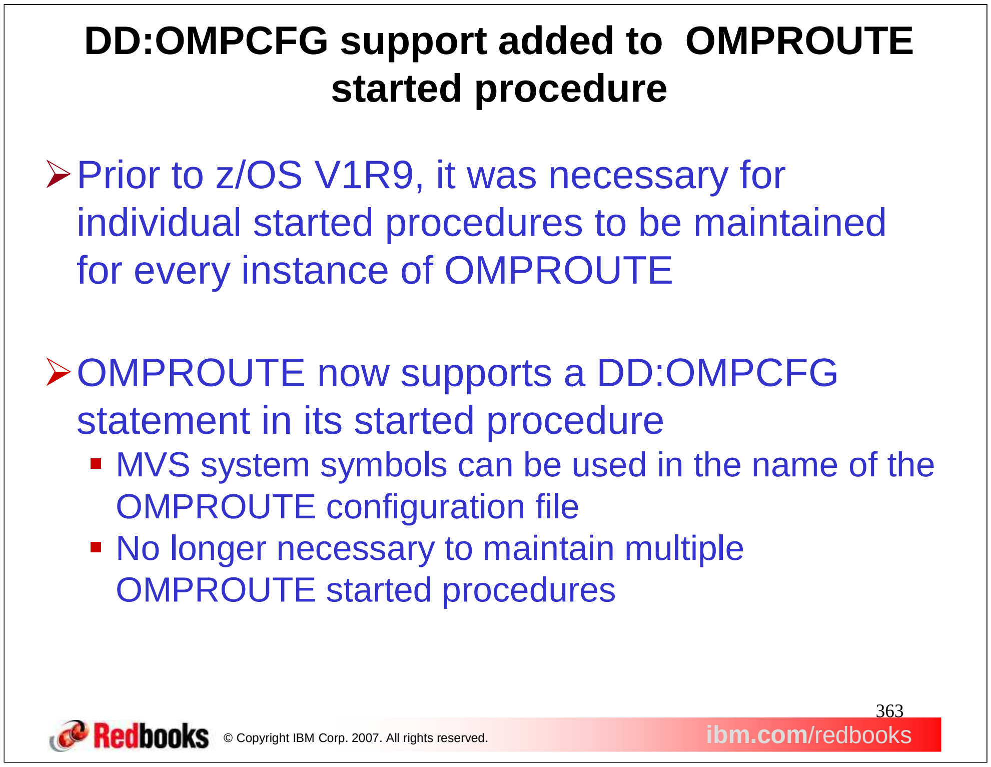
N O T E S

```
Routerid=1.1.1.&VIPA1
;
OSPF_Interface
    IP_ADDRESS=10.10.10.&VIPA1
    SUBNET_MASK=255.255.255.0
;
```

Where &VIPA1=1 in the IEASYMxx PARMLIB member, the above translates to:

```
Routerid=1.1.1.1
;
OSPF_Interface
    IP_ADDRESS=10.10.10.1
    SUBNET_MASK=255.255.255.0
;
```

This is an example of how symbol translation can be used in the OMPROUTE configuration file.

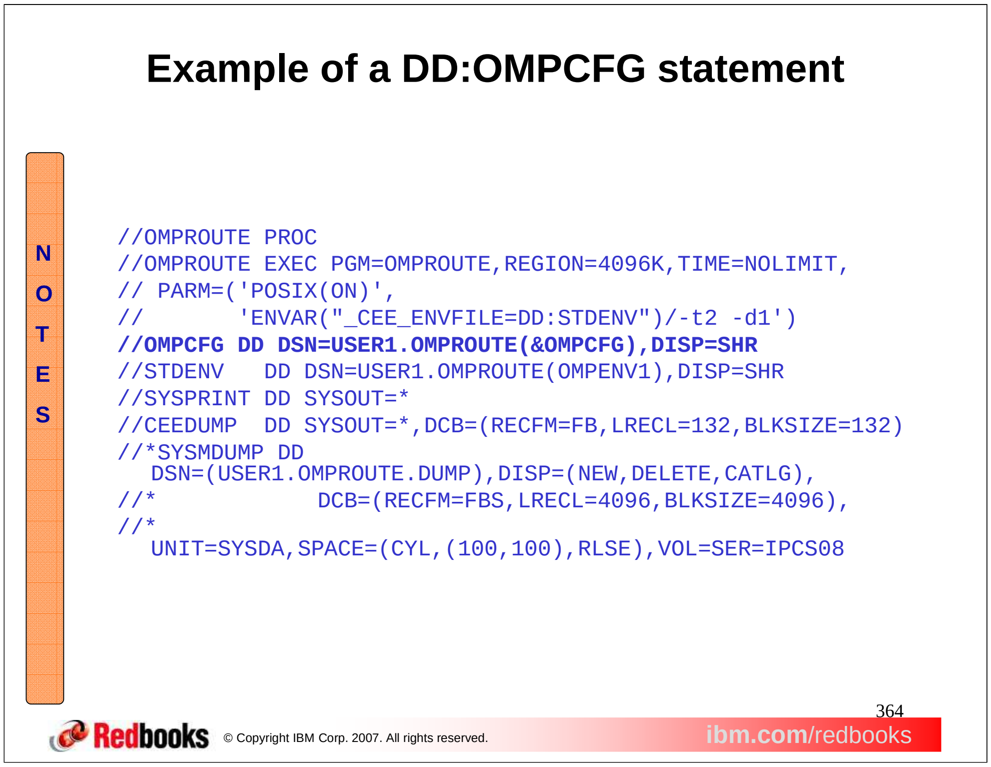


DD:OMPCFG support added to OMPROUTE started procedure

- Prior to z/OS V1R9, it was necessary for individual started procedures to be maintained for every instance of OMPROUTE
- OMPROUTE now supports a DD:OMPCFG statement in its started procedure
  - MVS system symbols can be used in the name of the OMPROUTE configuration file
  - No longer necessary to maintain multiple OMPROUTE started procedures

Prior to z/OS V1R9 it was necessary for individual started procedures to be maintained for each OMPROUTE instance.

Both internal and external users have requested a way to specify an OMPROUTE configuration file name which includes an MVS system symbol in the started procedure for OMPROUTE, so that one started procedure could be shared by multiple OMPROUTE instances.
OMPROUTE now supports a DD:OMPCFG statement in its started procedure. This allows for MVS system symbols to be used in the name of the OMPROUTE configuration file, eliminating the necessity to maintain multiple OMPROUTE started procedures



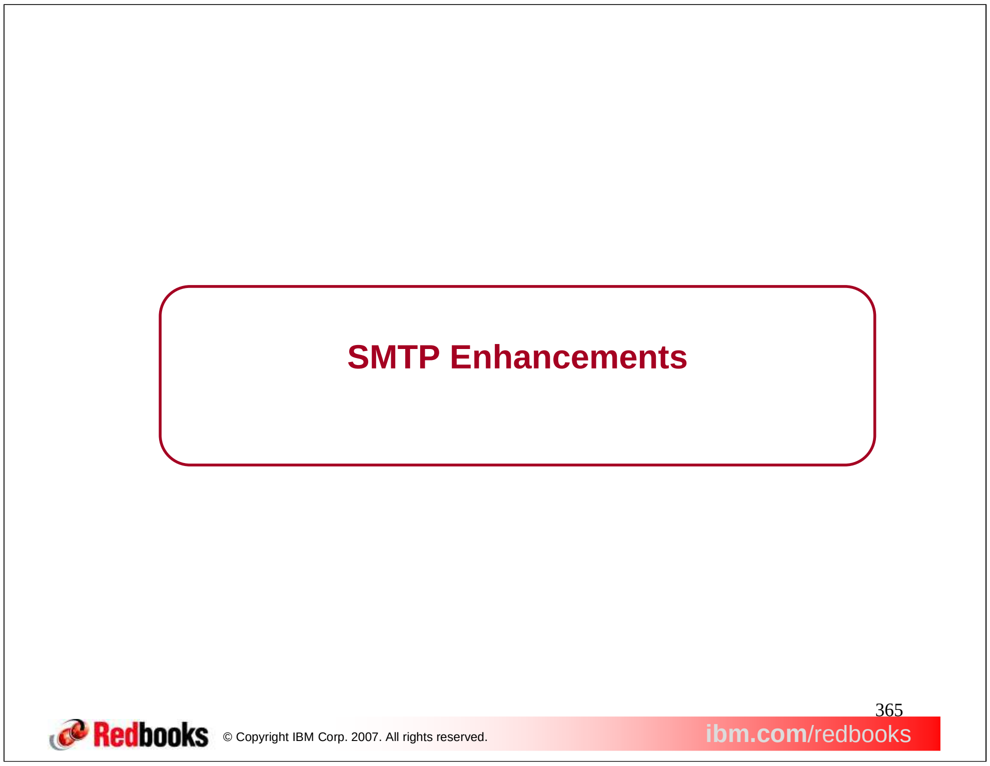
Example of a DD:OMPCFG statement

```
//OMPROUTE PROC
//OMPROUTE EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
// PARM=('POSIX(ON)',
//       'ENVAR("_CEE_ENVFILE=DD:STDENV")/-t2 -d1')
//OMPCFG DD DSN=USER1.OMPROUTE(&OMPCFG),DISP=SHR
//STDENV   DD DSN=USER1.OMPROUTE(OMPENV1),DISP=SHR
//SYSPRINT DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//*SYSMDUMP DD
   DSN=(USER1.OMPROUTE.DUMP),DISP=(NEW,DELETE,CATLG),
//*            DCB=(RECFM=FBS,LRECL=4096,BLKSIZE=4096),
//*
   UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),VOL=SER=IPCS08
```

364

Redbooks ibm.com/redbooks

This is an example of how the new OMPCFG DD may be used. USER1.OMPROUTE(&OMPCFG) is the name of the OMPROUTE configuration file, where &OMPCFG is an MVS system symbol defined in the IEASYSYMxx parmlib member.

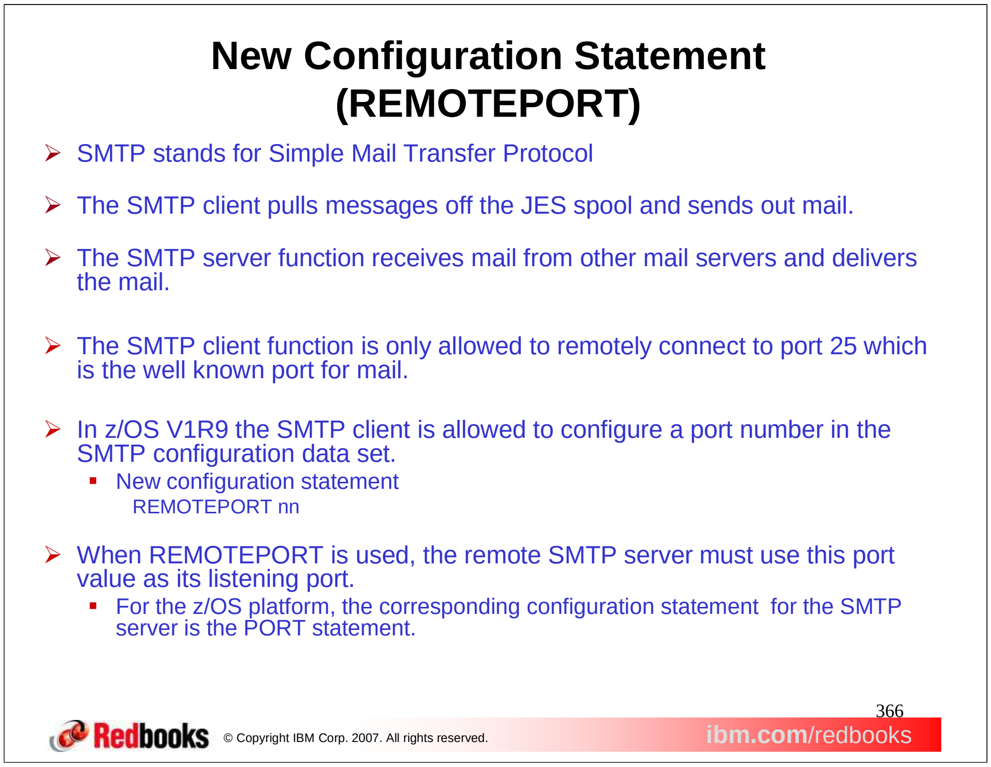


SMTP Enhancements

365

ibm.com/redbooks

This section covers the enhancements made to SMTP in z/OS V1R9.



New Configuration Statement (REMOTEPORT)

- SMTP stands for Simple Mail Transfer Protocol
- The SMTP client pulls messages off the JES spool and sends out mail.
- The SMTP server function receives mail from other mail servers and delivers the mail.
- The SMTP client function is only allowed to remotely connect to port 25 which is the well known port for mail.
- In z/OS V1R9 the SMTP client is allowed to configure a port number in the SMTP configuration data set.
  - New configuration statement
    REMOTEPORT nn
- When REMOTEPORT is used, the remote SMTP server must use this port value as its listening port.
  - For the z/OS platform, the corresponding configuration statement for the SMTP server is the PORT statement.

366

Redbooks

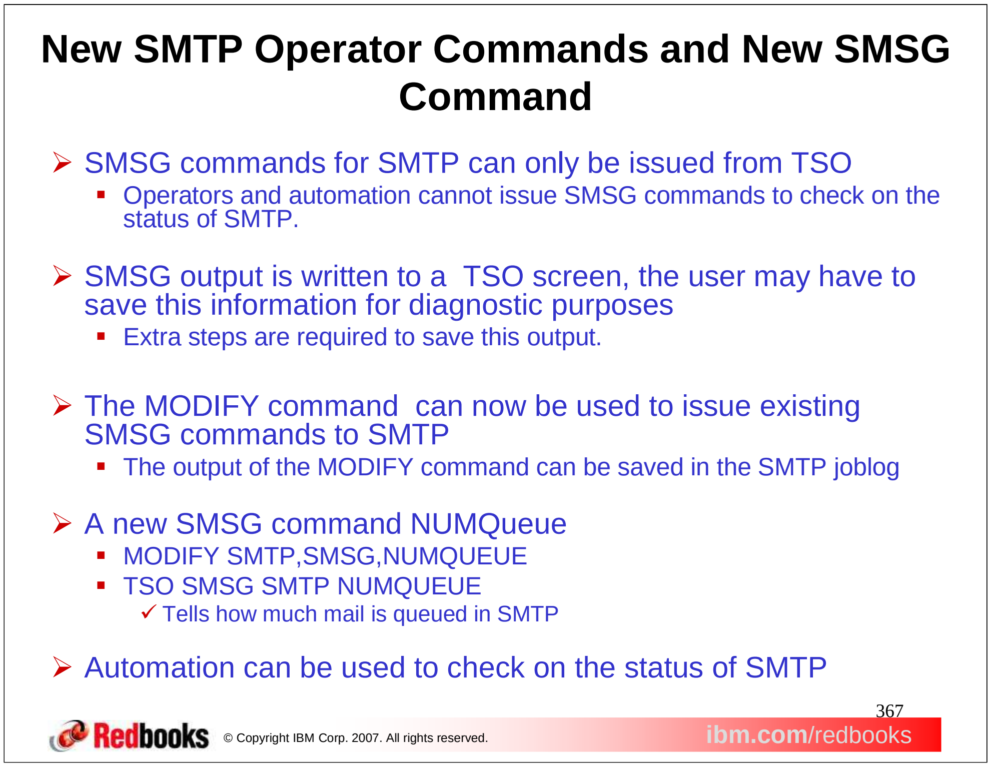
ibm.com/redbooks

SMTP stands for Simple Mail Transfer Protocol. It supports RFCs 821 and 822. SMTP is used to pull messages off the JES spool. Messages are put on the spool using IEBGENER, TSO TRANSMIT, or SMTPNOTE. The SMTP client function is used to send out the mail to either local users or to remote mail servers. The SMTP server function puts a listen up on the default port of 25 and accepts mail from other mail servers or client socket programs. It delivers the mail to local users, or relays the mail to remote mail servers. In prior releases, the SMTP client function is only allowed to remotely connect to the well known port for mail which is port 25. This is too restrictive. Users would like SMTP to support a configuration option so that the SMTP client function can remotely connect to a configured port value which the system administrator can chose. This configuration can be useful in a testing environment.
So we added support for a new configuration statement so that the SMTP client function can remotely connect to a configured port value which the system administrator can chose. The SMTP started task now supports a new configuration statement REMOTEPORT. This value will be identified at initialization time and used by the SMTP client so that this port number, and not 25, is used during connect processing. As a result, the SMTP client will use this port number as the remote port to send all outbound mail. This value may not be reset dynamically. To change it, the user must stop and restart the SMTP started task. Of course if the statement is not coded the default is port 25.

For parameters the value of the nn is a decimal number. This parameter must be within the range of 1 to 65534 and is limited to ten characters.

Note that if the REMOTEPORT statement is coded it must be within the range and there is no default taken if the statement is coded incorrectly. Furthermore, SMTP will not start.

If the statement is coded, then the SMTP client will use this port value to connect to the remote SMTP server. If no SMTP server is listening on that port then mail cannot be delivered. On the z/OS platform the corresponding configuration statement that needs to be modified for the SMTP server is the PORT statement in the SMTP configuration file on the system where SMTP server is started and the mail is to be sent. Also for z/OS platform, the SMTP server port value used should be reserved on the PORT statement in the TCP/IP configuration file as well.



New SMTP Operator Commands and New SMSG Command

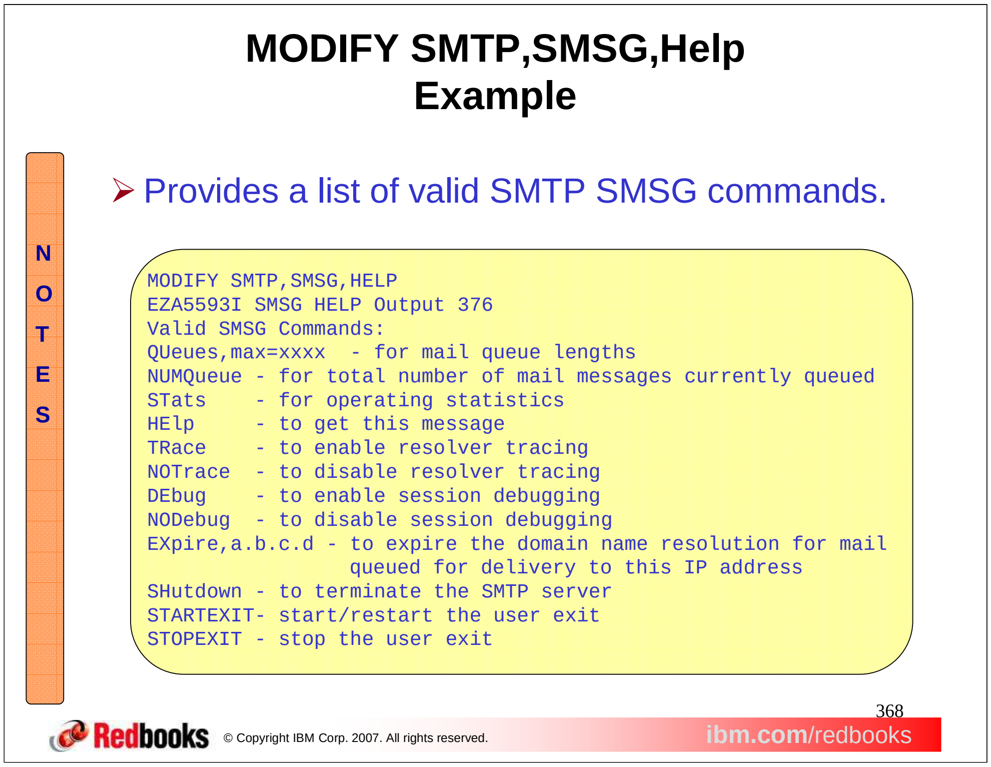
- SMSG commands for SMTP can only be issued from TSO
  - Operators and automation cannot issue SMSG commands to check on the status of SMTP.
- SMSG output is written to a TSO screen, the user may have to save this information for diagnostic purposes
  - Extra steps are required to save this output.
- The MODIFY command can now be used to issue existing SMSG commands to SMTP
  - The output of the MODIFY command can be saved in the SMTP joblog
- A new SMSG command NUMQueue
  - MODIFY SMTP,SMSG,NUMQUEUE
  - TSO SMSG SMTP NUMQUEUE
    - ✓ Tells how much mail is queued in SMTP
- Automation can be used to check on the status of SMTP

367

Redbooks ibm.com/redbooks

To monitor SMTP, it would help helpful to have automation or operators check on the status of SMTP. But, since TSO SMSG commands cannot be issued from automation or from operators (if they are not logged on to TSO), this cannot be done. Also, there is not an easy way to save the output of the TSO SMSG commands. So the customer must take extra steps to save this output.

In z/OS V1R9, we added support for a MODIFY command that can be issued by automation and operators. A new SMSG command NUMQueue was also added to tell how much mail is queued in SMTP. The new command can via the MODIFY command or TSO. The output of the Modify command is saved in the SMTP joblog. Now, automation and operators can check on the status of SMTP.



