



Communications Server z/OS V1R5 and V1R6 Technical Update

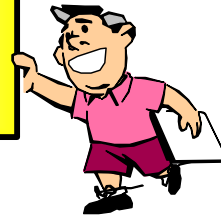
Overview of CS z/OS

© Copyright International Business Machines Corporation 2004. All rights reserved.



- z/OS V1R5
 - ┆ MAXSOC limit raised from 2000 to 64K
 - ┆ Asynchronous socket IO performance enhancement
 - ┆ Dynamic MVS system symbol resolution
 - ┆ SMTPNJE to use hostname as IPMAILER

- z/OS V1R6
 - ┆ SMTPNJE to recover from bad spool files (Abend S001)
 - ┆ Policy agent restructure



MAXSOC limit



- MAXSOC limit was prior to z/OS V1R5 2,000
 - ┆ MAXSOC defines how many sockets the application have have open concurrently - implicitly, it defines the maximum socket descriptor number this program will ever be given
 - ┆ In a concurrent server where the listener process and the child processes operate as independent socket tasks, NUMSOC does not impose a limit on how many concurrent connections can be active with the server.

- Need to support 64K sockets in:
 - ┆ the assembler macro sockets API
 - ┆ the REXX sockets API
 - ┆ the CICS sockets APIs
 - ┆ the IMS sockets API
 - ┆ and in the callable Sockets Extended API

- Need to limit support in the C Sockets API for CICS to 2000 sockets.

- Update the macro API so that MAXSOC values up to 65535 and socket numbers up to 65534 are supported.

- Update the REXX API so that MAXDESC values and socket numbers up to 65535 are supported.

- The new 64K sockets support in the macro API indirectly adds 64K sockets support to the CICS, IMS, and Sockets Extended APIs.

- Update the C Sockets API for CICS so that socket numbers greater than 2000 are not accepted.

© Copyright International Business Machines Corporation 2004. All rights reserved.

MAXSOC limit C sockets - LE, TCP/IP, and CICS



➤ C Sockets API

f The C Sockets API supplied by LE supports 64K sockets except for its select() function which can only monitor up to 32767 sockets.

f The C Sockets API supplied by the z/OS Communications Server (CMIUCSOC) still supports only up to 2000 sockets.

f Because the C Sockets API for CICS uses many of the same header files as the standard Communications Server C API, it is limited to 2000 sockets as well. The initapi() function in EZACIC07 and EZACIC17 has been amended to return -1 and an ERRNO of 10198 (EIBMTOOMANYSOCKETS) if the max_sock value exceeds 2000.

MAXSOC limit BPXPRMxx and storage use



➤ BPXPRMxx Parameters

┆ MAXFILEPROC

- If there will be applications exploiting the enhancement that allows more than 2,000 sockets to be opened, then programmers need to examine and potentially modify the MAXFILEPROC parameter in the BPXPRMxx member. They would need to determine the highest possible number of combined sockets requested by applications within a single UNIX System Services process and set MAXFILEPROC to that value.
- MAXFILEPROC can be overridden in a user's OMVS segment in RACF
- Before z/OS V1R6, the limit for MAXFILEPROC was 64K, z/OS V1R6 raises that to 128K

┆ MAXSOCKETS

- If there will be applications exploiting the enhancement that allows more than 2,000 sockets to be opened, then programmers may need to examine and potentially modify the MAXSOCKETS values in the NETWORK statements in the BPXPRMxx member. For each addressing family (AF_INET, AF_INET6, etc.), they would need to determine the highest combined possible number of sockets in the addressing family that can be opened by all applications in the system, and specify that number as the MAXSOCKETS value.
- MAXSOCKETS is a UNIX System Services limitation, the TCP/IP stack itself has no limits on number of open sockets

➤ Storage Concerns

- ┆ The macro API allocates 68 bytes for each potential socket. Thus, if a INITAPI is issued with MAXSOC=65535 then 4352 Kbytes of storage are allocated just for the socket array
- ┆ When an asynchronous SELECT is issued, an OE polling array is created which contains 8 bytes of storage for each socket being monitored by the SELECT. Thus, if the SELECT is monitoring 65535 sockets then 512 Kbytes of storage are allocated just for the polling array.
- ┆ There are no storage concerns when specifying large values for MAXSOCKETS and MAXFILEPROC

© Copyright International Business Machines Corporation 2004. All rights reserved.

