

4th European GSE / IBM TU for z/VSE, z/VM and Linux on System z

25.-27. Oktober 2010, München

VS05– z/VSE: Hints & Tipps

Torsten Röber (roeber@de.ibm.com) IBM Deutschland GmbH
Dagmar Kruse (dkruse@de.ibm.com) IBM Deutschland GmbH
Heinz Peter Maassen (hp.maassen@lattwein.de) Lattwein GmbH

26.10.2010

Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/390, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries. Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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 logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput 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 improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten



Kurz notiert: DB2 Hints & Tips



- **DB2 Linux für System z 9.1 & 9.5**
 - LI74523 enthalten ab FP5 (9.5) bzw. FP8 (9.1)
 - ‚flush package cache dynamic‘
- **DB2 Fixpack 3 (9.7) und 6 (9.5)**
 - 9.7 FP3 ‚Special Build‘ bzw. FP3a
 - 9.5 FP6 ersetzt durch FP6a
- **DB2RESILIENCE = ON (9.7)**
 - DB2_THREAD_SUSPENSION
 - Hang beim DB2STOP
 - IC69960 ab FP3 enthalten
- **DB2 Linux (ab 9.5)**
 - Process based → Thread based
 - SWAPINESS → 0 oder 1 (Default 60)



Kurz notiert: Problem Determination



Nützliche Broschüre „**Problem Determination under z/VSE**“

- Übersicht der **gängigsten Trace und Dump Optionen** im z/VSE und CICS Umfeld
- Mögliches Setup Ihres z/VSE, um im Fehlerfalle **die bestmögliche Fehlerdokumentation an den IBM Software Support** zu liefern.
- Hinweise zum **Dump Transfer**
- Erstellt von
Benno Sauer,
IBM Deutschland GmbH,
Technical Support Competence Center (TSCC Software)
E-Mail: Bsauer1@DE.IBM.COM
- Zu finden auf der z/VSE-Homepage unter:
<http://www-03.ibm.com/systems/z/os/zvse/support/problem.html>



Kurz notiert: COPY-Blocks überprüfen (1/ 2)



- „Copy-Blocks“ sind Supervisor-Puffer für die I/O-Verarbeitung
- Anzahl wird in der IPL-Prozedur mit Befehl „SYS BUFSIZE“ festgelegt. Default: 1500

```
...  
DEF SYSREC=SYSWK1  
SYS BUFSIZE=1500  
SYS NPARTS=44  
...
```

Das reicht für einige Produktionssystemen nicht aus !

- Stärkere I/O-Auslastung der z/VSE-Systeme
(Plattenzugriffe, Terminal-Sessions,...)
- ➔ Performance-Einbußen / System-Engpässe



Kurz notiert: COPY-Blocks überprüfen (2/ 2)



- Regelmäßiges Überprüfen der COPY-Blöcke mit dem SIR-Befehl:

```
SIR
...
AR 0015 DYN.PARTS = 00001    HIGH-MARK = 00002    MAX = 00032
AR 0015
AR 0015 COPY-BLKS = 00006    HIGH-MARK = 00077    MAX = 01544
AR 0015 CHANQ USED= 00018    HIGH-MARK = 00021    MAX = 00255
...
```

- Empfehlung:
ca.10-20% freien Copy-Blöcken je nach Environment

➔ evtl. „SYS BUFSIZE“ in IPL-Prozedur erhöhen,
ABER Achtung!



- Supervisor wird dadurch vergrößert
(+ 100 COPY-Blocks → + ca. 7K beim Supervisor)
- Shared 24-Bit- Bereich kann sich dadurch um 1 MB erhöhen!

7

4th European GSE / IBM TU for z/VSE, z/VM and Linux on System z

26.10.2010

Kurz notiert: TCP/IP 15F Service-Stand



- Neues Sammel-APAR für **Anfang/Mitte November** geplant
- Aktuelles **Sammel-APAR**: **PM18076** vom 07/2010
 - PTF UK58621 for z/VSE 4.2 (Release 0TP)
 - PTF UK58624 for z/VSE 4.1 (Release 9TP)In diesem APAR sind ZAPs bis zum 21.06.2010 enthalten.