MODIFY SMTP,SMSG,Help Example

N
O
T
E
S

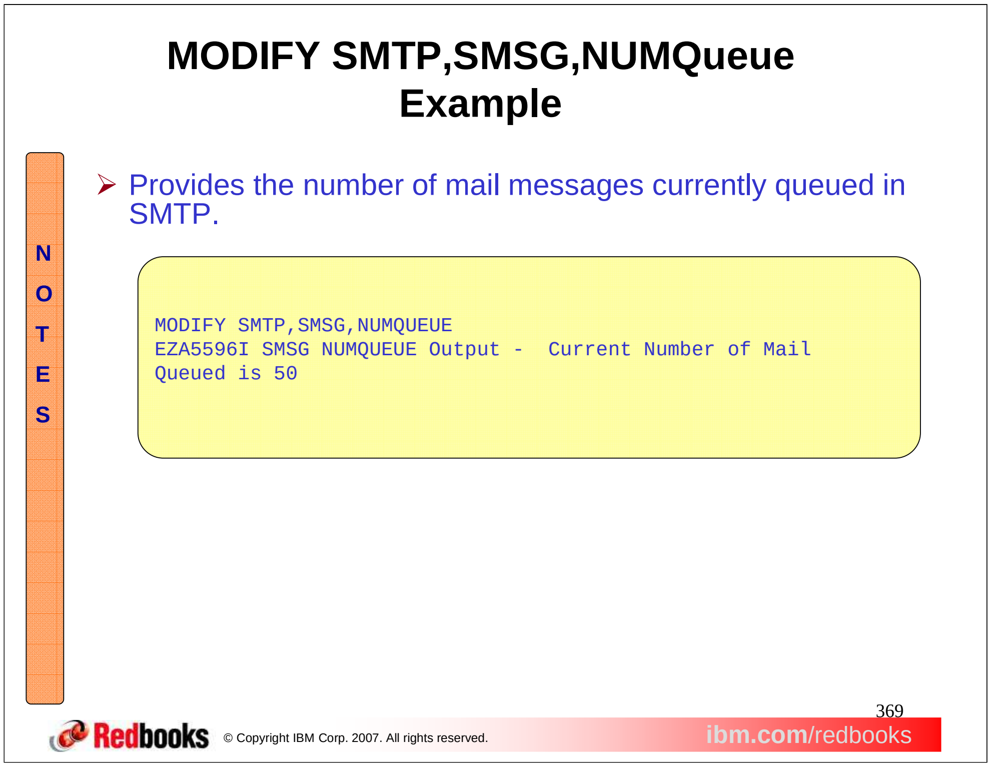
- Provides a list of valid SMTP SMSG commands.

```
MODIFY SMTP,SMSG,HELP
EZA5593I SMSG HELP Output 376
Valid SMSG Commands:
QUeues,max=xxxx  - for mail queue lengths
NUMQueue - for total number of mail messages currently queued
STats    - for operating statistics
HElp     - to get this message
TRace    - to enable resolver tracing
NOTrace  - to disable resolver tracing
DEbug    - to enable session debugging
NODebug  - to disable session debugging
EXpire,a.b.c.d - to expire the domain name resolution for mail
                 queued for delivery to this IP address
SHutdown - to terminate the SMTP server
STARTEXIT- start/restart the user exit
STOPEXIT - stop the user exit
```

368

Redbooks ibm.com/redbooks

The MODIFY SMTP,SMSG,HELP lists the valid MODIFY SMTP SMSG commands.

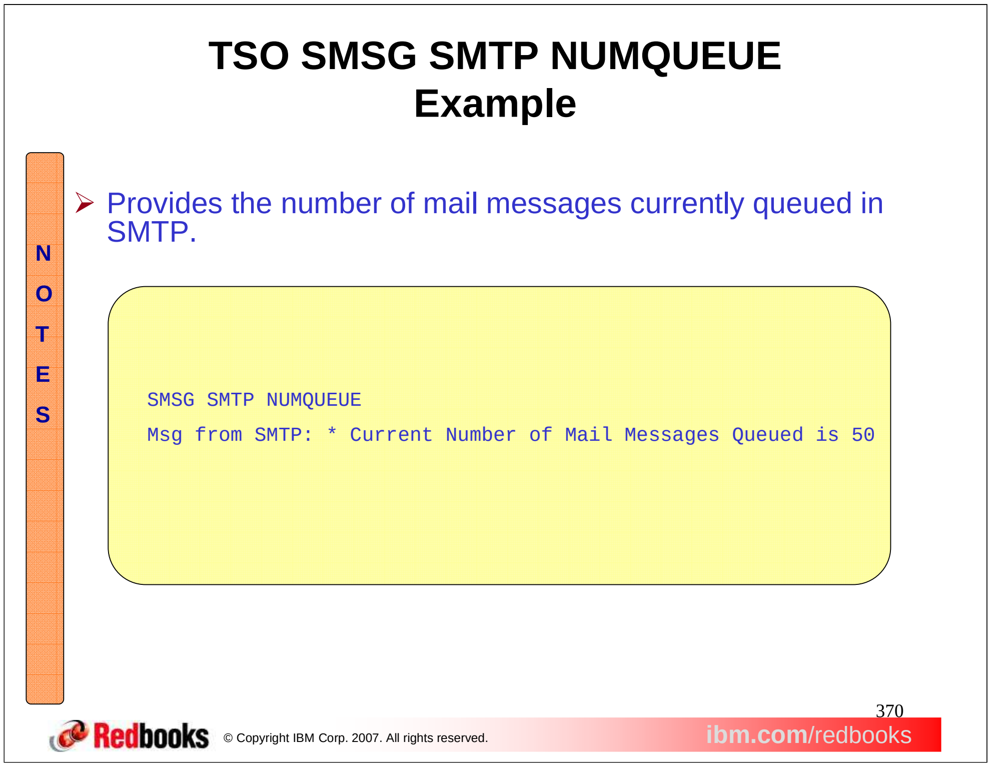


MODIFY SMTP,SMSG,NUMQueue Example

- Provides the number of mail messages currently queued in SMTP.

```
MODIFY SMTP,SMSG,NUMQUEUE
EZA5596I SMSG NUMQUEUE Output -  Current Number of Mail
Queued is 50
```

The MODIFY SMTP,SMSG,NUMQUEUE command provides the current number of mail messages queued in SMTP.



TSO SMSG SMTP NUMQUEUE Example

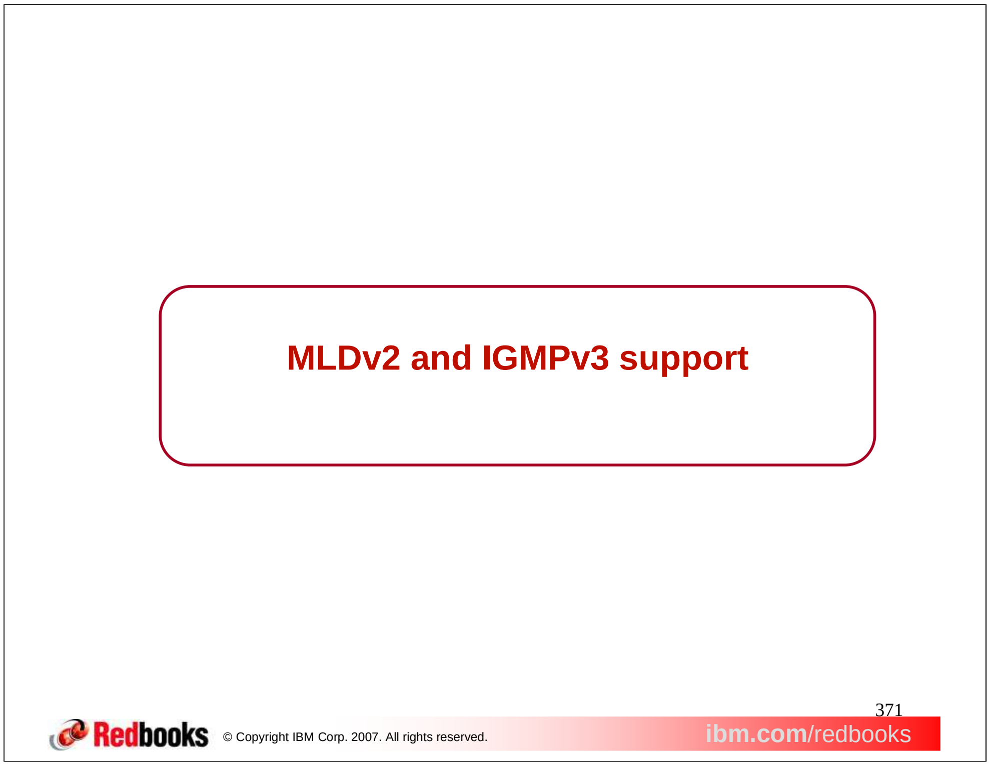
NOTES

- Provides the number of mail messages currently queued in SMTP.

```
SMSG SMTP NUMQUEUE
Msg from SMTP: * Current Number of Mail Messages Queued is 50
```

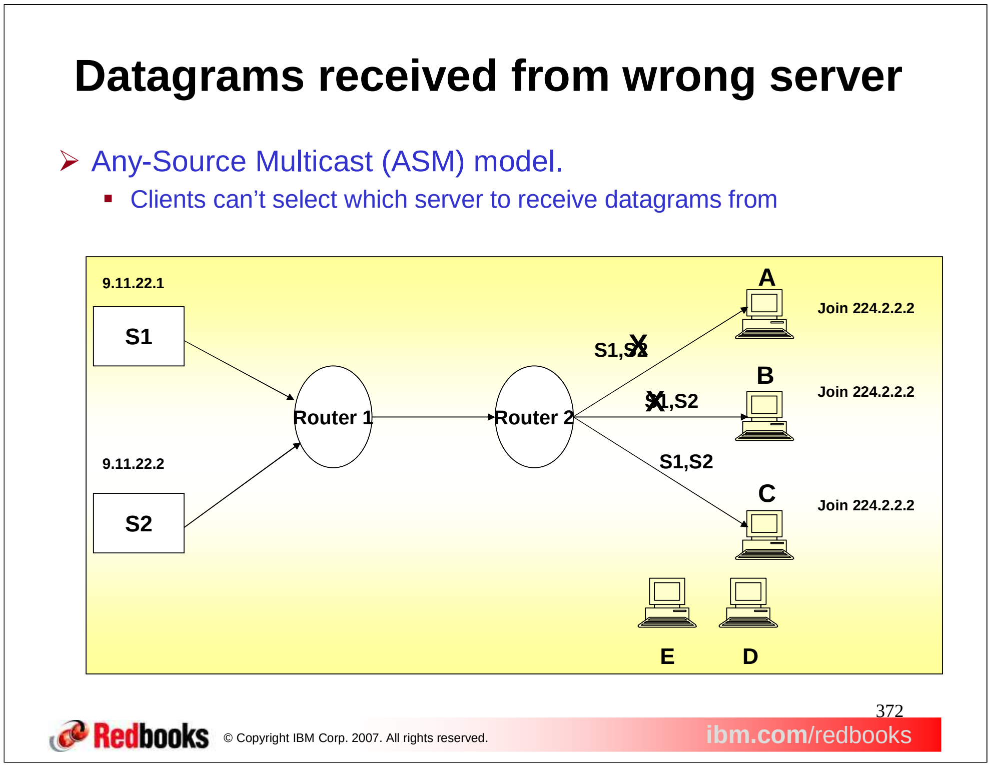
Redbooks ibm.com/redbooks

There is a new SMSG command, NUMQUEUE, that can be issued from TSO to provide the current number of mail messages queued in SMTP.



MLDv2 and IGMPv3 support

This section covers the implementation of the new versions of the Internet Group Management Protocol for IPv4 , IGMPv3, and Multicast Listening Discovery for IPv6, MLDv2.



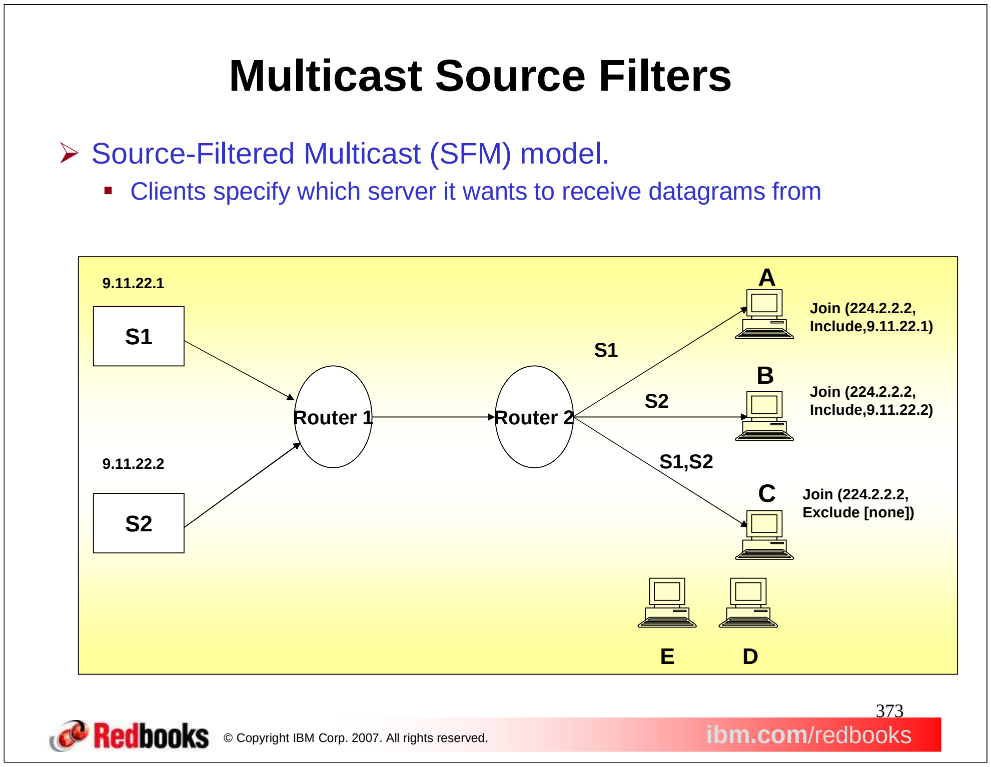
Datagrams received from wrong server

- Any-Source Multicast (ASM) model.
  - Clients can't select which server to receive datagrams from

The current multicast model is referred to as any-source multicast. Multicast server programs send out datagrams using a multicast address as the destination address. Any client program on the network can choose to receive the multicast datagrams by joining the multicast group. This means that a client program which has joined a multicast group will receive multicast datagram's from any server, regardless of the source IP address. In the example, clients A, B and C receive all datagram's from both server 1 and server 2.

An application could specify which multicast datagrams it wanted to receive by specifying the multicast address as the filter. However all multicast datagrams which met that criteria, regardless of the source address, would be delivered to the application. This is referred to as the Any-Source Multicast (ASM) model.   In the diagram, servers S1 and S2 send datagrams to the multicast group address 224.2.2.2. Clients A, B and C receive datagrams from servers S1 and S2 due to joining the multicast group address 224.2.2.2. Client D and and E do not receive any datagrams from S1 and S2 since they did not join the multicast group address 224.2.2.2.

It's possible for multiple multicast servers to be sending out different information using the same destination multicast address. With the any-source multicast model, this can cause problems if a client only wants to receive datagram's from a specific server. For example, if multiple servers are sending out different audio feeds, a client may only want to receive audio from one source. In the example, client A wants to receive datagram's from server 1 but does not want to receive datagrams from server 2 and client B wants to receive datagram's from server 2 but not server 1.



Multicast Source Filters

- Source-Filtered Multicast (SFM) model.
  - Clients specify which server it wants to receive datagrams from

9.11.22.1

S1

9.11.22.2

S2

Router 1

Router 2

S1

S2

S1,S2

A

Join (224.2.2.2, Include,9.11.22.1)

B

Join (224.2.2.2, Include,9.11.22.2)

C

Join (224.2.2.2, Exclude [none])

E

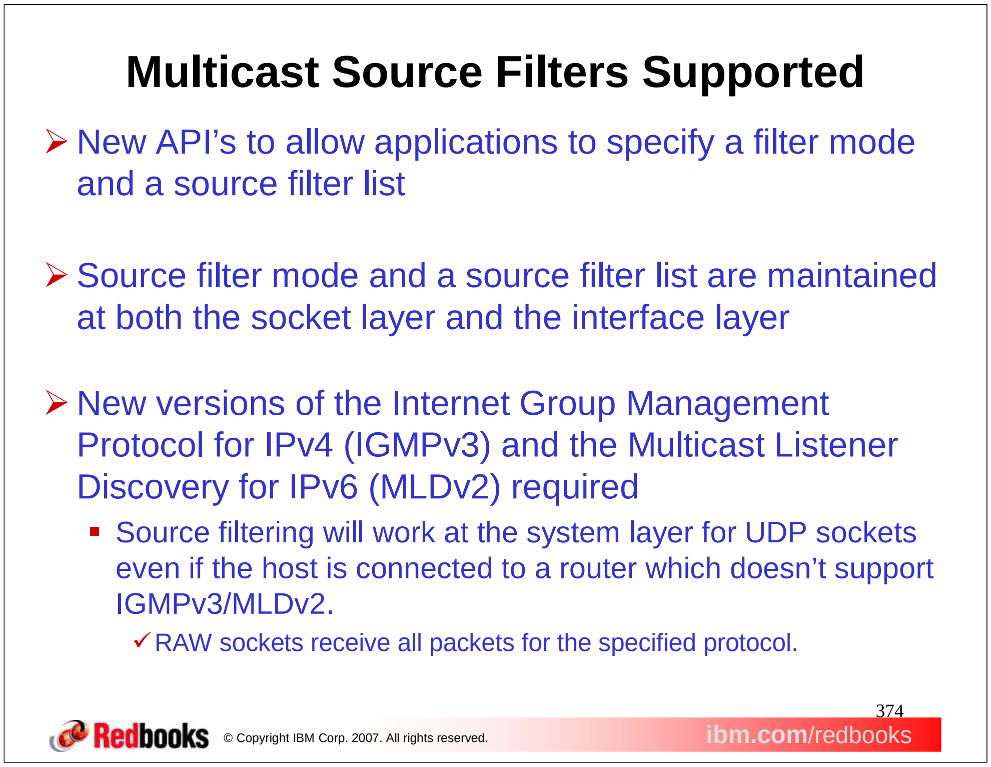
D

373

ibm.com/redbooks

The solution is to allow a client to specify filters based on the source IP address of the multicast datagram. This model is called source-filtered multicast. In the example, client A can specify that it only wants multicast datagram’s from server 1 and client B only wants to receive multicast datagram’s from server 2. Client C wants to receive all multicast datagram’s, regardless of the source IP address.

The SFM model allows an application to filter the datagrams it receives based on the source IP address. In the diagram servers S1 and S2 send datagrams to the multicast group address 224.2.2.2. Client A receives datagrams from server S1 only due to joining the multicast group address 224.2.2.2 with the source filter mode of INCLUDE for the source IP address 9.11.22.1. Client B receives datagrams from server S2 only due to joining the multicast group address 224.2.2.2 with the source filter mode of INCLUDE for the source IP address 9.11.22.2. Client C receives datagrams from servers S1 and S2 due to joining the multicast group address 224.2.2.2 with the source filter mode of EXCLUDE with an empty source list. Clients D and E do not receive any datagram from S1 and S2 because they did not join the multicast group address 224.2.2.2.



Multicast Source Filters Supported

- New API’s to allow applications to specify a filter mode and a source filter list
- Source filter mode and a source filter list are maintained at both the socket layer and the interface layer
- New versions of the Internet Group Management Protocol for IPv4 (IGMPv3) and the Multicast Listener Discovery for IPv6 (MLDv2) required
  - Source filtering will work at the system layer for UDP sockets even if the host is connected to a router which doesn’t support IGMPv3/MLDv2.
    - ✓ RAW sockets receive all packets for the specified protocol.

New API’s are required to allow the client programs to specify a source filter list and what is referred to as a filter mode. There are two types of API’s for multicast source filtering. Delta based or Basic adds or deletes to the source list, can have only one entry on a single call and does not allow changing the filter mode. Full state or Advanced allows the full replacement of a source filter list and the filter mode on a single call. This support is available on the following APIs: z/OS Language Environment C/C++, Macro – EZASMI, z/OS UNIX System Services: Assembler Callable Services, Callable – EZASOKET, CICS, and REXX. Please refer to the appropriate documentation for more details about Multicast Source Filter APIs.