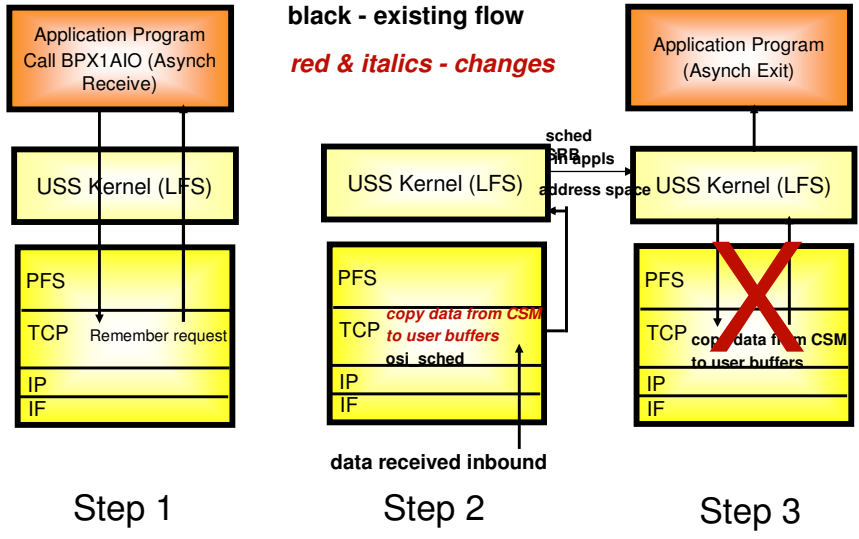
Asynchronous socket IO performance improvement



- The z/OS UNIX System Services Asynchronous Sockets API is very flexible and provides for a very scalable implementation of TCP/IP-based servers that handle a large number of connections. Its major drawback is that it is more expensive, in CPU utilization and throughput, than the synchronous sockets API.
- Using either the UNIX System Services Callable Services Sockets API or the LE C/C++ Sockets API, a stream socket application that issues an asynchronous `recv()`, `read()`, or `readv()` can specify in the AIOCB that its read buffers reside in common storage. The TCP/IP stack will copy newly-received data directly into those buffers and will notify UNIX System Services that the receive operation is complete. This eliminates an entry into the MVS Dispatcher to schedule an SRB to the application's address space to copy the received data and complete the receive operation.
 - /NOTE: The buffers may reside in any form of commonly addressible storage; for example, CSM CSA, CSM dataspace, CSA/ECSA, CADS.
- To take advantage of this performance enhancement, a stream socket application must:
 - /Execute in supervisor state or system key, or be APF authorized, or be assigned superuser status.
 - /Allocate its read buffers in CSM-managed storage, and not free or reuse them until the asynchronous receive request has completed or has been canceled.
 - /Set the `AioCommBuff` flag in the AIOCB for an asynchronous `recv()`, `read()`, or `readv()` socket request.

© Copyright International Business Machines Corporation 2004. All rights reserved.

TCP/IP Asynchronous I/O flows



© Copyright International Business Machines Corporation 2004. All rights reserved.



NOTES

- Step 1
 - ⌋ Application issues receive
 - ⌋ Now application can inform TCP/IP that receive buffers are in common storage
- Step 2
 - ⌋ Data received inbound
 - ⌋ In existing flow, data cannot be copied because buffers are in application's address space, and TCP/IP cannot access them from TCP/IP's address space.
 - ⌋ In the new flow, the application has allocated and informed TCP/IP that the receive buffers are in common storage, so TCP/IP can move the data to the application buffers before scheduling the SRB.
- Step 3
 - ⌋ When the SRB is scheduled in the application's address space, there is now no reason to call TCP/IP to copy the buffers, so this entire call is eliminated. The asynch I/O exit is driven with the data.

- This support is for high-performance subsystems, such as DB2.
 - ⌋ Normal socket application programs would typically not use this type of support.



➤ Problem:

Automatic resolution of MVS system symbols was not supported for the Resolver setup file nor for the TCPIP.DATA file. It was necessary to use the EZACFSM1 utility program to resolve MVS system symbols for those files.

➤ Solution:

Support automatic resolution of MVS system symbols in the Resolver setup file and in the TCPIP.DATA file. Symbols (such as &SYSNAME, &SYSPLEX, etc.) are resolved as the file records are read.

SMTPNJE hostname as IPMAILER



➤ Problem:

/ Currently SMTPNJE (pascal SMTP program) supports a single static IP address (IPMAILERADDRESS) which could be set in order to forward mail in the IP network. This single configuration option was too limited in scope. Multiple IP addresses and dynamic update of IP addressing information were needed.

➤ Solution:

/ Add a new statement to the SMTPNJE configuration data set (IPMAILERNAME). Using the information provided on this statement, SMTPNJE will use name resolution to retrieve the IP addressing information from the name server.

/ PTFed backwards

- CSV1R2 - APAR PQ71580
- CSV1R4 - APAR PQ73161

➤ Configuration:

/ The new statement in the SMTPNJE configuration data set is IPMAILERNAME which has no defaults when coded. You must specify a fully qualified mailer name or a mailer hostname and a routing instruction indicating which mail to forward (ALL or UNKNOWN).