**Die PTFs enthalten nur Änderungen,
vorherigen PTFs sind Prereq's**

- **Online Message Explanation**: **PM18149** vom 07/2010
 - PTF UK58665 for z/VSE 4.2 (Release 0TP)
 - PTF UK58666 for z/VSE 4.1 (Release 9TP)

8

4th European GSE / IBM TU for z/VSE, z/VM and Linux on System z

26.10.2010



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten



TCP/IP 1.5.F Programming Styles



- Mini Umfrage:
 - Wer hat schon ein Programm mit einem TCP/IP API geschrieben oder kennt sich mit der Kommunikation zwischen heterogenen Systemen aus?
 - Ohne Frage: Sehr wenige.
 - Es gibt einige Fehler in den API's, die bis heute noch niemandem aufgefallen sind. Daher die Vermutung:
 - **Kaum jemand hat eigene programmierte TCP/IP Anwendungen.**

- Folgende Application Programming Interfaces werden mit CSI TCP/IP unterstützt:
 - 1. EZASMI - Das ist ein Assembler Interface mit einem großem Funktionsumfang. Laut Handbücher größtenteils z/OS kompatibel Jedoch in der Praxis leider nur im Batch. **Es gibt keine Unterstützung von EZASMI im CICS TS für z/OS – daher nicht kompatibel !**

Beispiel:

```

EZASMI TYPE=INITAPI,MAXSOC=MAXSOC,MAXSNO=MAXSNO,ERRNO=ERRNO, *
      ASYNC='NO',RETCODE=RETCODE,SUBTASK=NSTASK
. . .
EZASMI TYPE=GETCLIENTID,CLIENT=SCLIENT, *
      ERRNO=ERRNO,RETCODE=RETCODE,ECB=SERVECB
. . .
EZASMI TYPE=SOCKET,AF='INET',SOCTYPE='STREAM', *
      PROTO=PROTOCOL,ERRNO=ERRNO,RETCODE=RETCODE,ECB=SERVECB
. . .
EZASMI TYPE=BIND,S=SOCKET1+2,NAME=MYNAME, *
      ERRNO=ERRNO,RETCODE=RETCODE,ECB=SERVECB
    
```

- 2. EZA CALL Interface. Die Programmierunterstützung für Cobol, PL/I und Assembler. Im Prinzip arbeitet das EZA Call API genau so wie das EZASMI API jedoch werden die Befehle mit einem CALL aufgerufen. Laut Beschreibung auch für z/OS fast 100 % kompatibel.
 - Nachteil: Alle Parameter beim CALL müssen in einer fest vorgegebenen Reihenfolge definiert werden. Die Reihenfolge spielt keine Rolle beim EZASMI API.
 - Vorteil: Geht nicht nur mit Assembler sondern mit allen Sprachen die einen Standard CALL unterstützen.

Beispiel:

```

MVC SOCKFUNC,=CL16'INITAPI'
MVC MAXSOC,=H'0'          SOCKETNUMBER = 0
. . .
CALL EZASOKET,(SOCKFUNC,MAXSOC,IDENT,MYSUBT,MAXSNO,ERRNO, *
      RETCODE),VL
. . .
MVC COMND,=CL8'SOCKET'    SET COMMAND NAME FOR TRACE
CALL EZASOKET,(SOCKFUNC,AF,SOCTYPE,PROTOCOL,ERRNO, *
      RETCODE),VL
    
```



TCP/IP 1.5.F Programming Styles



- **3. SOCKET Assembler API** – nur möglich für Assembler und leider nur **Quasi Reentrant**. Beim ersten Socket Makro wird DSECT Code ins Programm eingefügt, beim OPEN wurde auch das Basis Register 4 zerstört (* bis 1.5E). Das kann zu Problemen führen.

Beispiel:

```
* * OPEN A PASSIVE CONNECTION...  
  
SOCKET  OPEN,TCP,          OPEN USING TCP PROTOCOL          X  
        PASSIVE=YES,      JUST WAIT AROUND                  X  
        LOPORT=TCPPORT,   USE THIS LOCAL PORT NUMBER       X  
        DESC=TSKTOKEN,    SOCKET DESCRIPTOR(1 FWORD)      X  
        USESYS=YES,       USE THE OPTION SYSPARM CARD       X  
        ECB=TSKRSLT      SOCKET ECB(14 FWORDS)
```



TCP/IP 1.5.F Programming Styles



- **4.** Das EXEC Level API, das auf Basis des Socket API's aufbaut. Diese Schnittstelle generiert Socket Code auf Grund von EXEC Befehlen. Das funktioniert ähnlich wie der Command Level API im CICS. Jedoch nur bis zur Version 1.5.E (IPNETRAN).