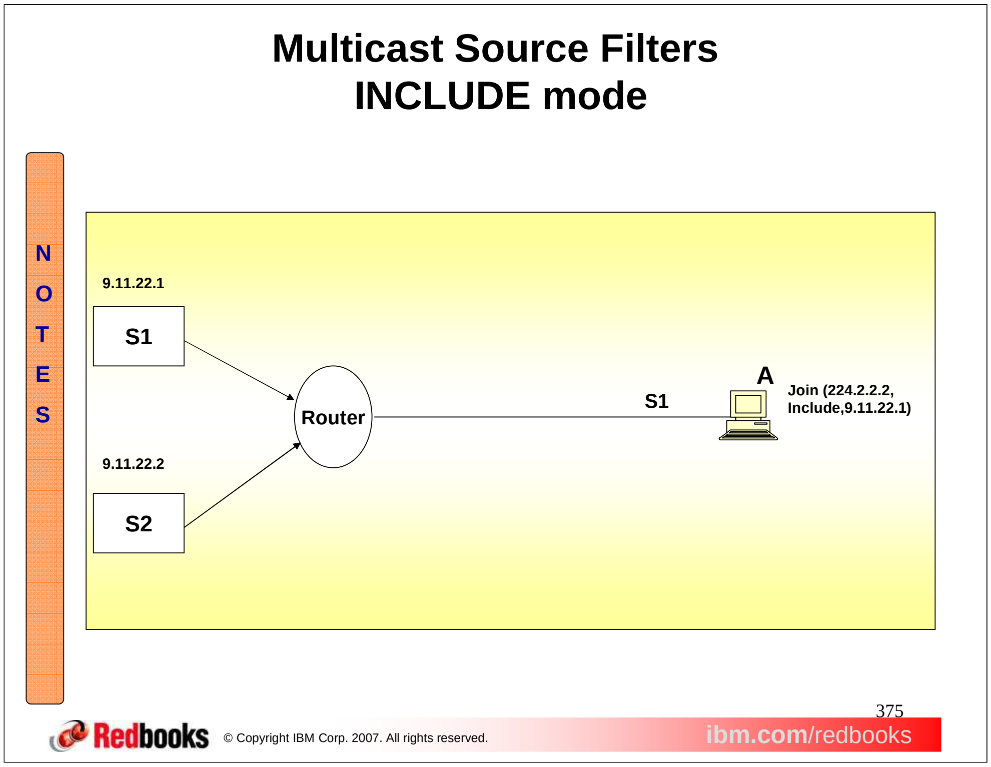
The filter mode can be either INCLUDE or EXCLUDE. With INCLUDE mode, a client specifies which multicast datagram’s they want to receive, based on the source IP address. With EXCLUDE mode, a client specifies which multicast datagram’s they don’t want to receive, based on the source IP address.

The source filter mode and a source filter list are maintained at both the socket layer and the interface layer. At the socket layer, the source filter mode and the source list reflect what was specified by the application. The filter mode and the source filter list for the interface is derived from all socket layer filter modes and source filter lists which have joined a multicast group for the interface. Information at the interface layer is what is reported to multicast routers. This allows the multicast routers to determine which multicast datagrams to forward.

The interface layer filter mode is determined by two rules; 1) If all socket layer filter modes are INCLUDE, the interface layer filter mode is INCLUDE. 2) If any socket layer filter mode is EXCLUDE, the interface layer filter mode is EXCLUDE.

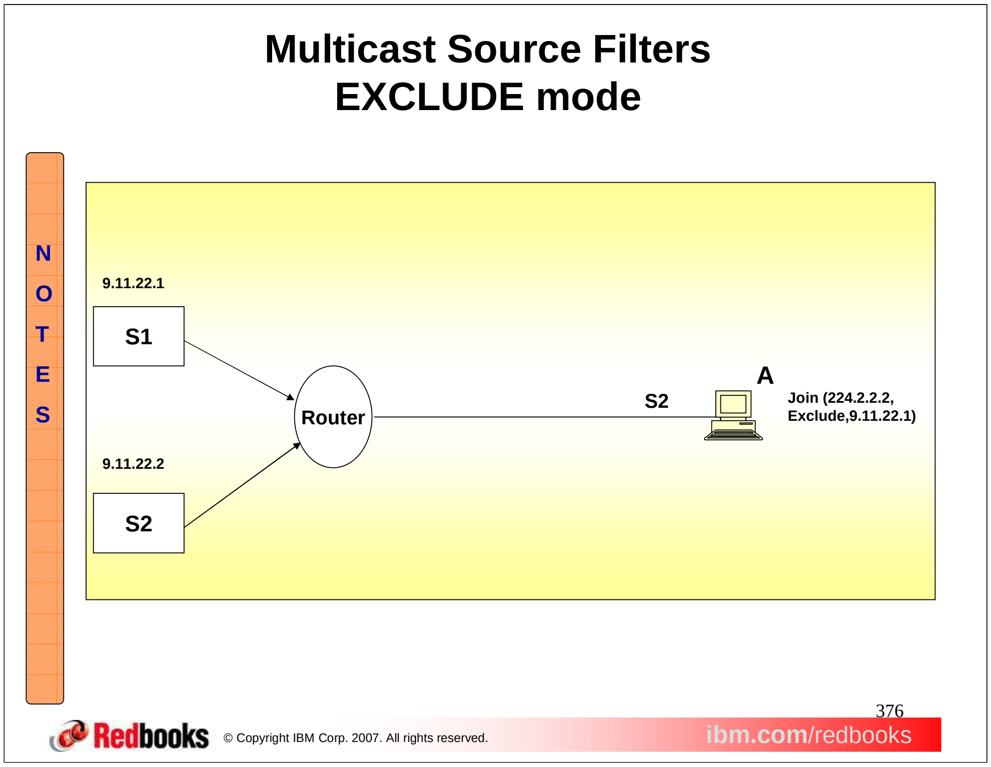
The interface layer source filter list is determined by two rules; 1) If all socket layer filter modes are INCLUDE, the interface’s source filter list is the union of all source filter addresses. 2) If any socket layer filter mode is EXCLUDE, the interface’s source filter list is derived by taking the intersection of all EXCLUDE mode lists minus any IP addresses in any INCLUDE mode lists.

New versions of IGMP and MLD are required to communicate the new filtering information to multicast routers. Note that this solution also allows the local system to filter on source addresses even if the system is not attached to a multicast router which supports source address filtering.
RFC3678 defines new socket options and functions to manage source filters. RFC 3376 and 3810 define the IGMPv3 (IPv4) and MLDv2 (IPv6) protocols used by systems to report their IP multicast group memberships to neighboring multicast routers. With the new versions of the protocols, multicast routers are informed of the source IP filtering of any applications on a system. This allows the multicast router to send only multicast datagrams, which the system has applications interested in receiving. z/OS communications Server does not support any multicast routing protocols and only supports source filtering for user datagram protocol (UDP) sockets.



Multicast Source Filters
INCLUDE mode

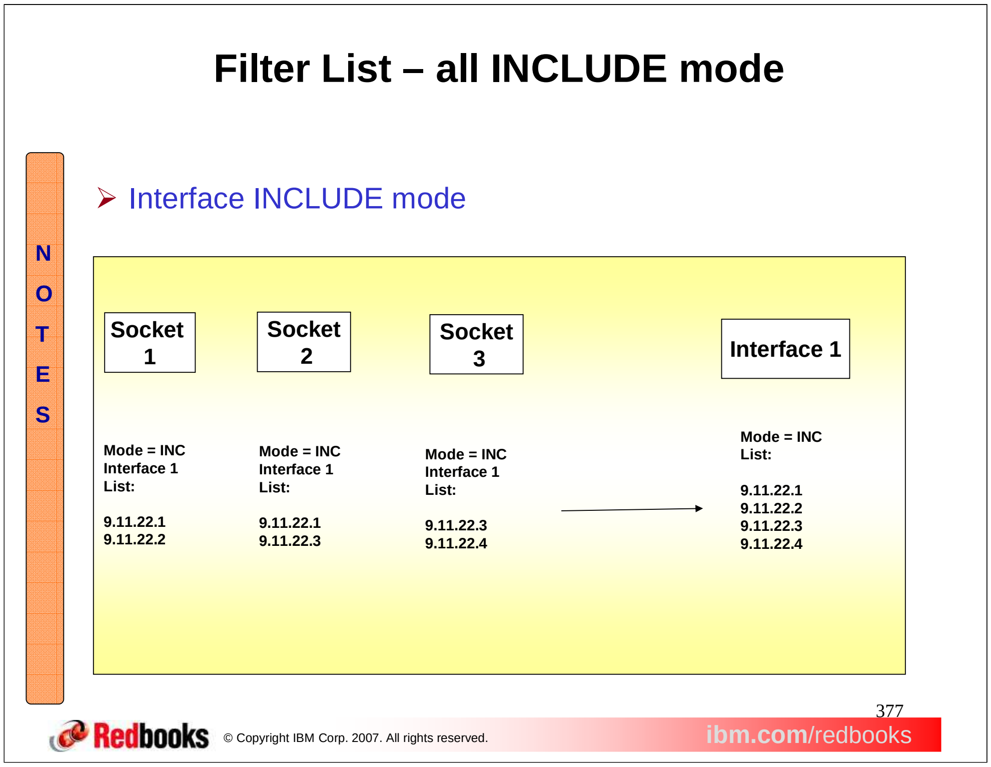
In this example, client A specifies it wants to receive multicast datagram's from server 1 only. All multicast datagram's from server 2 are dropped by the router.



Multicast Source Filters
EXCLUDE mode

Redbooks ibm.com/redbooks

In this example, client A specifies it doesn’t want to receive multicast datagram’s from server 1. All multicast datagram’s from server 1 are dropped by the router.



Filter List – all INCLUDE mode

N O T E S

- Interface INCLUDE mode

Socket 1

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.2

Socket 2

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.3

Socket 3

Mode = INC
Interface 1
List:

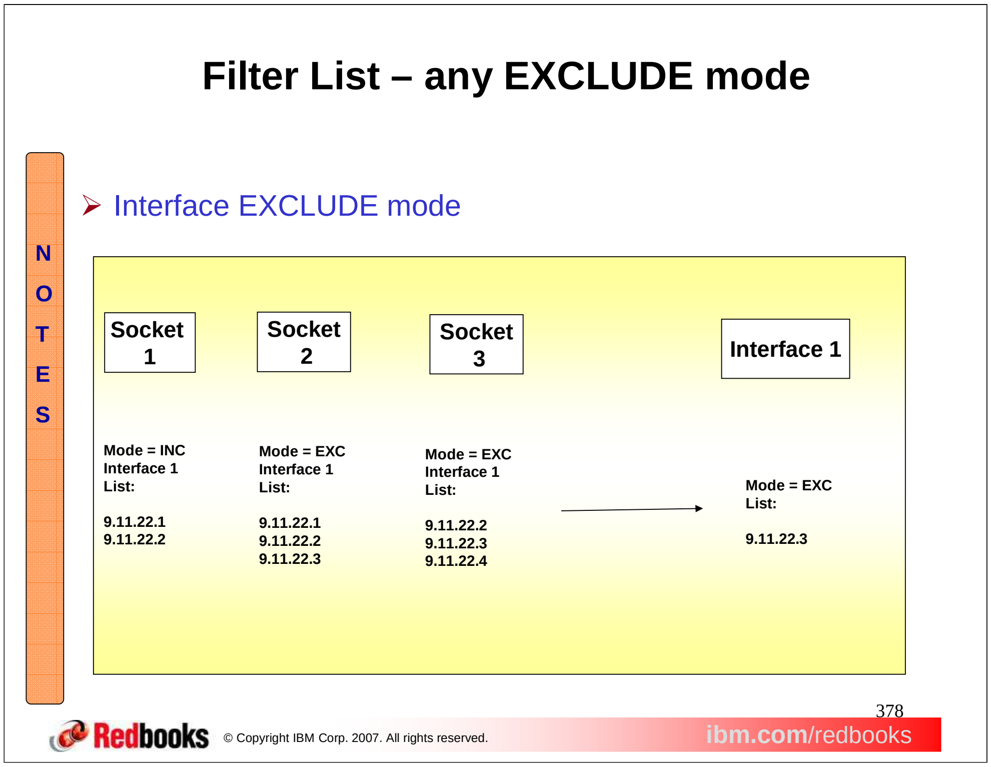
9.11.22.3
9.11.22.4

Interface 1

Mode = INC
List:

9.11.22.1
9.11.22.2
9.11.22.3
9.11.22.4

In this example, all three sockets have specified INCLUDE mode. Therefore the interface layer’s filter mode is also INCLUDE and it’s source filter list is the union of the socket layer’s source filter list.



Filter List – any EXCLUDE mode

- Interface EXCLUDE mode

Socket 1

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.2

Socket 2

Mode = EXC
Interface 1
List:

9.11.22.1
9.11.22.2
9.11.22.3

Socket 3

Mode = EXC
Interface 1
List:

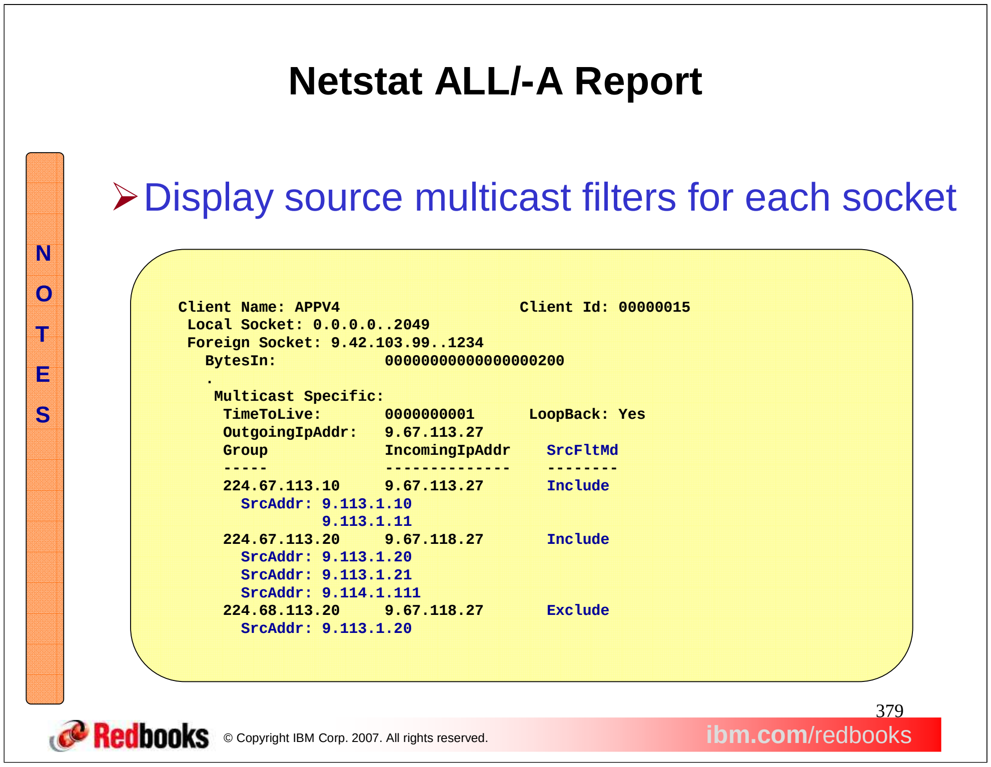
9.11.22.2
9.11.22.3
9.11.22.4

Interface 1

Mode = EXC
List:

9.11.22.3

In this example, socket 1 specified a filter mode of INCLUDE and sockets 2 and 3 specified a filter mode of EXCLUDE. The interface layer mode is therefore EXCLUDE. The source filter list is derived by first taking the intersection of socket 2 and socket 3's lists. In this case the intersection is 9.11.22.2 and 9.11.22.3. Then any addresses which are in socket 1's source list are deleted. In this case 9.11.22.2 is in the list so it is deleted from the interface list. The interface layer therefore has a filter mode of EXCLUDE and a source filter list of 9.11.22.3.



Netstat ALL/-A Report

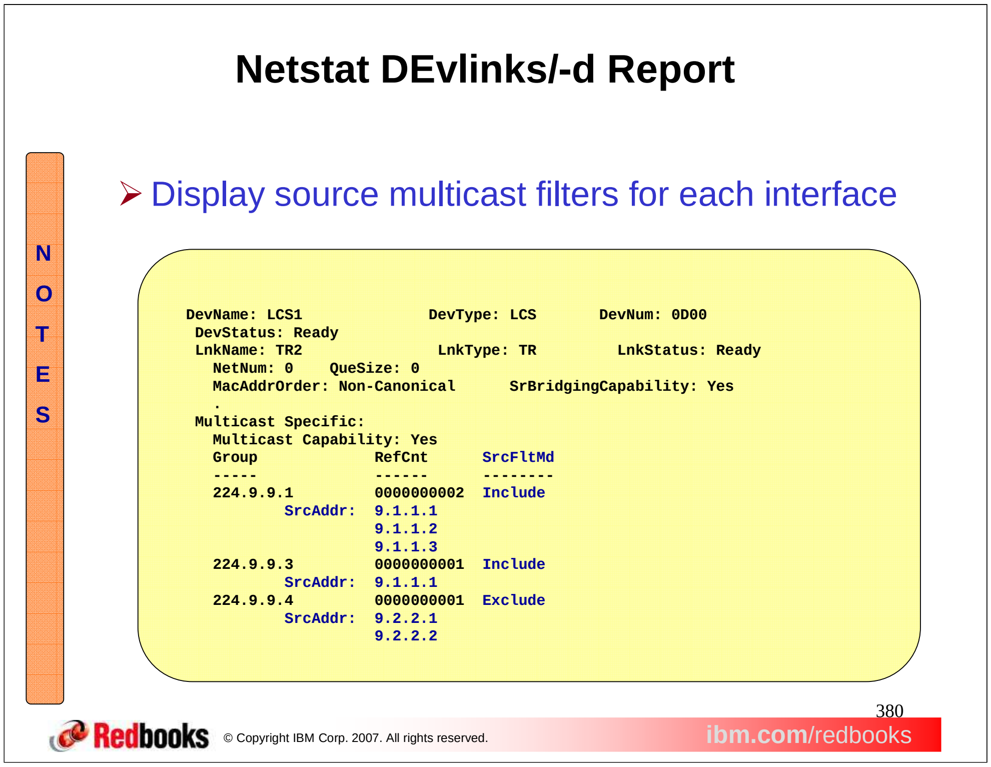
N O T E S

- Display source multicast filters for each socket

```
Client Name: APPV4                    Client Id: 00000015
 Local Socket: 0.0.0.0..2049
 Foreign Socket: 9.42.103.99..1234
   BytesIn:            00000000000000000200
    .
    Multicast Specific:
     TimeToLive:       0000000001      LoopBack: Yes
     OutgoingIpAddr:   9.67.113.27
     Group             IncomingIpAddr    SrcFltMd
     -----             --------------    --------
     224.67.113.10     9.67.113.27       Include
       SrcAddr: 9.113.1.10
                9.113.1.11
     224.67.113.20     9.67.118.27       Include
       SrcAddr: 9.113.1.20
       SrcAddr: 9.113.1.21
       SrcAddr: 9.114.1.111
     224.68.113.20     9.67.118.27       Exclude
       SrcAddr: 9.113.1.20
```

379

The netstat ALL/-A report has been updated to include the source filter mode and source filter list.