Syntax

```
>>-----IPMAILERNAME-----<mailername>-----ALL----->
|----- UNKNOWN-----|
```

SMTPNJE to recover from Abend S001 when reading bad spool files



- SMTP is written in PASCAL and uses some of the PASCAL runtime functions to do I/O. Some of these functions are readln(), writeln() and reset().
- At a lower layer, SMTP retrieves job from the JES spool using the Subsystem Interface (SSI Function Code 1).
- Badly developed applications writing to the JES spool have caused the SMTP mailer to abend with S001 abends when they had created bad spool file data.
 - ┆ Such a situation took SMTPNJE down
 - ┆ When restarting SMTPNJE, the S001 abend would re-occur if the bad spool file hadn't been manually identified and deleted before restarting SMTPNJE
 - ┆ In a MAS complex where multiple SMTPNJE servers read off the shared spool data set, there had been cases where all servers reading off that spool data set abended
- A previous release had added support to detect when a spool file was empty and recover from that situation, but if the spool file had inconsistent record structures, an abend S001 would occur.
- A SYNAD exit has in z/OS V1R6 been added to the PASCAL runtime support code to handle the I/O error and prevent the ABENDS001 from occurring.
- Added an interface between the SYNAD exit and the SMTPNJE server function so that SMTPNJE can determine when the SYNAD exit has been invoked.
- Add a new SMTP configuration statement to allow SMTPNJE to delete JES spool file that when accessed would have resulted in an ABENDS001.
- If the new configuration statement is NOT coded, allow SMTPNJE to stop gracefully rather than abend.

© Copyright International Business Machines Corporation 2004. All rights reserved.

SMTPNJE to recover from Abend S001 when reading bad spool files



- New SMTP configuration statement is supported:
 - ┆ DELETEBADSPoolFILE Statement

- Use the DELETEBADSPoolFILE statement to change the behavior of SMTP when it detects a spool file on the JES spool that would cause a ABENDS001 on the JES spool.
 - ┆ The default behavior if this statement is not coded is SMTP will generate error message EZA5469E and terminate. This will give the system administrator a chance to look at the offending spool file.
 - ┆ The application generating the spool file will need to be changed. While that is being done, the DELETEBADSPoolFILE statement can be used so SMTP will continue running and automatically delete any offending spool file. Note that these spool files may contain customer data and therefore it is the system administrator's responsibility to give SMTP permission to do automatic deletion.
 - ┆ If a spool file is automatically deleted, SMTP will generate message EZA5470E to alert the user.

- Syntax

```
> >>-----DELETEBADSPoolFILE-----  
  
Parameters  
None
```

Policy Agent restructure



- Policy Agent developed/deployed on several different platforms.
 - ┆ Although similar, each platform version has unique function and behavior.
 - ┆ Difficult to add new functions across multiple platforms.

- Single, platform-independent version of Policy Agent.
 - ┆ Developed using z/OS V1R5 as the base.
 - ┆ Common version allows for platform-specific portion.
 - ┆ z/OS V1R6 is a platform-specific version of the common base.

- Common version includes a superset of all policy attributes supported on all platforms - not all of those are supported by z/OS, even though the schema definition and sample files may contain them. Refer to IP Configuration Reference for supported attributes.

- Common version includes minor external changes, including a memory tracing function:
 - ┆ New start option to activate memory tracing.
 - ┆ New debug level to activate inline memory logging.
 - ┆ New MODIFY command to take a memory use snapshot.
 - ┆ New PEPInstance statement added as a synonym for the existing TcpImage statement.
 - Statement name may be more meaningful on different platforms.
 - PEP refers to "Policy Enforcement Point", which is the primary policy-related job of the TCP/IP stack.

```
                | -NOFLUSH- | | -NOPURGE- | | -1800- |
>>---| -TcpImage---| -name- | ------| -|-----| -|-----| ---<<
      | -PEPInstance-|      | -path-| | -FLUSH---| | -PURGE---| | -i-----|
                | -NOFLUSH- | | -NOPURGE- |
```

© Copyright International Business Machines Corporation 2004. All rights reserved.

Warning



➤ z/OS V1R6 is the last release where the old ORoutedD server is shipped.

⌘ If continuing to use RIP - migrate to OMPROUTE RIP

© Copyright International Business Machines Corporation 2004. All rights reserved.

Warning



➤ With z/OS V1R6, IBM MVS TCP/IP Enterprise-specific MIB will no longer support SMIv1

- ⌘ SMIv2 has been around for 10 years
- ⌘ Standard MIBs no longer support SMIv1
- ⌘ Difficult to maintain SMIv1 versions of the MIB since many of the new data types and textual conventions are not supported (ex: 64bit counters)
- ⌘ Conversion tools are commonly available if still need to use SMIv1

© Copyright International Business Machines Corporation 2004. All rights reserved.

Trademarks, Copyrights, and Disclaimers



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

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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 registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

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

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

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

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005. All rights reserved.

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

© Copyright International Business Machines Corporation 2004. All rights reserved.