Netstat DEvlinks/-d Report

N O T E S

- Display source multicast filters for each interface

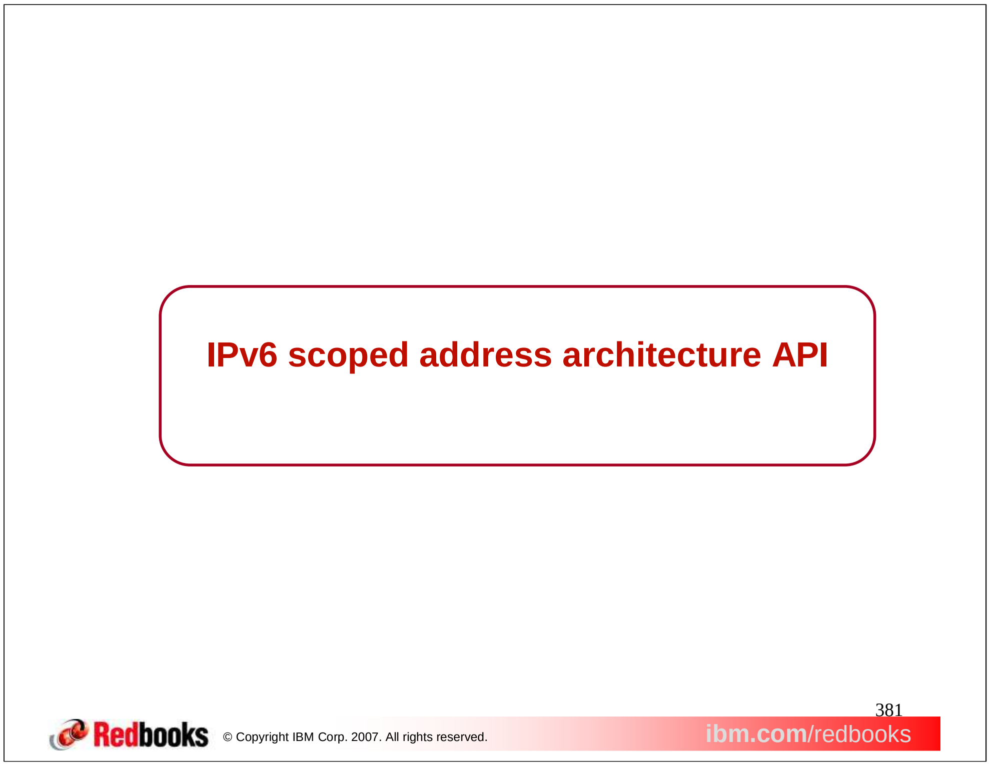
```
DevName: LCS1            DevType: LCS       DevNum: 0D00
 DevStatus: Ready
 LnkName: TR2             LnkType: TR         LnkStatus: Ready
   NetNum: 0    QueSize: 0
   MacAddrOrder: Non-Canonical      SrBridgingCapability: Yes
   .
 Multicast Specific:
   Multicast Capability: Yes
   Group            RefCnt      SrcFltMd
   -----            ------      --------
   224.9.9.1        0000000002  Include
           SrcAddr: 9.1.1.1
                    9.1.1.2
                    9.1.1.3
   224.9.9.3        0000000001  Include
           SrcAddr: 9.1.1.1
   224.9.9.4        0000000001  Exclude
           SrcAddr: 9.2.2.1
                    9.2.2.2
```

380

Redbooks

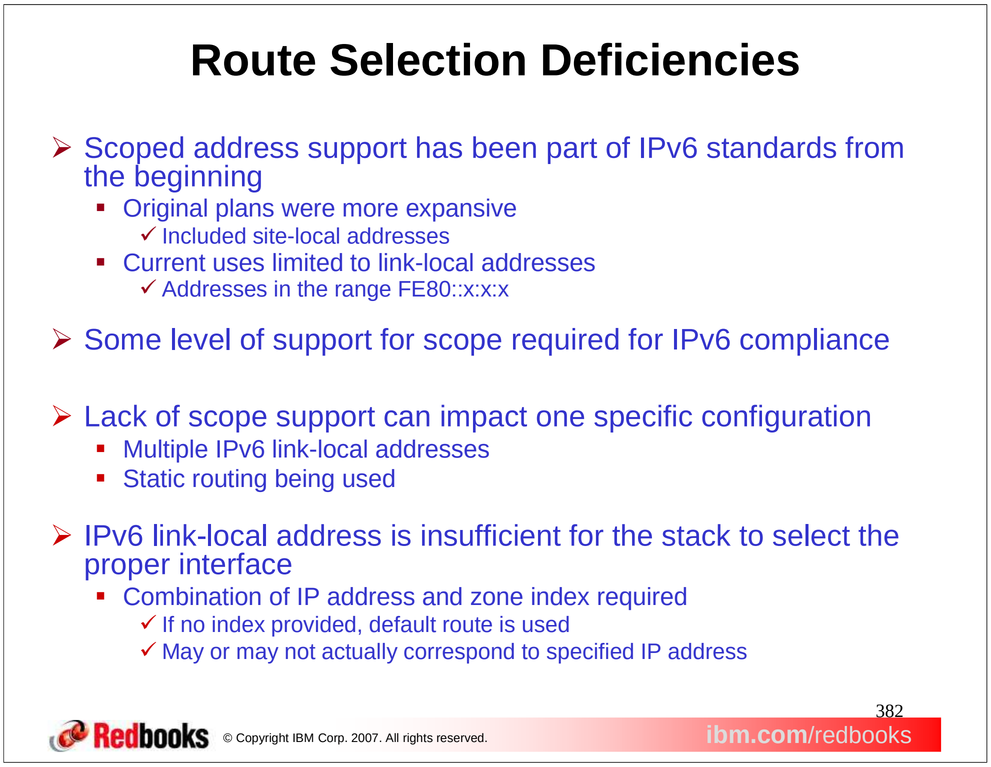
ibm.com/redbooks

The netstat DEVLINKS/-d report has been updated to display the source filter mode and the source filter lists.



IPv6 scoped address architecture API

This section discusses the support added to z/OS Communications Server for the IPv6 scoped address API changes, primarily to the Resolver Getaddrinfo and Getnameinfo functions.



Route Selection Deficiencies

- Scoped address support has been part of IPv6 standards from the beginning
  - Original plans were more expansive
    - ✓ Included site-local addresses
  - Current uses limited to link-local addresses
    - ✓ Addresses in the range FE80::x:x:x
- Some level of support for scope required for IPv6 compliance
- Lack of scope support can impact one specific configuration
  - Multiple IPv6 link-local addresses
  - Static routing being used
- IPv6 link-local address is insufficient for the stack to select the proper interface
  - Combination of IP address and zone index required
    - ✓ If no index provided, default route is used
    - ✓ May or may not actually correspond to specified IP address

IPv6 has always had the concept of scoped addresses, but z/OS has never fully supported the function. Part of that was due to the cost of fully implementing support for scoped addresses (sometimes also referred to as zones), but also because the concept, while present, had not been fully defined. For instance, at one point scoped addresses included a set of addresses defined as site-local addresses, but that class of addresses has since been downplayed by the IPv6 standards. The only addresses left for which the concept of scope now applies are link-local addresses.

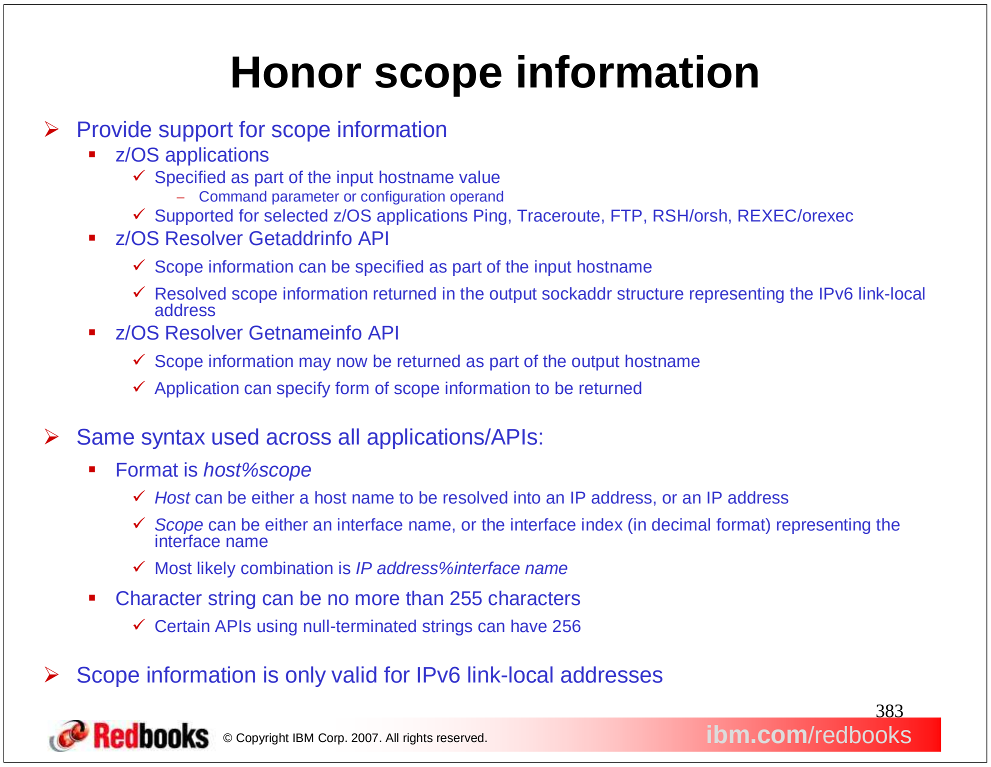
Link-local addresses are addresses that refer to only a particular physical link, or the physical network directly attached to that link (e.g., LAN). They are used only for local communication on that physical link, and routers are designed to not forward datagrams that use link-local addresses. Link-local addresses are typically dynamically assigned by the TCP/IP stack, and are mostly used for so-called “bootstrap” functions or diagnostic purposes.

In order to maintain compliance with IPv6 standards, z/OS needs to implement some additional level of scoped address support. The level chosen could range from full implementation of zones, to recognizing and utilizing scope information on the various z/OS applications and APIs.

There is one configuration in particular where the absence of support for scoped addresses, at the API level, could impact a z/OS user. The situation would involve a configuration where multiple IPv6 link-local addresses have been dynamically assigned. To further complicate matters, static routing is being utilized in the configuration. There are other possibilities where even the use of dynamic routing could lead to complications, but the more likely situation is that static routing is being utilized.

In such a situation, to successful route data over a given IPv6 link-local address, both the address and the zone index value need to be specified. The zone index is a value assigned by the stack to represent the correct entry (or interface) in the routing table. If the zone index is not present, then the stack uses the “default route” for this configuration. If the default route uses the interface that matches the IPv6 link-local address that was specified, everything works just fine. If, however, the default route does not use the correct interface for the specified IPv6 link-local address, then a routing error is encountered and the application request fails or times out.

Some applications, such as Ping and Traceroute, have had parameters defined that allow them to work around this problem, but other applications such as FTP have no such work-around. Ideally, here would be a standard mechanism in place to allow the z/OS user or the application to specify the proper interface to use with link-local addresses.



Honor scope information

- Provide support for scope information
  - z/OS applications
    - ✓ Specified as part of the input hostname value
      - – Command parameter or configuration operand
    - ✓ Supported for selected z/OS applications Ping, Traceroute, FTP, RSH/orsh, REXEC/orexec
  - z/OS Resolver Getaddrinfo API
    - ✓ Scope information can be specified as part of the input hostname
    - ✓ Resolved scope information returned in the output sockaddr structure representing the IPv6 link-local address
  - z/OS Resolver Getnameinfo API
    - ✓ Scope information may now be returned as part of the output hostname
    - ✓ Application can specify form of scope information to be returned
- Same syntax used across all applications/APIs:
  - Format is *host%scope*
    - ✓ *Host* can be either a host name to be resolved into an IP address, or an IP address
    - ✓ *Scope* can be either an interface name, or the interface index (in decimal format) representing the interface name
    - ✓ Most likely combination is *IP address%interface name*
  - Character string can be no more than 255 characters
    - ✓ Certain APIs using null-terminated strings can have 256
- Scope information is only valid for IPv6 link-local addresses

383

Redbooks ibm.com/redbooks

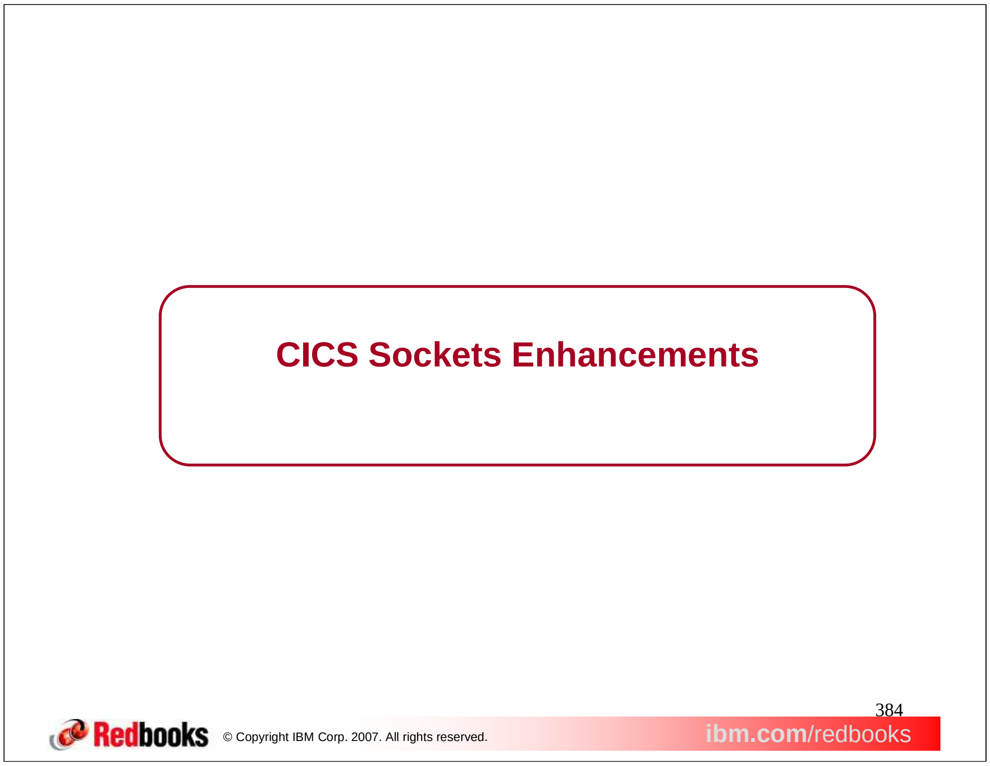
The generalized solution is to permit scope information to be present as part of the hostname parameter or configuration operand. This support is extended to a subset of the z/OS applications listed on the slide. The z/OS Resolver was also updated to process the scope information correctly.

The underlying basis for providing support of scope information is the z/OS Resolver updates for Getaddrinfo, since the z/OS applications that support scope information will issue Getaddrinfo under the covers to manipulate the scope information. The scope information will only be processed by Getaddrinfo when the input hostname is either an IPv6 link-local address, or when the hostname provided resolves into one or more IPv6 link-local addresses. Be careful with the latter situation however, especially if for some reason you have configured one hostname to represent multiple link-local addresses. The Resolver has no way of determining which of the resolved link-local addresses really is represented by the input scope information, so by default the Resolver will apply the scope information to EVERY resolved link-local address. Typically this will result in only one output sockaddr structure being correct, so it is best to stick to one link-local address per hostname, if you even bother to use hostnames of link-local addresses. Assuming there is an IPv6 link-local address in play, the Resolver will handle the scope information in one of two ways: If the input scope information is an interface name, then the Resolver issues a system IOCTL to acquire the routing table. A simple lookup is performed to find the specified interface name, and the corresponding interface index value is returned as the *sin6_scope_id* value in the output sockaddr structure. If the input scope information is an interface index, the Resolver will perform a sanity check to ensure that the index works on this system. The same IOCTL is issued to get the same routing table, but the lookup is performed to find the index value in the table, not the name. If the index is present, we will echo the input value in the sockaddr structure as described above. This sanity check is performed to ensure, as much as possible, that any sockaddr structure returned by the Resolver to the user is valid for use in establishing a connection or for sending data to the target host. If the Resolver attempts resolution but the lookup fails, the Resolver call fails.

The other Resolver API that manipulates scope information is Getnameinfo. Getnameinfo processing will take the scope information in the input sockaddr structure, namely the *sin6_scope_id* field, and append it to the end of the output hostname value. The appended scope information is returned using the same syntax discussed earlier for Getaddrinfo. The appending of the scope information is only performed for IPv6 link-local addresses, when the input sockaddr sin6_scope_id field is non-zero, and when the Getnameinfo caller passes an output buffer to be used for the hostname. There is no switch or mechanism for preventing Getnameinfo from returning this information --- if all the correct conditions are met, then scope information gets appended to the hostname. This means that you may see scope information appearing in displays or in diagnostics if your logic invokes Getnameinfo with this set of conditions, even if you hadn't overtly intended to use scope information. While the user cannot specify on the Getnameinfo call if scope is appended or not, the user does have an option on how the scope information is formatted. A new flag, NI_NUMERICSCOPE, can be used to indicate that instead of getting the default format (interface name), the user would rather get the numeric form of scope, namely the interface index. The same system IOCTL that Getaddrinfo uses to acquire the system routing table is used by Getnameinfo if a lookup of the interface name is required to append the scope information. However, if the numeric form of scope is to be returned, there is no validity checking performed to verify that the value in *sin6_scope_id* is a valid index in this system. This is different philosophy from Getaddrinfo processing, where we did look up the numeric value, even if no translation was required. The assumption with Getnameinfo is that the zone index was stored into the sockaddr structure by a trusted component, for instance the stack, and so the likelihood of the index being incorrect is smaller. Also, the impact of an incorrect index as output from Getnameinfo is less than the impact of Getaddrinfo returning garbage in the *sin6_scope_id* field, so added validation by Getnameinfo is not necessary. If we do have to resolve the zone index into an interface name, and the resolution is unsuccessful (i.e., there is no zone index of that value in the routing table), the Getnameinfo request fails.

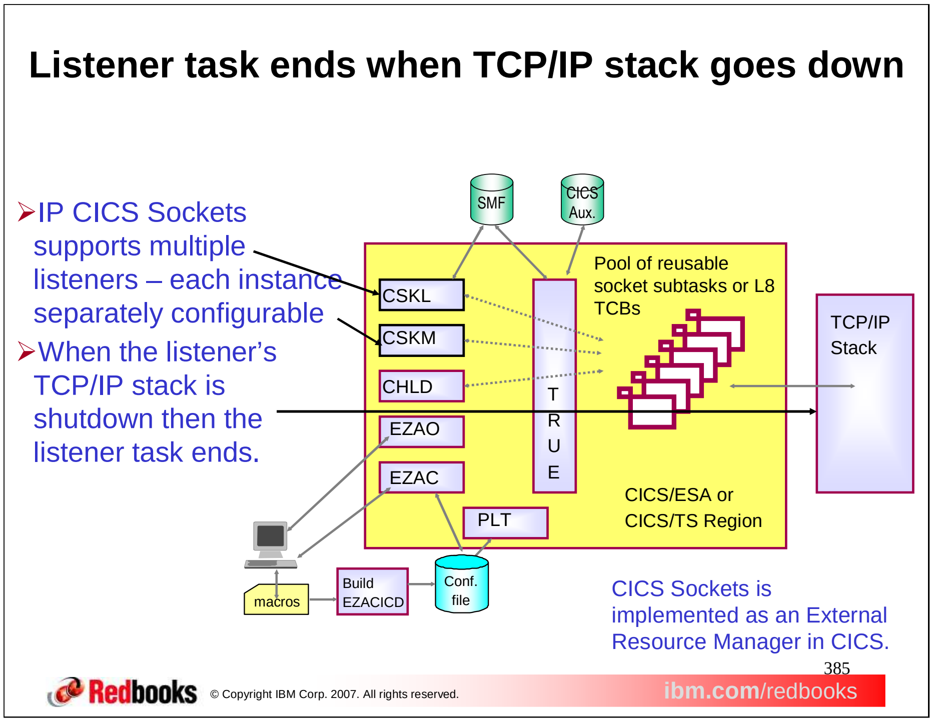
In any instance where scope information is permitted to be specified, the same syntax is used. The scope information is appended to the hostname, with a percent sign used as the delimiter value. Note that when we say "hostname" in this discussion, we mean one of two things: An actual hostname defined to DNS that maps to an IP address or an IP address. The IP address specified, or resolved to using the hostname, must be an IPv6 link-local address, or scope information is meaningless. Likewise, in this discussion, "scope information" can take on two distinct forms: The name assigned to the interface (physical link) by the user on the INTERFACE statement or the zone index assigned to the interface by the stack. The value must be specified in decimal form, not hexadecimal. While any combination of host name or IP address with interface name or interface index is permitted, in general, the most likely choice would be IP address with interface name. In most situations, link-local addresses would not have a hostname assigned to them, leaving the IP address as the only choice. The zone index value for a given interface can only be acquired programmatically, not via operator displays, and in any event can change for an interface from one TCP/IP stack activation to the next. The interface name, on the other hand, is likely to be constant and can be displayed (along with the associated link-local address) using Netstat commands, so interface name is much more accessible than the zone index. We will come back to this discussion later in the presentation. In order to minimize the impact of the addition of scope information to the z/OS APIs, the existing restriction of 255 characters (or 256, for APIs that utilize null-termination characters) for a "hostname" has been maintained. This was not believed to be a concern because (a) most hostnames are far less than 255 characters long and (b) IPv6 link-local addresses typically would not have a hostname assigned to them anyway, since they are dynamically assigned to an interface by the stack.

The full range of z/OS IPv6 capable APIs that provide support for Getaddrinfo and Getnameinfo calls are capable of handling scope information



CICS Sockets Enhancements

This section describes the enhancements to IP CICS Sockets.



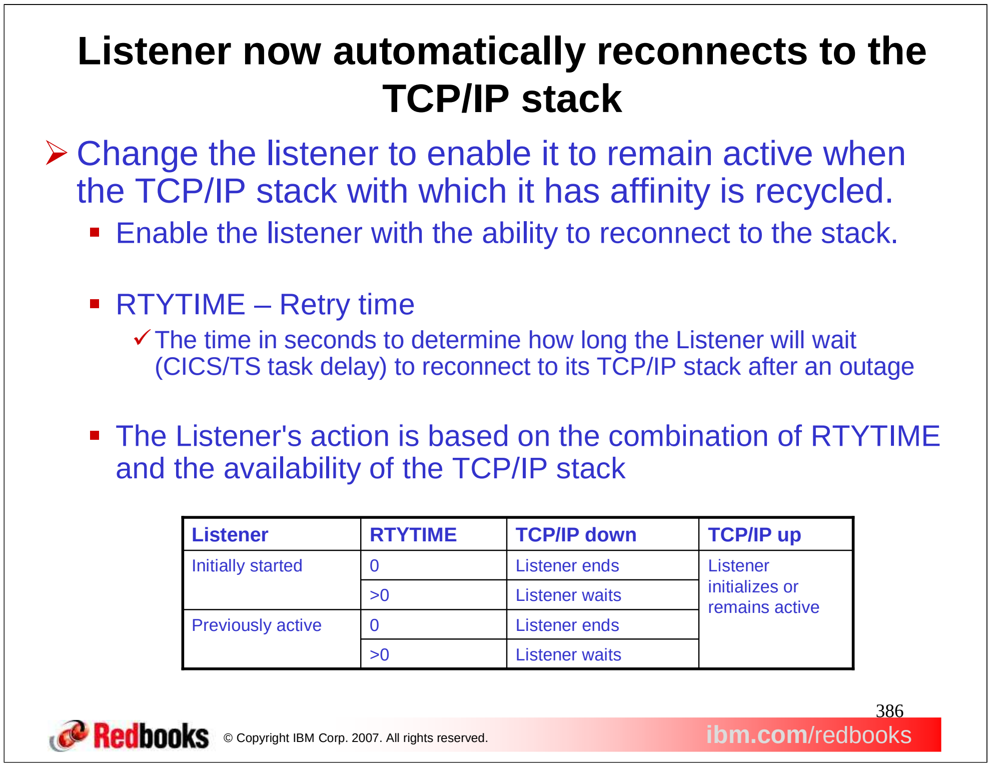
Listener task ends when TCP/IP stack goes down

- IP CICS Sockets supports multiple listeners – each instance separately configurable
- When the listener's TCP/IP stack is shutdown then the listener task ends.

CICS Sockets is implemented as an External Resource Manager in CICS.

The IP CICS Socket interface is comprised of the a CICS task related user exit (TRUE), an application programming interface (API), a listener program (EZACIC02) and CICS transactions for configuring (EZAC) and managing (EZAO) the interface and listeners. The socket interface consists of a set of calls that your CICS application programs can use to set up connections, send and receive data, and perform general communications control functions. The programs can be written in COBOL, PL/I, assembler language, or the C language.

The IP CICS Sockets listener task ends when its TCP/IP stack with which it has an affinity is shutdown. A subsequent operation is required to restart the listener task.



Listener now automatically reconnects to the TCP/IP stack

- Change the listener to enable it to remain active when the TCP/IP stack with which it has affinity is recycled.
  - Enable the listener with the ability to reconnect to the stack.
  - RTYTIME – Retry time
    - ✓ The time in seconds to determine how long the Listener will wait (CICS/TS task delay) to reconnect to its TCP/IP stack after an outage
  - The Listener's action is based on the combination of RTYTIME and the availability of the TCP/IP stack

Listener	RTYTIME	TCP/IP down	TCP/IP up
Initially started	0	Listener ends	Listener initializes or remains active
	>0	Listener waits	
Previously active	0	Listener ends	
	>0	Listener waits	

386

Redbooks

ibm.com/redbooks

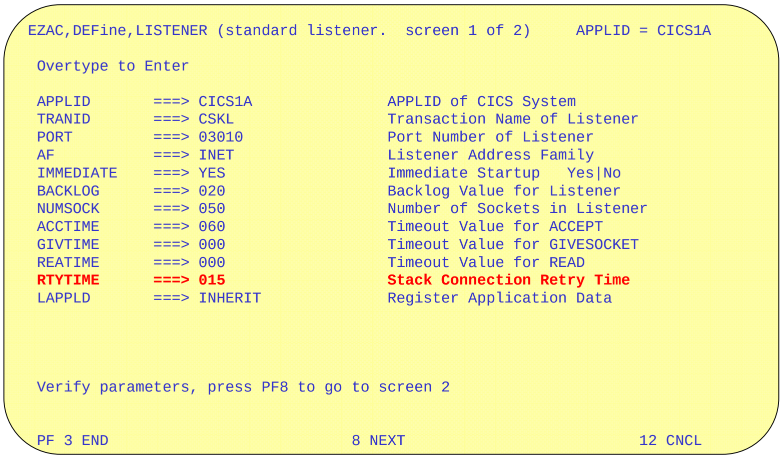
Change the listener to automatically re-establish itself (based on configuration) when its stack is restarted.

The retry time configuration option is used to indicate that we want the listener to re-connect to its TCP/IP stack. The value specified by RTYTIME affects the initial connection or re-connection of the listener to its TCP/IP stack. The range for RTYTIME is either 0 or 15-999 seconds. The value of 0 indicates that the listener will not be delayed but will retry to connect to its TCP/IP stack once and will end if that attempt fails. If RTYTIME is configured with the values 1 through 14 then the value of 15 seconds will be used instead to prevent excessive connection attempts. If RTYTIME is not specified then the default value of 15 seconds is used.

The chart shows how the listener will react based on how the RTYTIME configuration option is specified and the availability of the TCP/IP stack.
The RTYTIME configuration option can be specified on the listener definition, EZACICD TYPE=LISTENER. The IP CICS Sockets configuration macro, EZACICD, is designed to support the configuration of the IP CICS Socket interface and listeners exploiting that interface. Regenerate the IP CICS Socket configuration dataset, EZACONFG, with the new RTYTIME configuration option.

Configuration
Standard listener

➢ Standard listener definition: EZAC transaction

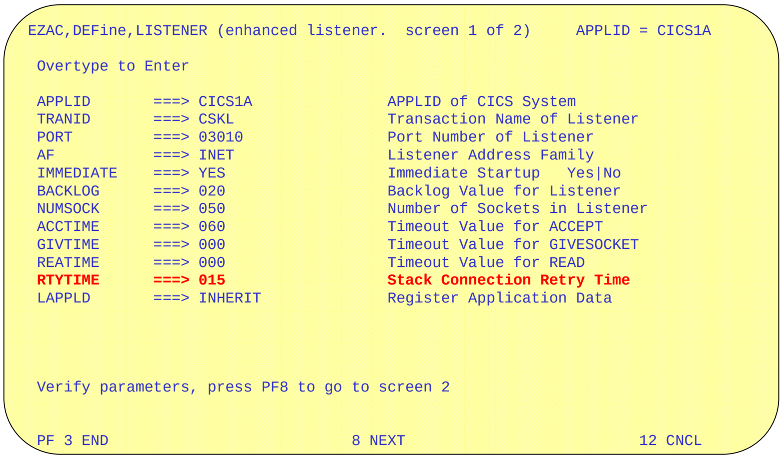


```
EZAC,DEFine,LISTENER (standard listener.  screen 1 of 2)     APPLID = CICS1A

 Overtype to Enter

 APPLID       ===> CICS1A                 APPLID of CICS System
 TRANID       ===> CSKL                   Transaction Name of Listener
 PORT         ===> 03010                  Port Number of Listener
 AF           ===> INET                   Listener Address Family
 IMMEDIATE    ===> YES                    Immediate Startup   Yes|No
 BACKLOG      ===> 020                    Backlog Value for Listener
 NUMSOCK      ===> 050                    Number of Sockets in Listener
 ACCTIME      ===> 060                    Timeout Value for ACCEPT
 GIVTIME      ===> 000                    Timeout Value for GIVESOCKET
 REATIME      ===> 000                    Timeout Value for READ
 RTYTIME      ===> 015                    Stack Connection Retry Time
 LAPPLD       ===> INHERIT                Register Application Data




 Verify parameters, press PF8 to go to screen 2


 PF 3 END                           8 NEXT                          12 CNCL
```

The RTYTIME configuration option can also be specified using the online configuration transaction, EZAC. The EZAC transaction will update the configuration contained in the EZACONFG VSAM dataset but not the original EZACICD macro. Ensure you mirror any updates against your sites EZACICD configuration macros.

Configuration
Enhanced listener

➢ Enhanced listener definition: EZAC transaction

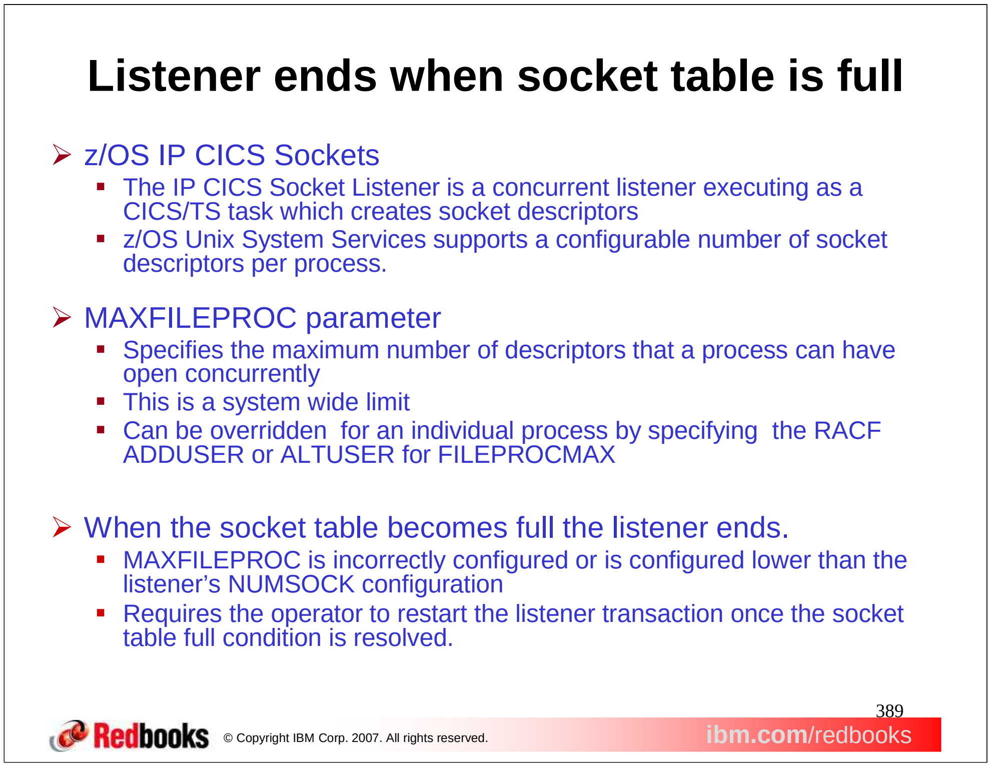


```
EZAC,DEFine,LISTENER (enhanced listener.  screen 1 of 2)     APPLID = CICS1A

 Overtype to Enter

 APPLID       ===> CICS1A                 APPLID of CICS System
 TRANID       ===> CSKL                   Transaction Name of Listener
 PORT         ===> 03010                  Port Number of Listener
 AF           ===> INET                   Listener Address Family
 IMMEDIATE    ===> YES                    Immediate Startup   Yes|No
 BACKLOG      ===> 020                    Backlog Value for Listener
 NUMSOCK      ===> 050                    Number of Sockets in Listener
 ACCTIME      ===> 060                    Timeout Value for ACCEPT
 GIVTIME      ===> 000                    Timeout Value for GIVESOCKET
 REATIME      ===> 000                    Timeout Value for READ
 RTYTIME      ===> 015                    Stack Connection Retry Time
 LAPPLD       ===> INHERIT                Register Application Data




 Verify parameters, press PF8 to go to screen 2


 PF 3 END                           8 NEXT                          12 CNCL
```

The RTYTIME configuration option can also be specified, for the enhanced listener, using the online configuration transaction, EZAC. The EZAC transaction will update the configuration contained in the EZACONFG VSAM dataset but not the original EZACICD macro. Ensure you mirror any updates against your sites EZACICD configuration macros.

The EZAC transaction will issue particular messages when validating the RTYTIME configuration option.

A non-numeric value is considered incorrect. A value other than 0 or 15-999 is considered an incorrect value. A value of 1-14 is considered to be below the minimum non-zero value of 15. Change any incorrect value and press the enter key.



Listener ends when socket table is full

- z/OS IP CICS Sockets
  - The IP CICS Socket Listener is a concurrent listener executing as a CICS/TS task which creates socket descriptors
  - z/OS Unix System Services supports a configurable number of socket descriptors per process.
- MAXFILEPROC parameter
  - Specifies the maximum number of descriptors that a process can have open concurrently
  - This is a system wide limit
  - Can be overridden for an individual process by specifying the RACF ADDUSER or ALTUSER for FILEPROCMAX
- When the socket table becomes full the listener ends.
  - MAXFILEPROC is incorrectly configured or is configured lower than the listener’s NUMSOCK configuration
  - Requires the operator to restart the listener transaction once the socket table full condition is resolved.

389

Redbooks ibm.com/redbooks

The listener is designed to function as a concurrent listener. It creates and manages its sockets giving them to child server tasks. It creates a listening socket and accepts sockets from client connections.

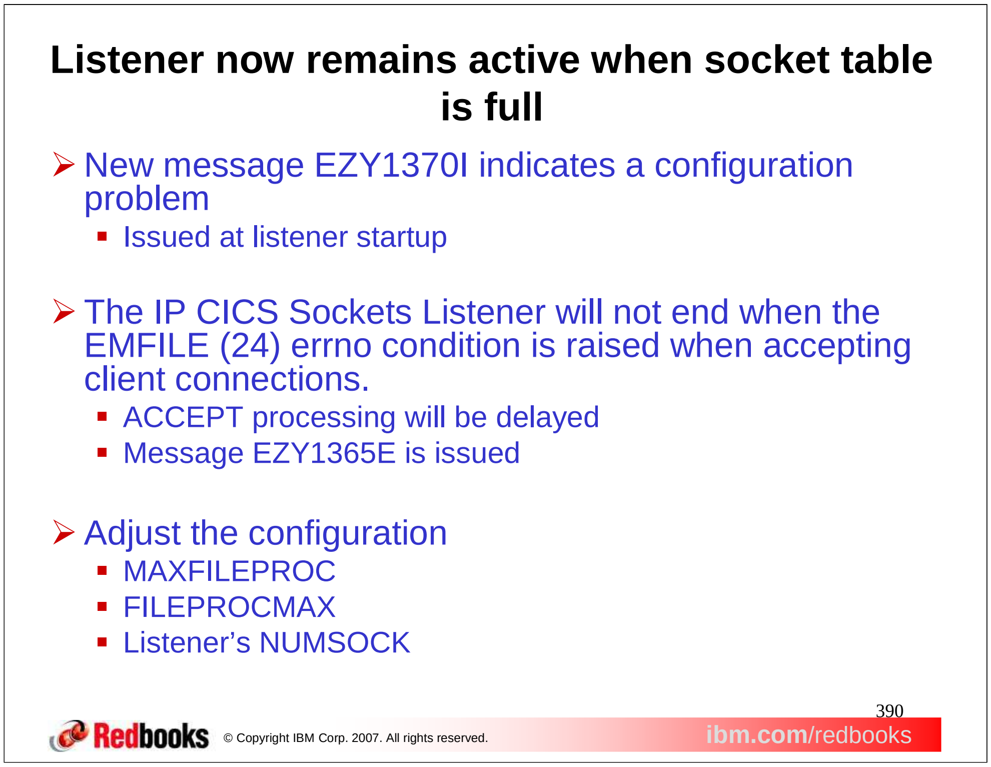
CICS/TS 2.2 is dubbed to OMVS as one process per address space. CICS/TS 2.3 is dubbed to OMVS as one process per CICS/TS task.

MAXFILEPROC is a z/OS UNIX System Services parameter that specifies the maximum number of descriptors for files, sockets, directories, and any other file system objects that a single process can have concurrently active or allocated. It is defined in SYS1.PARMLIB member, BPXPRMxx. It can be changed using the SETOMVS console command. Note that MAXFILEPROC is the same as the OPEN_MAX variable in the POSIX standard.

FILEPROCMAX specifies the maximum number of files the user is allowed to have concurrently active or open. The files-per-process you define to RACF is a numeric value from 3 and 542287. The value specified for FILEPROCMAX overrides any value provided by the MAXFILEPROC parameter of BPXPRMxx. The RACF ALTUSER command is used to change the information in a user's profile. The RACF ADDUSER command is used to define a new user to RACF and establish the user's relationship to an existing RACF-defined group.

When the listener reaches its maximum number of allowed descriptors then it will end.

The MAXFILEPROC default is 64000 (The POSIX standard is 16) and the IP CICS Socket NUMSOCK configuration option default is 50.



Listener now remains active when socket table is full

- New message EZY1370I indicates a configuration problem
  - Issued at listener startup
- The IP CICS Sockets Listener will not end when the EMFILE (24) errno condition is raised when accepting client connections.
  - ACCEPT processing will be delayed
  - Message EZY1365E is issued
- Adjust the configuration
  - MAXFILEPROC
  - FILEPROCMAX
  - Listener's NUMSOCK

The solution to the problem is to configure the environment by setting the listener's NUMSOCK value to be less than or equal to the FILEPROCMAX value for the listener's user ID or the value specified for MAXFILEPROC if FILEPROCMAX is not being used.

A run-time check is done when the listener starts to determines if the z/OS UNIX System Services MAXFILEPROC value is less than or equal to the listener's NUMSOCK value.   If so the following message is issued.

```
EZY1370I mm/dd/yy hh:mm:ss LISTENER transactionid NUMSOCK numsock IS EQUAL
TO OR GREATER THAN MAXFILEPROC maxfileproc
```

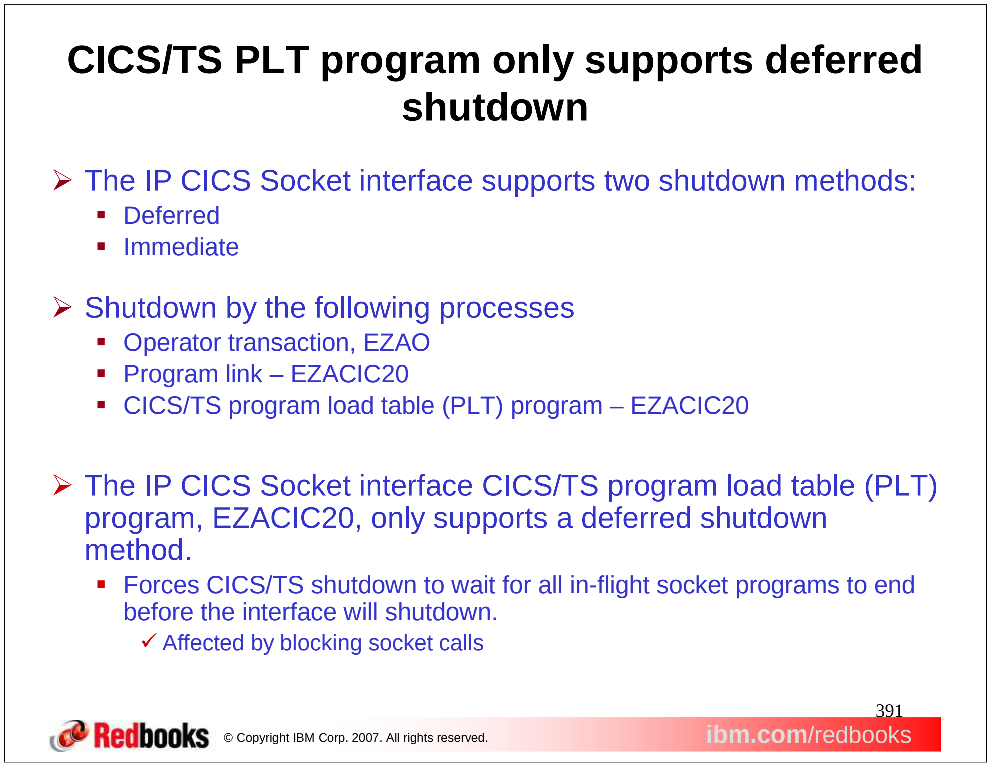
The listener's accept processing will pause when the number of sockets being handled exceeds the MAXFILEPROC value. No new connections will be accepted until the number of sockets falls below the MAXFILEPROC value.

Also, the value specified for the user ID's FILEPROCMAX should be configured appropriately. If the number of sockets the listener creates exceeds the listeners user ID's FILEPROCMAX value or the MAXFILEPROC value then a EMFILE error condition occurs and the listener will cease accepting new sockets until its active sockets are at the FILEPROCMAX value or less. If FILEPROCMAX is not being used then the number of active sockets will have to be equal to or less than the MAXFILEPROC value.

The listener has been changed to remain active when it reaches its maximum number of descriptors.   The EMFILE errno is used to indicate that the process table is full. The following message will be issued when the listener has received the EMFILE error condition from ACCEPT processing.

```
EZY1365E mm/dd/yy hh:mm:ss LISTENER transactionid taskno IS NOT ACCEPTING
REQUESTS ON PORT port
```

This message indicates that the listener identified by the specified transaction ID and task number is not able to process inbound connections because the listener's socket descriptor table is full. Once a successful ACCEPT is processed then this condition will be relieved.



CICS/TS PLT program only supports deferred shutdown

- The IP CICS Socket interface supports two shutdown methods:
  - Deferred
  - Immediate
- Shutdown by the following processes
  - Operator transaction, EZAO
  - Program link – EZACIC20
  - CICS/TS program load table (PLT) program – EZACIC20
- The IP CICS Socket interface CICS/TS program load table (PLT) program, EZACIC20, only supports a deferred shutdown method.
  - Forces CICS/TS shutdown to wait for all in-flight socket programs to end before the interface will shutdown.
    - ✓ Affected by blocking socket calls

391

Redbooks ibm.com/redbooks

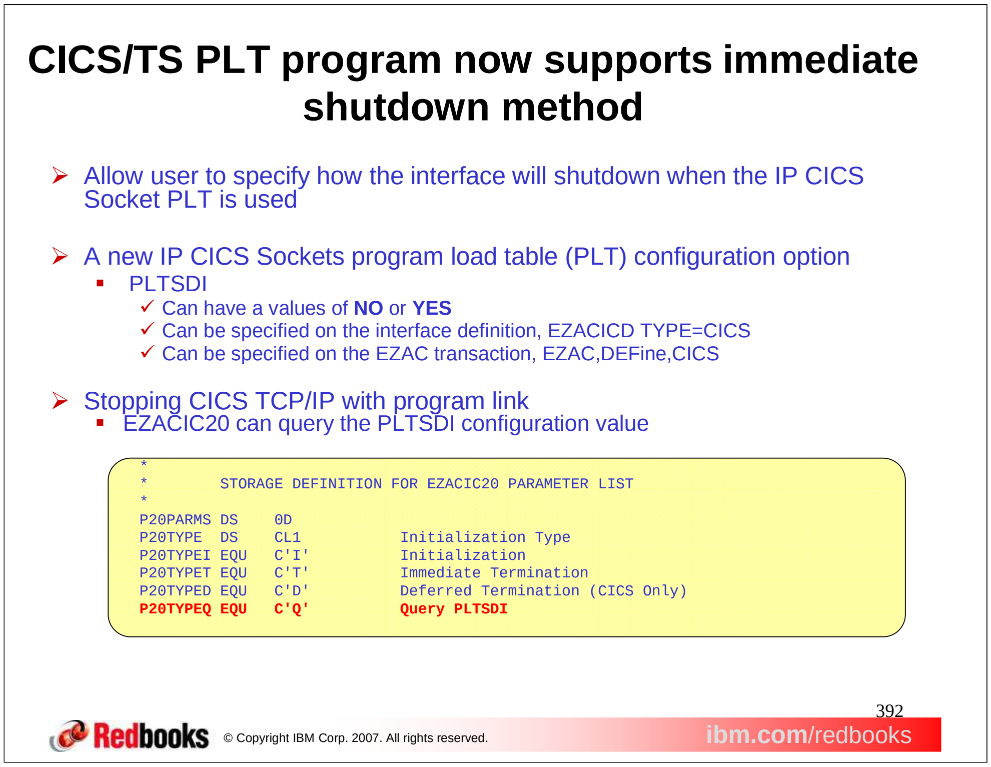
Historically, the IP CICS Socket interface can be shutdown using many different methods. A deferred shutdown enables all IP CICS sockets tasks to end gracefully. An immediate shutdown directs all IP CICS sockets tasks to be immediately terminated**.**

Shutdown can be either deferred or immediate using the EZAO operator transaction.

Shutdown can also be deferred or immediate using the program link EZACIC20. The P20TYPE field in the COMMAREA provided via the EXEC CICS LINK to program EZACIC20 specifies whether immediate or a deferred termination is requested.

You can allow automatic shutdown of the CICS socket interface through updates to the program load table (PLT). This is achieved through placing the EZACIC20 module in the appropriate PLT. Ideally, the CICS system programmer should add the IP CICS Socket shutdown program, EZACIC20, to their program load table (PLT) to facilitate in shutting down the IP CICS Socket interface and listener. Only the deferred shutdown method is supported when using the PLT.

When the IP CICS Socket PLT program is used any transactions being blocked by blocking sockets command will wait for them to return. This may require the CICS system programmer to use CICS shutdown assist or manually terminate those blocking tasks. When recycling CICS/TS this manual action will elongate the user’s down times due to having to wait for CICS/TS to shutdown before being restarted.



CICS/TS PLT program now supports immediate shutdown method

- Allow user to specify how the interface will shutdown when the IP CICS Socket PLT is used
- A new IP CICS Sockets program load table (PLT) configuration option
  - PLTSDI
    - ✓ Can have a values of **NO** or **YES**
    - ✓ Can be specified on the interface definition, EZACICD TYPE=CICS
    - ✓ Can be specified on the EZAC transaction, EZAC,DEFine,CICS
- Stopping CICS TCP/IP with program link
  - EZACIC20 can query the PLTSDI configuration value

```
*
*        STORAGE DEFINITION FOR EZACIC20 PARAMETER LIST
*
P20PARMS DS    0D
P20TYPE  DS    CL1           Initialization Type
P20TYPEI EQU   C'I'          Initialization
P20TYPET EQU   C'T'          Immediate Termination
P20TYPED EQU   C'D'          Deferred Termination (CICS Only)
P20TYPEQ EQU   C'Q'          Query PLTSDI
```

392

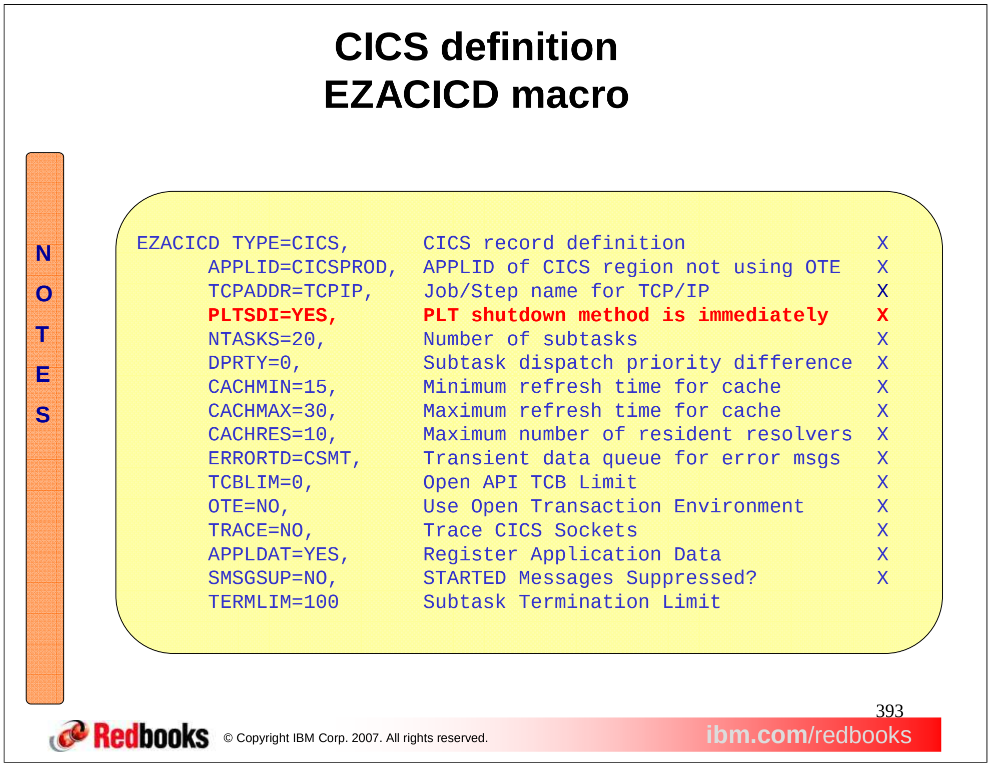
Redbooks ibm.com/redbooks

The PLT program, EZACIC20, now has the ability to shutdown using the immediate method whereby any transaction being blocked by a blocking socket call is immediately unblocked. The new PLT shutdown immediate configuration option, PLTSDI, is used to tell the PLT program how to shutdown the interface and listeners when CICS/TS is being shutdown. This option can have a value of NO or YES. The value of NO is used to signify that a deferred shutdown is desired. This is the default. The value of YES is used to signify that an immediate shutdown is desired.

The configuration macro, EZACICD, is used to build the configuration data set. TYPE=CICS identifies a CICS object. The new PLTSDI option can be configured on the interface definition, EZACICD TYPE=CICS. If PLTSDI is not specified then a deferred shutdown is performed.

The EZAC transaction is a panel-driven interface that lets you add, delete, or modify the configuration file. The DEFINE function is used to create CICS objects and their listener objects. The PLTSDI can be configured on the definition of a CICS object.

In prior releases, the user can specify a shutdown method of deferred or immediate when using the program link EZACIC20. In V1R9 you know have the option of querying the PLTSDI configuration to determine which shutdown method to be used. You still have the option of selecting a deferred or immediate shutdown method. If you have created your own maintenance transaction then you may specify that the EZACIC20 program query the PLTSDI configuration option by setting the P20TYPE field in the P20PARM COMMAREA to the value of ‘Q’ or P20TYPEQ before linking to EZACIC20.



CICS definition
EZACICD macro

NOTES

```
EZACICD TYPE=CICS,      CICS record definition                X
      APPLID=CICSPROD,  APPLID of CICS region not using OTE   X
      TCPADDR=TCPIP,    Job/Step name for TCP/IP              X
      PLTSDI=YES,       PLT shutdown method is immediately    X
      NTASKS=20,        Number of subtasks                    X
      DPRTY=0,          Subtask dispatch priority difference  X
      CACHMIN=15,       Minimum refresh time for cache        X
      CACHMAX=30,       Maximum refresh time for cache        X
      CACHRES=10,       Maximum number of resident resolvers  X
      ERRORTD=CSMT,     Transient data queue for error msgs   X
      TCBLIM=0,         Open API TCB Limit                    X
      OTE=NO,           Use Open Transaction Environment      X
      TRACE=NO,         Trace CICS Sockets                    X
      APPLDAT=YES,      Register Application Data             X
      SMSGSUP=NO,       STARTED Messages Suppressed?          X
      TERMLIM=100       Subtask Termination Limit
```

Redbooks

ibm.com/redbooks

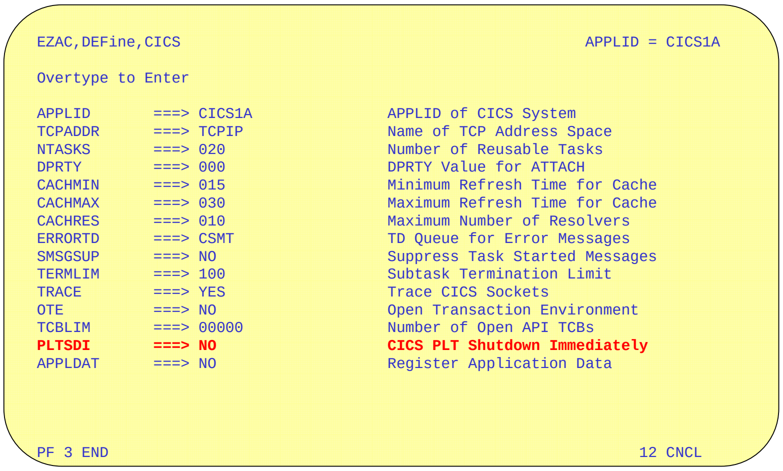
The PLTSDI configuration option has been added to the interface definition, EZACICD TYPE=CICS. The IP CICS Sockets configuration macro, EZACICD, is designed to support the configuration of the IP CICS Socket interface and listeners exploiting that interface. Regenerate the IP CICS Socket configuration dataset, EZACONFG, with the new PLTSDI configuration option.

The EZACICD macro will issue this MNOTE when validating the PLTSDI configuration option for a value other than YES or NO:

```
MNOTE 12,'INVALID VALUE SPECIFIED FOR PLTSDI, GENERATION TERMINATED'.
```

CICS definition
EZAC transaction

N O T E S

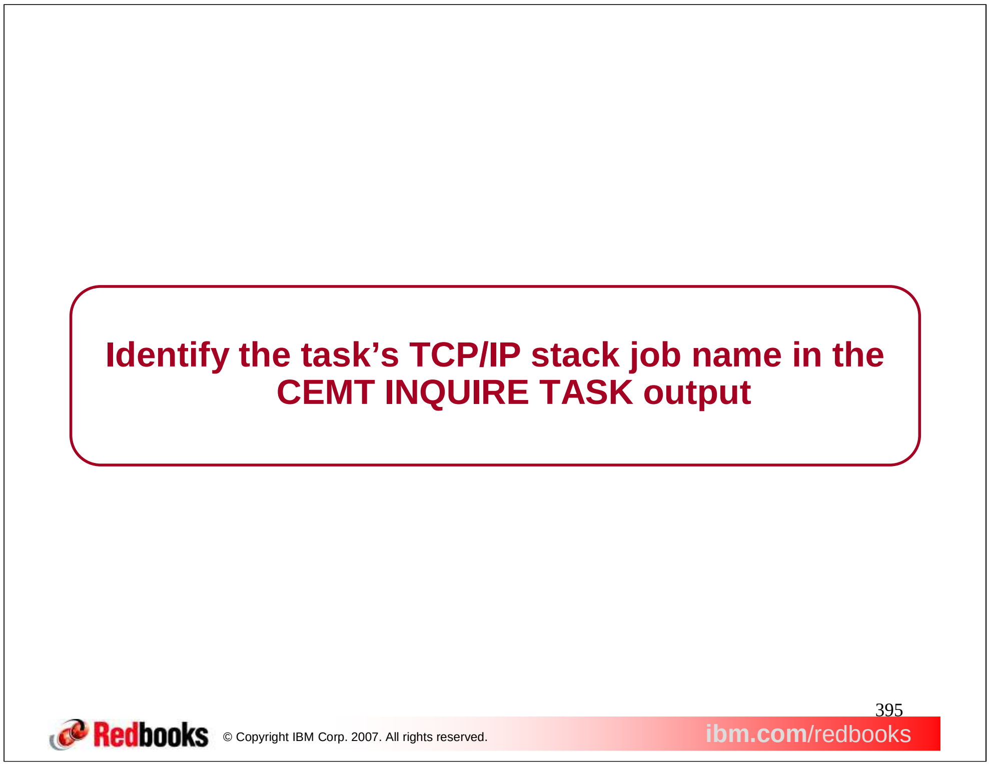


```
EZAC,DEFine,CICS                                              APPLID = CICS1A

Overtype to Enter

APPLID       ===> CICS1A                 APPLID of CICS System
TCPADDR      ===> TCPIP                  Name of TCP Address Space
NTASKS       ===> 020                    Number of Reusable Tasks
DPRTY        ===> 000                    DPRTY Value for ATTACH
CACHMIN      ===> 015                    Minimum Refresh Time for Cache
CACHMAX      ===> 030                    Maximum Refresh Time for Cache
CACHRES      ===> 010                    Maximum Number of Resolvers
ERRORTD      ===> CSMT                   TD Queue for Error Messages
SMSGSUP      ===> NO                     Suppress Task Started Messages
TERMLIM      ===> 100                    Subtask Termination Limit
TRACE        ===> YES                    Trace CICS Sockets
OTE          ===> NO                     Open Transaction Environment
TCBLIM       ===> 00000                  Number of Open API TCBs
PLTSDI       ===> NO                     CICS PLT Shutdown Immediately
APPLDAT      ===> NO                     Register Application Data



PF 3 END                                                            12 CNCL
```

The online configuration option also supports the PLTSDI configuration option. The EZAC transaction will update the configuration contained in the EZACONFG VSAM dataset but not the original EZACICD macro. Ensure you mirror any updates against your sites EZACICD configuration macros.

The EZAC configuration transaction will issue the followings message when validating the PLTSDI configuration option for a value other than YES or NO.

```
Incorrect or no entry for PLTSDI field, re-enter parameters or press ENTER
to continue.
```

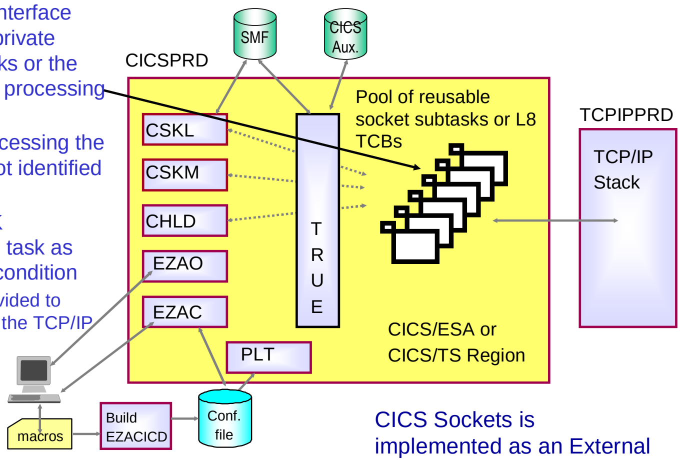


Identify the task’s TCP/IP stack job name in the CEMT INQUIRE TASK output

CICS Tasks using the IP CICS Socket API will show the job name of the TCP/IP stack processing the socket call.

TCP/IP stack jobname not in CEMT INQUIRE TASK output

- The IP CICS Socket interface uses either a pool of private MVS reusable subtasks or the CICS/TS L8 TCBs for processing socket commands.
- The TCP/IP stack processing the socket command is not identified with the waiting task.
- The CEMT INQ TASK transaction shows the task as being in USERWAIT condition
  - No information provided to identify the name of the TCP/IP stack



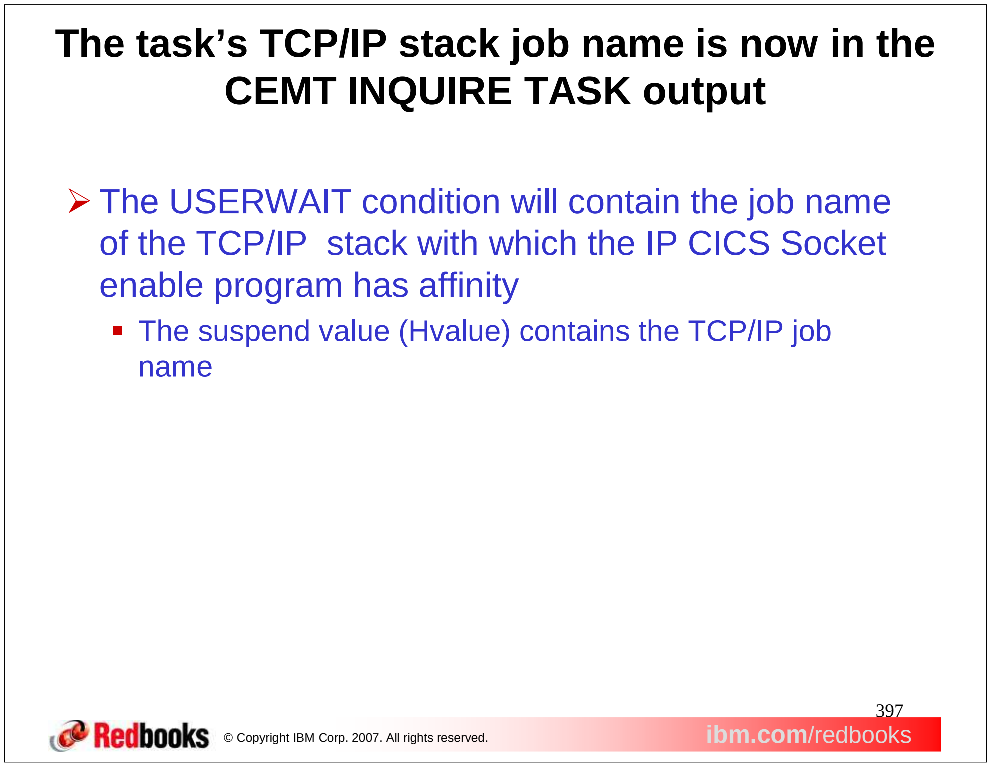
CICS Sockets is implemented as an External Resource Manager in CICS.

The IP CICS Socket interface uses two distinct tasking methods to process socket commands. A MVS subtask and a L8 TCB provided by CICS/TS when using the open transaction environment.

CICS/TS Open Transaction Environment (OTE) introduces a new task of Task Control Blocks (TCBs) called open TCB, which can be used by applications. It is characterized by the fact that it is assigned to a CICS task for the life of the CICS task. Multiple OTE TCBs may run concurrently in CICS.
In the CICS/TS environment a socket command is considered a non-CICS resource. CICS/TS supports non-CICS resource through a task related user exit (TRUE). So when using the IP CICS Socket API(s) then a TRUE will be driven to support the non-CICS resource. The socket command will be processed on a TCB other than the CICS main TCB (QR TCB) to prevent the entire CICS workload from waiting when processing blocking socket commands (such as RECV). This TCB used for processing the socket command will be either one from the pool of private MVS subtasks (as established by the NTASKS IP CICS Socket configuration option) or one of the L8 TCBs supplied when exploiting CICS/TS open transaction environment (OTE).

IP CICS socket enabled programs are put into an external CICS wait while TCP/IP is processing the socket call (OTE=NO configured).

The CICS supplied transaction, CEMT, can be used by an operator to inquire about and change the values of parameters used by CICS, alter the status of the system resources, terminate tasks, and shut down the CICS system. The CEMT INQUIRE TASK transaction returns information about user tasks. Only information about user tasks can be displayed or changed. The Htype field displays the reason why the task is suspended. In prior releases, when Htype indicates a USERWAIT condition, then there is no further information to indicate which TCP/IP stack is associated with this task.



The task’s TCP/IP stack job name is now in the CEMT INQUIRE TASK output

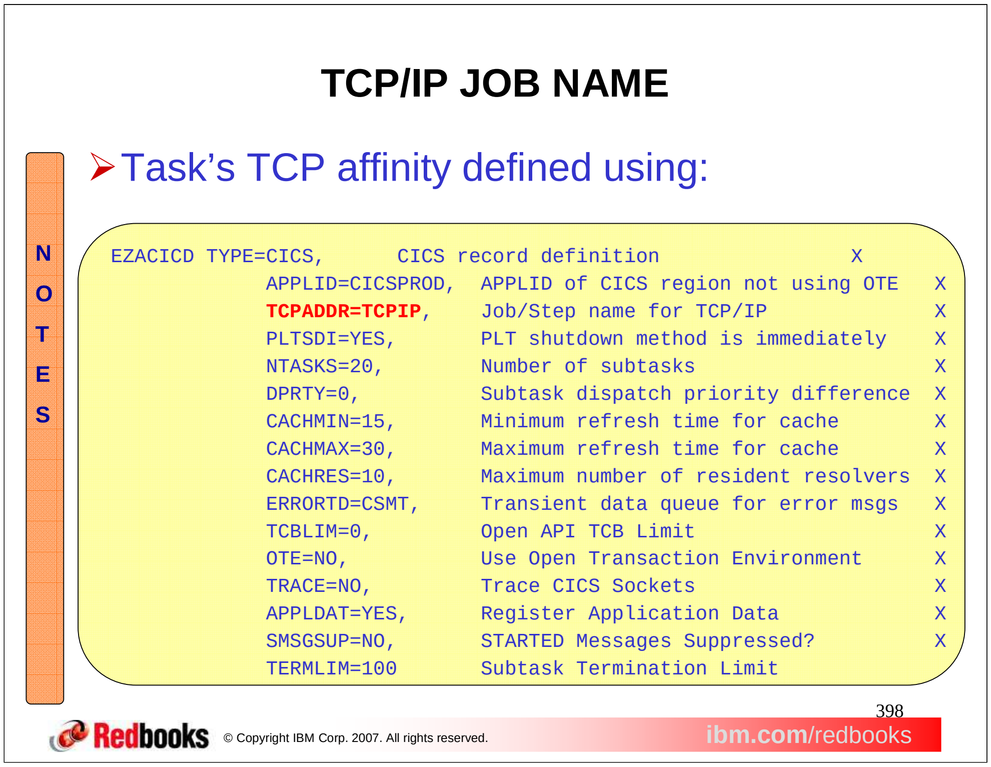
- The USERWAIT condition will contain the job name of the TCP/IP stack with which the IP CICS Socket enable program has affinity
  - The suspend value (Hvalue) contains the TCP/IP job name

397

Redbooks

ibm.com/redbooks

The user wait condition will now be associated with the job name of the TCP/IP stack which the CICS transaction has affinity. Tasks processing a socket command can then be associated with the TCP/IP stack job name processing that socket command.



N
O
T
E
S

TCP/IP JOB NAME

- Task’s TCP affinity defined using:

```
EZACICD TYPE=CICS,      CICS record definition                X
             APPLID=CICSPROD,  APPLID of CICS region not using OTE   X
             TCPADDR=TCPIP,    Job/Step name for TCP/IP              X
             PLTSDI=YES,       PLT shutdown method is immediately    X
             NTASKS=20,        Number of subtasks                    X
             DPRTY=0,          Subtask dispatch priority difference  X
             CACHMIN=15,       Minimum refresh time for cache        X
             CACHMAX=30,       Maximum refresh time for cache        X
             CACHRES=10,       Maximum number of resident resolvers  X
             ERRORTD=CSMT,     Transient data queue for error msgs   X
             TCBLIM=0,         Open API TCB Limit                    X
             OTE=NO,           Use Open Transaction Environment      X
             TRACE=NO,         Trace CICS Sockets                    X
             APPLDAT=YES,      Register Application Data             X
             SMSGSUP=NO,       STARTED Messages Suppressed?          X
             TERMLIM=100       Subtask Termination Limit
```

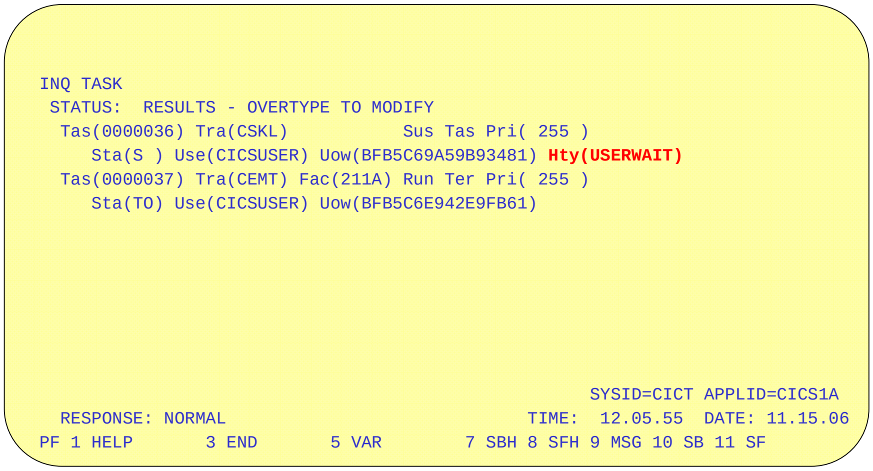
398

Redbooks ibm.com/redbooks

The IP CICS Socket interface TCPADDR configuration option was not changed in z/OS V1R9 but this shows the TCP/IP job name with which IP CICS Socket enabled transaction will have affinity.

CEMT INQ TASK

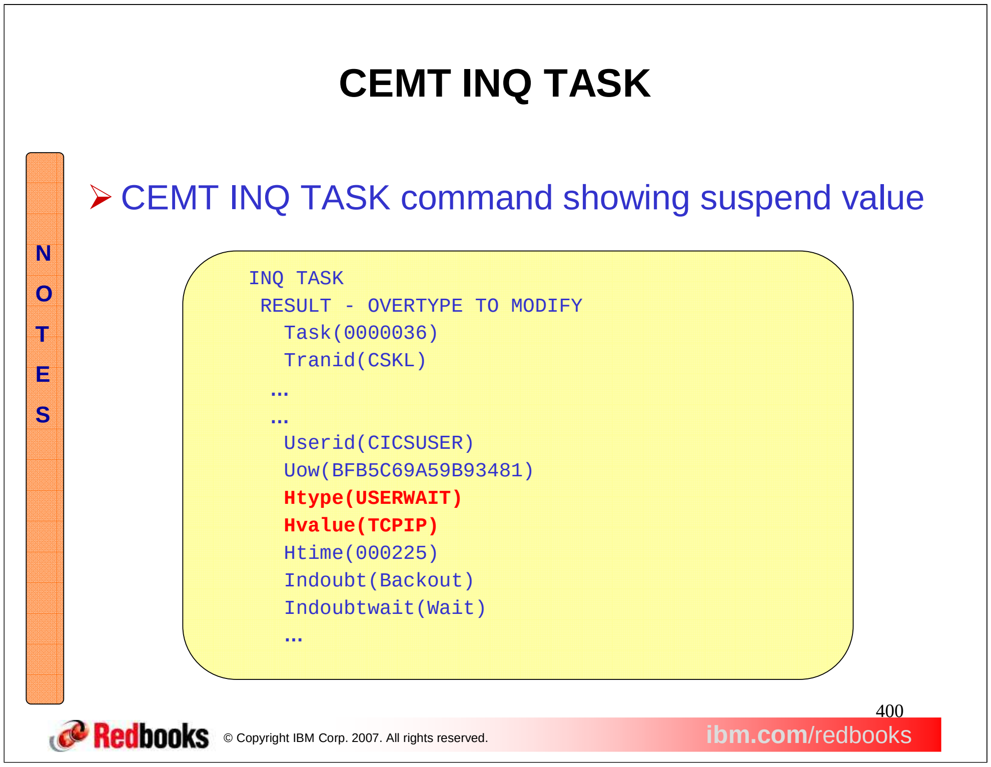
- CEMT INQ TASK command showing suspend type:



```
INQ TASK
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tas(0000036) Tra(CSKL)           Sus Tas Pri( 255 )
     Sta(S ) Use(CICSUSER) Uow(BFB5C69A59B93481) Hty(USERWAIT)
  Tas(0000037) Tra(CEMT) Fac(211A) Run Ter Pri( 255 )
     Sta(TO) Use(CICSUSER) Uow(BFB5C6E942E9FB61)








                                                     SYSID=CICT APPLID=CICS1A
  RESPONSE: NORMAL                             TIME:  12.05.55  DATE: 11.15.06
PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Redbooks ibm.com/redbooks

When issuing the CEMT INQ TASK transaction a user will see the suspend type (Hty) as USERWAIT. This indicates that task is processing a socket command and is waiting for its processing to complete in the TCP/IP stack.

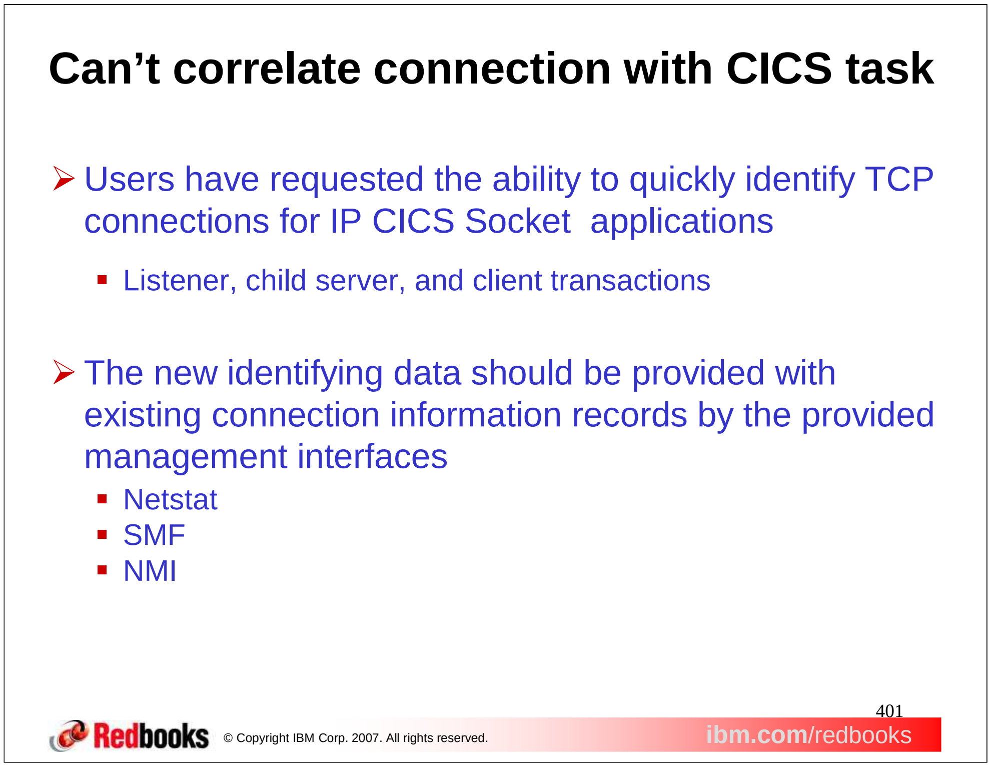


CEMT INQ TASK

- CEMT INQ TASK command showing suspend value

```
INQ TASK
 RESULT - OVERTYPE TO MODIFY
   Task(0000036)
   Tranid(CSKL)
  …
  …
   Userid(CICSUSER)
   Uow(BFB5C69A59B93481)
   Htype(USERWAIT)
   Hvalue(TCPIP)
   Htime(000225)
   Indoubt(Backout)
   Indoubtwait(Wait)
   …
```

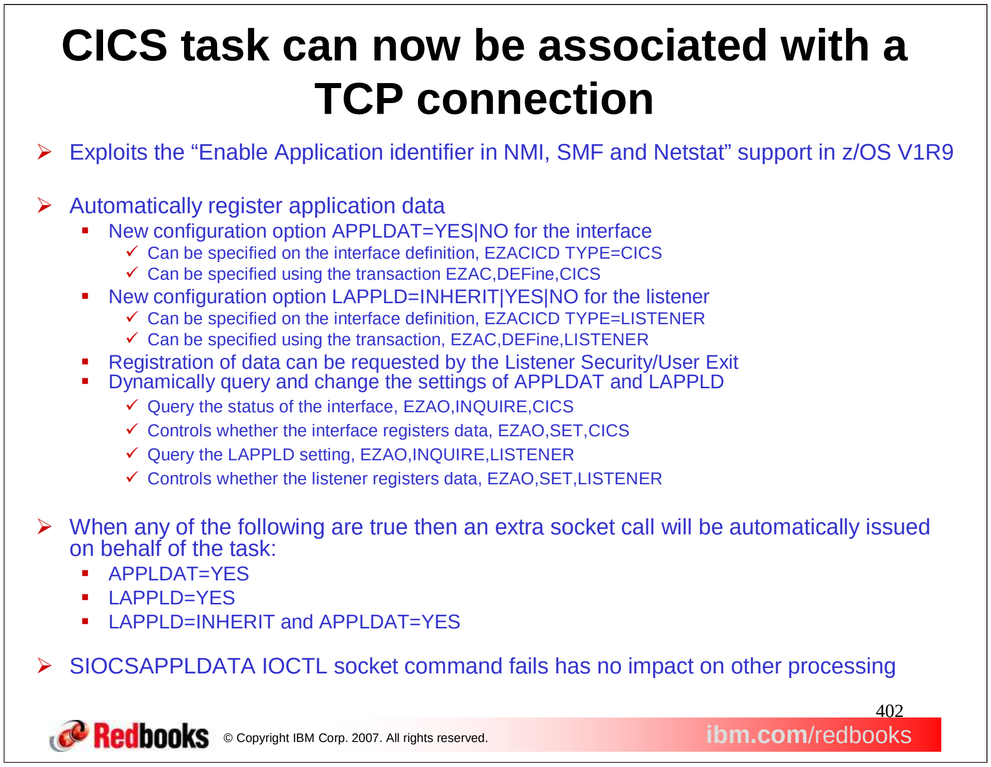
Using the question mark command, you can drill down to further CICS task attributes and see the suspend value (Hvalue) of the job name of the TCP/IP stack with which the task has affinity.



Can’t correlate connection with CICS task

- Users have requested the ability to quickly identify TCP connections for IP CICS Socket applications
  - Listener, child server, and client transactions
- The new identifying data should be provided with existing connection information records by the provided management interfaces
  - Netstat
  - SMF
  - NMI

It is very difficult to correlate a connection in a Netstat, SMF or NMI report with the actual CICS/TS task. Users have requested that a mechanism be provided such that they can easily correlate a connection with an actual CICS/TS task.



CICS task can now be associated with a TCP connection

- Exploits the “Enable Application identifier in NMI, SMF and Netstat” support in z/OS V1R9
- Automatically register application data
  - New configuration option APPLDAT=YES|NO for the interface
    - ✓ Can be specified on the interface definition, EZACICD TYPE=CICS
    - ✓ Can be specified using the transaction EZAC,DEFine,CICS
  - New configuration option LAPPLD=INHERIT|YES|NO for the listener
    - ✓ Can be specified on the interface definition, EZACICD TYPE=LISTENER
    - ✓ Can be specified using the transaction, EZAC,DEFine,LISTENER
  - Registration of data can be requested by the Listener Security/User Exit
  - Dynamically query and change the settings of APPLDAT and LAPPLD
    - ✓ Query the status of the interface, EZAO,INQUIRE,CICS
    - ✓ Controls whether the interface registers data, EZAO,SET,CICS
    - ✓ Query the LAPPLD setting, EZAO,INQUIRE,LISTENER
    - ✓ Controls whether the listener registers data, EZAO,SET,LISTENER
- When any of the following are true then an extra socket call will be automatically issued on behalf of the task:
  - APPLDAT=YES
  - LAPPLD=YES
  - LAPPLD=INHERIT and APPLDAT=YES
- SIOCSAPPLDATA IOCTL socket command fails has no impact on other processing

In z/OS V1R9 Communication Server we introduced a new function to enable application identifiers in NMI, SMF and the Netstat report. IP CICS Sockets is exploiting that function. Enabling the IP CICS Socket interface and listener with the ability to automatically register application data with its TCP connections. The automatic registration of the application data occurs at the IP CICS Socket Interface listeners, clients and child server programs. The application data used will be enough to direct the user back to the CICS/TS region where the connection originates. The IP CICS socket interface supports a new configuration option, APPLDATA, designed to cause application data to be automatically registered before LISTEN and GIVESOCKET or after CONNECT and TAKESOCKET. Valid values for this option are Yes or No. The default is No.

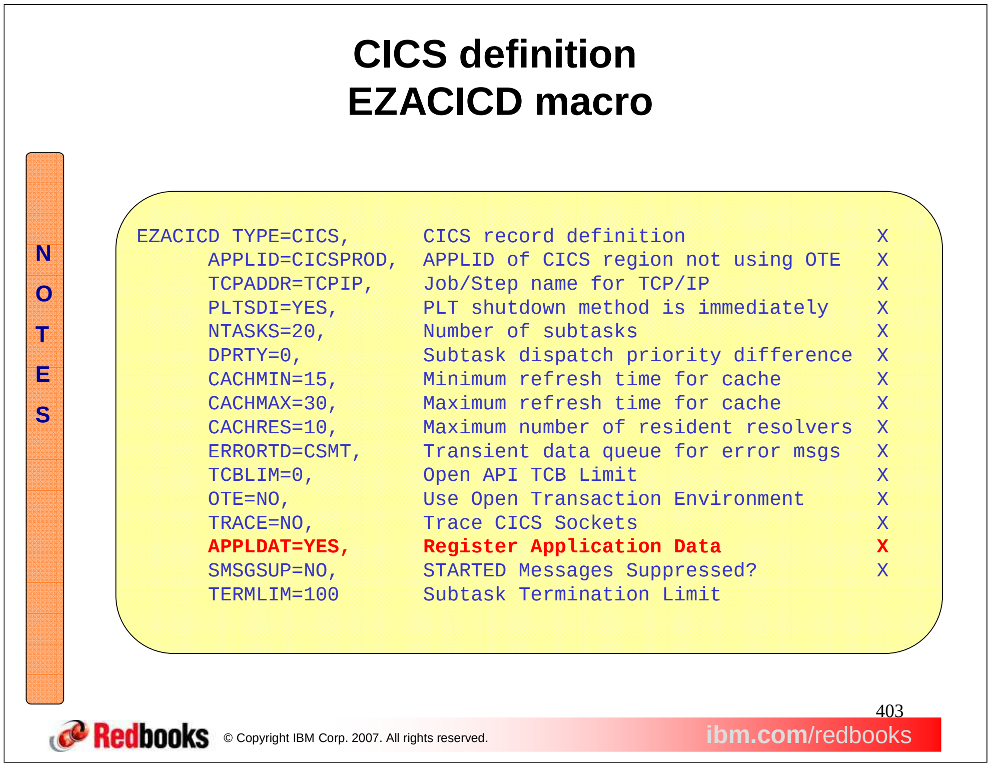
You may want to control listeners apart from the interface. The LAPPLD listener configuration option will allow you to override the APPLDAT configuration option. The LAPPLD option determines whether the IP CICS Sockets listener will automatically register IP CICS Sockets unique application data against its TCP connections. Valid values are Yes, No or the default of INHERIT. The value of INHERIT causes the listener to use the value specified for APPLDAT. LAPPLD affects the SIOCSAPPLDATA IOCTL being issued by the IBM Listener and therefore only affects the IOCTL surrounding the GIVESOCKET socket call.
The IP CICS Sockets security/user exit COMMAREA provides a one byte field that can be used by the security exit to indicate that the listener will register application data for the accepted socket to be given. Changes to this field will be honored by the listener. The character value of 0 indicates that no application data is being registered. The character value of 1 indicates that application data is being registered. If a security exit is used then it has the ability to override whether the listener will register application data for the accepted socket to be given.

To summarize the interface setting, APPLDAT, determines whether application data is processed for the overall interface. The listener setting, LAPPLD, will either supersede or inherit the value established by APPLDAT. And the security exit will reflect the listener’s LAPPLD setting and may alter that setting upon return to the listener.

The IP CICS Socket operator transaction, EZAO, is enhanced to allow the APPLDAT and LAPPLD settings to be dynamically changed without stopping and restarting the listener. This will be handy for users who only want to register application data during problem analysis. When EZAO is done for a listener, you can select from a list of active listeners or specify a specific listener. Optionally you can do a query or set for a specific listener by specifying the transaction ID.

When enabled, the interface and/or listener may register application data before or after specific socket commands. Application data is registered after CONNECT or connect(), before GIVESOCKET (IBM Listener only), before LISTEN or listen(), and after TAKESOCKET or takesocket().

An extra pass to the TCP/IP stack will be generated on behalf of applications using the interface and listener. If you use the IP CICS Socket control blocks then re-compile your programs to get the latest versions of the



CICS definition
EZACICD macro

NOTES

```
EZACICD TYPE=CICS,       CICS record definition                X
      APPLID=CICSPROD,   APPLID of CICS region not using OTE   X
      TCPADDR=TCPIP,     Job/Step name for TCP/IP              X
      PLTSDI=YES,        PLT shutdown method is immediately    X
      NTASKS=20,         Number of subtasks                    X
      DPRTY=0,           Subtask dispatch priority difference  X
      CACHMIN=15,        Minimum refresh time for cache        X
      CACHMAX=30,        Maximum refresh time for cache        X
      CACHRES=10,        Maximum number of resident resolvers  X
      ERRORTD=CSMT,      Transient data queue for error msgs   X
      TCBLIM=0,          Open API TCB Limit                    X
      OTE=NO,            Use Open Transaction Environment      X
      TRACE=NO,          Trace CICS Sockets                    X
      APPLDAT=YES,       Register Application Data             X
      SMSGSUP=NO,        STARTED Messages Suppressed?          X
      TERMLIM=100        Subtask Termination Limit
```

403

Redbooks © Copyright IBM Corp. 2007. All rights reserved. ibm.com/redbooks

The APPLDAT configuration option can be specified on the interface definition, EZACICD TYPE=CICS. The IP CICS Sockets configuration macro, EZACICD, is designed to support the configuration of the IP CICS Socket interface and listeners exploiting that interface. Regenerate the IP CICS Socket configuration dataset, EZACONFG, with the new APPLDAT configuration option.

The following Assembler MNOTE will be produced by the EZACICD macro when processing the APPLDAT configuration option for a value other than YES or NO.

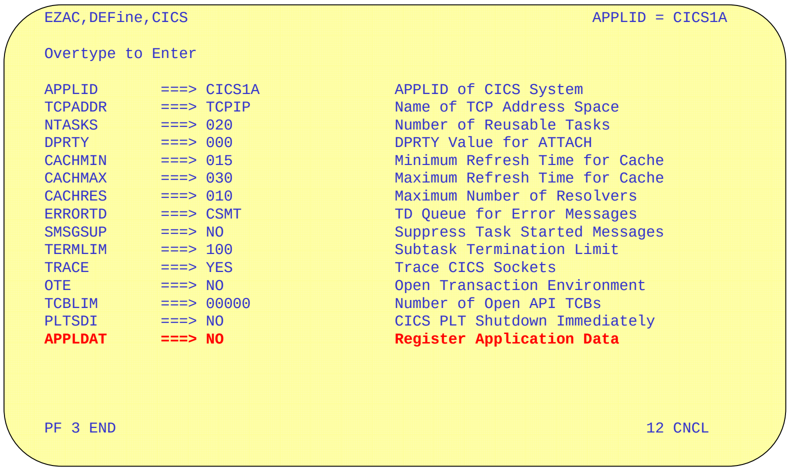
```
MNOTE 12,'INVALID VALUE SPECIFIED FOR APPLDAT, GENERATION TERMINATED'.
```

The following Assembler MNOTE will be produced by the EZACICD macro when the APPLDAT is not specified.

```
MNOTE 0,'APPLDAT NOT SPECIFIED BUT DEFAULTS TO NO'
```

CICS definition
EZAC transaction

N O T E S



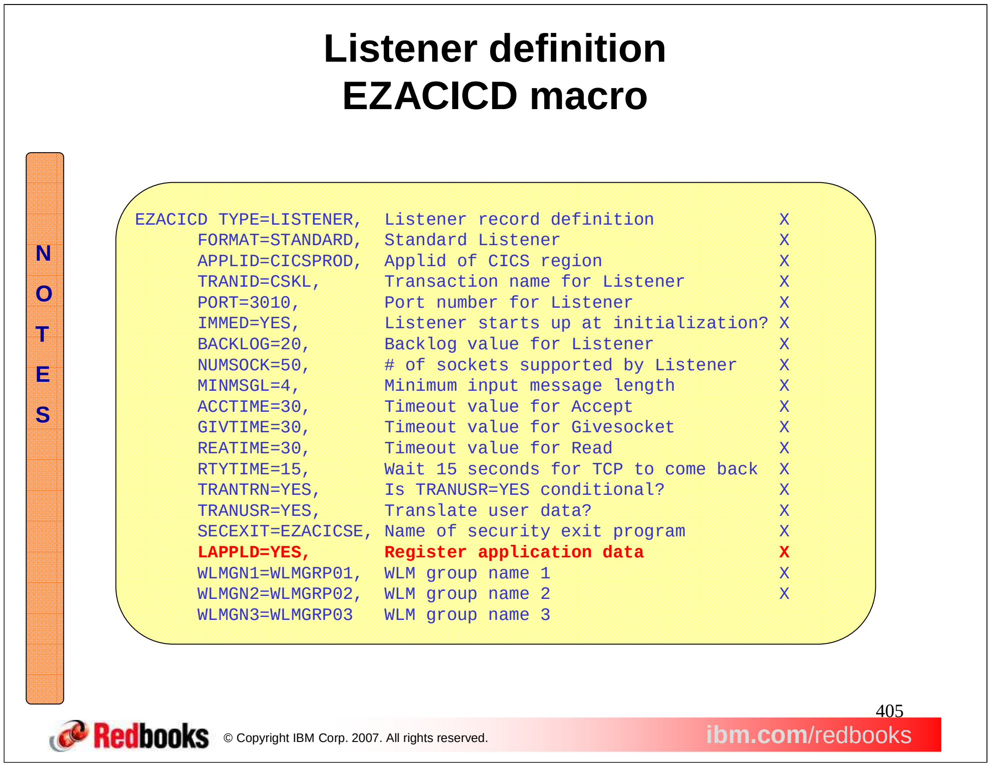
```
EZAC,DEFine,CICS                                              APPLID = CICS1A

Overtype to Enter

APPLID       ===> CICS1A               APPLID of CICS System
TCPADDR      ===> TCPIP                Name of TCP Address Space
NTASKS       ===> 020                  Number of Reusable Tasks
DPRTY        ===> 000                  DPRTY Value for ATTACH
CACHMIN      ===> 015                  Minimum Refresh Time for Cache
CACHMAX      ===> 030                  Maximum Refresh Time for Cache
CACHRES      ===> 010                  Maximum Number of Resolvers
ERRORTD      ===> CSMT                 TD Queue for Error Messages
SMSGSUP      ===> NO                   Suppress Task Started Messages
TERMLIM      ===> 100                  Subtask Termination Limit
TRACE        ===> YES                  Trace CICS Sockets
OTE          ===> NO                   Open Transaction Environment
TCBLIM       ===> 00000                Number of Open API TCBs
PLTSDI       ===> NO                   CICS PLT Shutdown Immediately
APPLDAT      ===> NO                   Register Application Data



PF 3 END                                                            12 CNCL
```

Redbooks ibm.com/redbooks

The APPLDAT configuration option can also be specified using the online configuration transaction, EZAC. The EZAC transaction will update the configuration contained in the EZACONFG VSAM dataset but not the original EZACICD macro. Ensure you mirror any updates against your sites EZACICD configuration macros.

The following message will be produced by the EZAC transaction when processing the APPLDAT configuration option for a value other than YES or NO.

```
Incorrect or no entry for APPLDAT field
Re-enter parameters or press ENTER to continue.
```



Listener definition EZACICD macro

N O T E S

```
EZACICD TYPE=LISTENER,  Listener record definition            X
      FORMAT=STANDARD,  Standard Listener                     X
      APPLID=CICSPROD,  Applid of CICS region                 X
      TRANID=CSKL,      Transaction name for Listener         X
      PORT=3010,        Port number for Listener              X
      IMMED=YES,        Listener starts up at initialization? X
      BACKLOG=20,       Backlog value for Listener            X
      NUMSOCK=50,       # of sockets supported by Listener    X
      MINMSGL=4,        Minimum input message length          X
      ACCTIME=30,       Timeout value for Accept              X
      GIVTIME=30,       Timeout value for Givesocket          X
      REATIME=30,       Timeout value for Read                X
      RTYTIME=15,       Wait 15 seconds for TCP to come back  X
      TRANTRN=YES,      Is TRANUSR=YES conditional?           X
      TRANUSR=YES,      Translate user data?                  X
      SECEXIT=EZACICSE, Name of security exit program         X
      LAPPLD=YES,       Register application data             X
      WLMGN1=WLMGRP01,  WLM group name 1                      X
      WLMGN2=WLMGRP02,  WLM group name 2                      X
      WLMGN3=WLMGRP03   WLM group name 3
```

Redbooks ibm.com/redbooks

The LAPPLD configuration option can be specified on the interface definition, EZACICD TYPE=LISTENER. The IP CICS Sockets configuration macro, EZACICD, is designed to support the configuration of the IP CICS Socket interface and listeners exploiting that interface. Regenerate the IP CICS Socket configuration dataset, EZACONFG, with the new LAPPLD configuration option.

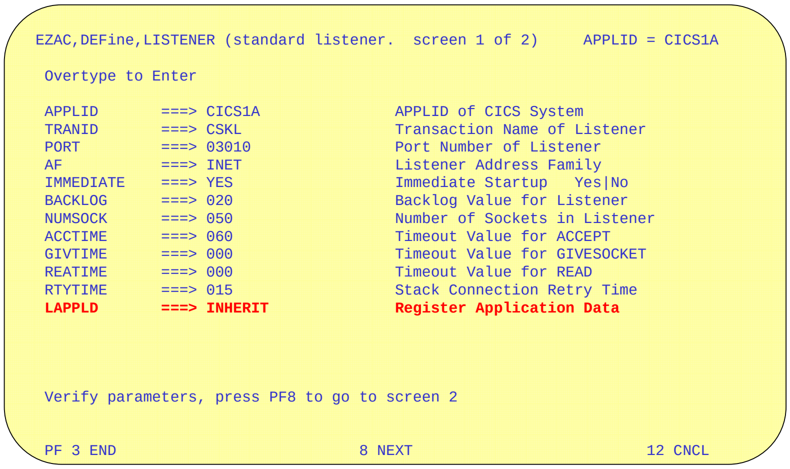
The following Assembler MNOTE will be produced by the EZACICD macro when processing the LAPPLD configuration option for a value other than INHERIT, YES or NO.

MNOTE 12,’INVALID VALUE SPECIFIED FOR LAPPLD, GENERATION TERMINATED’

The following Assembler MNOTE will be produced by the EZACICD macro when the LAPPLD is not specified.

MNOTE 0,'LAPPLD NOT SPECIFIED BUT INHERITS INTERFACE APPLDAT'

Standard listener definition
EZAC transaction



```
EZAC,DEFine,LISTENER (standard listener.  screen 1 of 2)     APPLID = CICS1A

 Overtype to Enter

 APPLID       ===> CICS1A                  APPLID of CICS System
 TRANID       ===> CSKL                    Transaction Name of Listener
 PORT         ===> 03010                   Port Number of Listener
 AF           ===> INET                    Listener Address Family
 IMMEDIATE    ===> YES                     Immediate Startup   Yes|No
 BACKLOG      ===> 020                     Backlog Value for Listener
 NUMSOCK      ===> 050                     Number of Sockets in Listener
 ACCTIME      ===> 060                     Timeout Value for ACCEPT
 GIVTIME      ===> 000                     Timeout Value for GIVESOCKET
 REATIME      ===> 000                     Timeout Value for READ
 RTYTIME      ===> 015                     Stack Connection Retry Time
 LAPPLD       ===> INHERIT                 Register Application Data



 Verify parameters, press PF8 to go to screen 2


 PF 3 END                           8 NEXT                          12 CNCL
```

The LAPPLD configuration option can also be specified using the online configuration transaction, EZAC. The EZAC transaction will update the configuration contained in the EZACONFG VSAM dataset but not the original EZACICD macro. Ensure you mirror any updates against your sites EZACICD configuration macros. Shown here is the panel for the standard listener. The LAPPLD configuration option can also be specified for the enhanced listener.

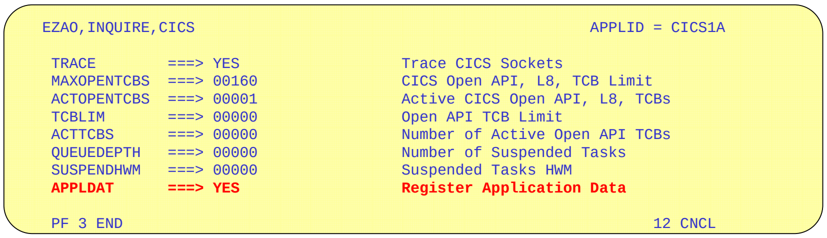
The following message will be produced by the EZAC transaction when processing the LAPPLD configuration option for a value other than INHERIT, YES or NO.

```
Incorrect or no entry for LAPPLD field
Re-enter parameters or press ENTER to continue.
```

Query or Set interface using EZAO transaction

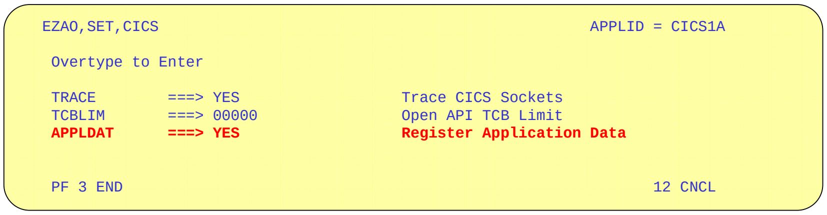
NOTES

- Query system setting



```
EZAO,INQUIRE,CICS                                           APPLID = CICS1A

 TRACE         ===> YES                     Trace CICS Sockets
 MAXOPENTCBS   ===> 00160                   CICS Open API, L8, TCB Limit
 ACTOPENTCBS   ===> 00001                   Active CICS Open API, L8, TCBs
 TCBLIM        ===> 00000                   Open API TCB Limit
 ACTTCBS       ===> 00000                   Number of Active Open API TCBs
 QUEUEDEPTH    ===> 00000                   Number of Suspended Tasks
 SUSPENDHWM    ===> 00000                   Suspended Tasks HWM
 APPLDAT       ===> YES                     Register Application Data

 PF 3 END                                                      12 CNCL
```

- Set system setting



```
EZAO,SET,CICS                                               APPLID = CICS1A

 Overtype to Enter

 TRACE         ===> YES                     Trace CICS Sockets
 TCBLIM        ===> 00000                   Open API TCB Limit
 APPLDAT       ===> YES                     Register Application Data


 PF 3 END                                                      12 CNCL
```

The operator can query the status of the interface automatically registering application data by using the EZAO,INQUIRE,CICS transaction. The operator can also control whether the interface automatically registers application data by using the EZAO,SET,CICS transaction.

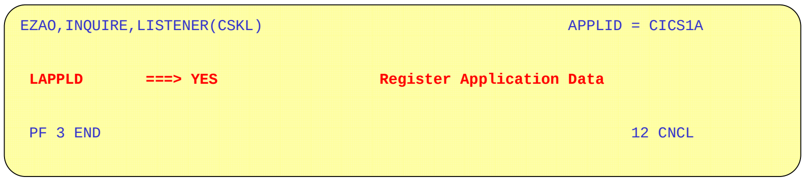
The following messages can be produced by the EZAO transaction when processing the APPLDAT configuration option.

Error on APPLDAT Entry, Please Re-Enter (Re-enter parameters or press ENTER to continue.)
CICS Sockets application data registration is already YES (Press ENTER to continue or PF3 to exit)
CICS Sockets application data registration is already NO (Press ENTER to continue or PF3 to exit)
Automatic application data registration is now enabled

Automatic application data registration is now disabled

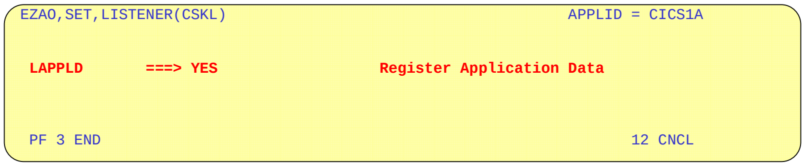
Query or Set a specific listener using EZAO transaction

- Query a specific listener setting



```
EZAO,INQUIRE,LISTENER(CSKL)                                  APPLID = CICS1A

 LAPPLD       ===> YES                 Register Application Data

 PF 3 END                                                            12 CNCL
```

- Set a specific listener setting



```
EZAO,SET,LISTENER(CSKL)                                      APPLID = CICS1A

 LAPPLD       ===> YES                 Register Application Data

 PF 3 END                                                            12 CNCL
```

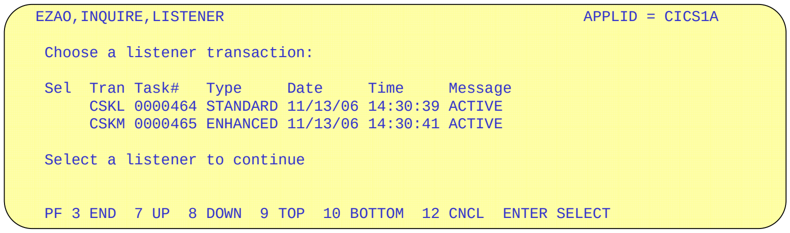
The operator can query the status of the listener automatically registering application data by using the EZAO,INQUIRE,LISTENER transaction. The operator can control whether the listener automatically registers application data.

The following messages will be produced by the EZAO transaction when processing the LAPPLD configuration option.

Error on LAPPLD Entry, Please Re-Enter (Re-enter parameters or press ENTER to continue)
Listener application data registration is already YES (Press ENTER to continue or PF3 to exit)
Listener application data registration is already NO (Press ENTER to continue or PF3 to exit)
Automatic application data registration is now enabled
Automatic application data registration is now disabled

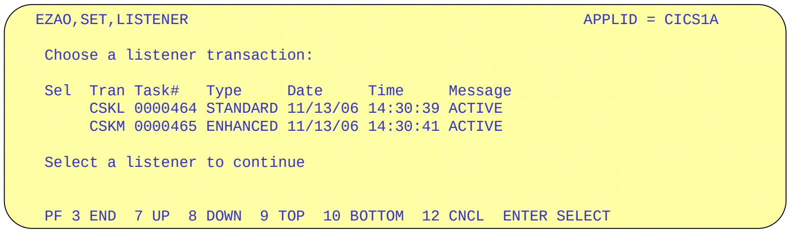
Query or Set an active listener using EZAO transaction

- Select an active listener to query



```
 EZAO,INQUIRE,LISTENER                                          APPLID = CICS1A

  Choose a listener transaction:

  Sel  Tran Task#   Type     Date     Time     Message
       CSKL 0000464 STANDARD 11/13/06 14:30:39 ACTIVE
       CSKM 0000465 ENHANCED 11/13/06 14:30:41 ACTIVE

  Select a listener to continue


  PF 3 END  7 UP  8 DOWN  9 TOP  10 BOTTOM  12 CNCL  ENTER SELECT
```

- Select an active listener to set



```
 EZAO,SET,LISTENER                                              APPLID = CICS1A

  Choose a listener transaction:

  Sel  Tran Task#   Type     Date     Time     Message
       CSKL 0000464 STANDARD 11/13/06 14:30:39 ACTIVE
       CSKM 0000465 ENHANCED 11/13/06 14:30:41 ACTIVE

  Select a listener to continue


  PF 3 END  7 UP  8 DOWN  9 TOP  10 BOTTOM  12 CNCL  ENTER SELECT
```

The EZAO operator transaction now supports a list of active listeners to choose from when querying the status of a listener automatically registering application data. The operator transaction also supports a list of active listener to choose from when controlling whether a listener automatically registers application data.

The following messages will be produced by the EZAO transaction when selecting active listeners.

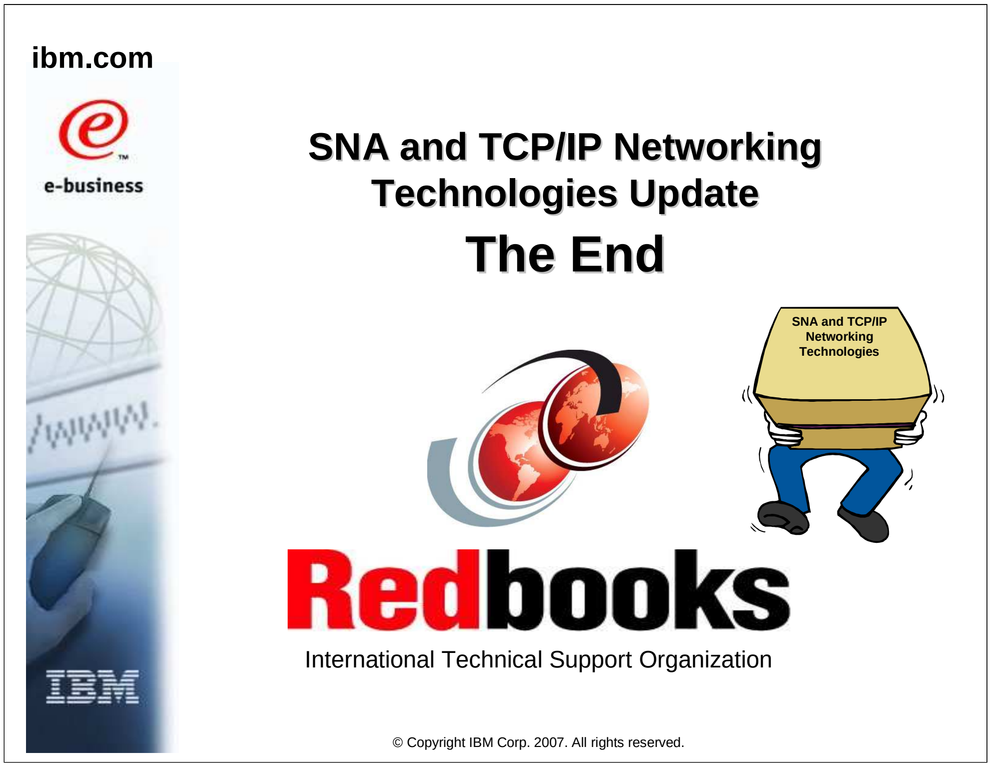
Select a listener to continue

No listeners active

Listener is no longer active, select another listener

At the top of data

At the bottom of data



ibm.com

SNA and TCP/IP Networking Technologies Update

The End

Redbooks

International Technical Support Organization