Beispiel:

```
EXEC  TCP OPEN FOREIGNPORT(0)          X  
      FOREIGNIP(0)                    X  
      LOCALPORT(SERVPORT)              X  
      RESULTAREA(RSLTAREA)             X  
      DESCRIPTOR(CLNOKEN)              X  
      PASSIVE                           X  
      WAIT(YES)                         X  
      ERROR(ERROROP)
```

TCP/IP 1.5.F Programming Styles



- 5. Ab 1.5F wird mit IPNETPRE ein eigener CALL erzeugt. Dadurch werden keine Benutzer Register mehr zerstört- und der generierte Code ist voll reentrant. Leider sollte jedem Befehl ein "END-EXEC." hinzugefügt werden. Jedes Programm muss das Statement: XOBLOK DSECT=NO den Definitionen hinzufügen.

Beispiel:

```

SERVPORT DS CL2
          XOBLOK DSECT=NO
. . .
EXEC TCP OPEN FOREIGNPORT(0) X
      FOREIGNIP(0) X
      LOCALPORT(SERVPORT) X
      SYSID(TSYSID) X
      RESULTAREA(RSLTAREA) X
      DESCRIPTOR(CLNTOKEN) X
      PASSIVE X
      WAIT(YES) X
      ERROR(ERROROP) X
          END-EXEC.
    
```

TCP/IP 1.5.F Programming Styles



- Leider fehlen in der hierzu gehörigen DSECT (XOBLOK) einige Definitionen. Deshalb werden Assembler Fehler erzeugt. Der folgende Block fehlt auch beim aktuellen Download! **Das ZP15F176 ist nicht mehr verfügbar!**

```

* Fields added at ZP15F176.
XOIDENT DS CL6'XOBLOK' Eyecatcher/verification field
XOBLKLEN DS AL2(XOBLKLN) Length of XOBLOK (set by IPNETPRE)
XOAPIV1 DS CL8 IPNETPRE version (vvrrpzzz)
* VV - Version
* RR - Release
* P - Service pack
* zzz - Fix number
* eg, 1.5 F GA == 0105F
* 1.5 F with ZP15F432 == 0105F432
XOAPIV2 DS CL8 IPNETXCC (etc) version (as above)
XOREG0 DS F Reg 0 Reason code area
XOREG1 DS F Reg 1 Information return area
XODEBUG DS X Y - Debugging mode
          DS X Reserved
XOCALLID DS H Unique ID. Identifies calling loc
XOBLKLN EQU *-XOBLOK
          MEND
/+
    
```




TCP/IP 1.5.F Programming Styles



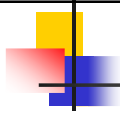
Folgende ZAPS sind bei Assembler und IPNETPRE noch zu installieren:

- ZP15F552 **IPNETPRE.L settings are not overridden by EXEC card parameters**
 - When the API preprocessor is invoked, it reads IPNETPRE.L, if present, to obtain default values for settable parameters. These settings can then be overridden by user-specified values. However, the PARM field values are being processed before IPNETPRE.L values are applied.
- ZP15F550* **CICS Assembler Socket program issues an XOBLOK error message**
 - For those users who write CICS Assembler code and use the API (IPNETPRE generated code) rather than the SOCKET macro, the CLOSE call fails to pass a pointer to the XOBLOK block. The subroutine that processes the request (IPNETXC) then reports an XOBLOK error.
- Am besten alle EXEC TCP Befehle mit einem "END-EXEC." beenden.

```

EXEC TCP OPEN FOREIGNPORT(0) X
FOREIGNIP(0) X
LOCALPORT(SERVPORT) X
SYSID(TSYSID) X
RESULTAREA(RSLTAREA) X
DESCRIPTOR(CLNTOKEN) X
PASSIVE X
WAIT(YES) X
ERROR(ERROROP) X
END-EXEC.

```



TCP/IP 1.5.F Programming Styles



Noch ein Tipp:

Beim EXEC TCP Close darf **kein** Parameter WAIT spezifiziert sein. Das gilt für IPNETPRE. GETVIS Storage exhausted. Das Ergebnis ist nicht vorhersehbar:

```

EXEC TCP CLOSE X
RESULTAREA(RSLTAREA) X
DESCRIPTOR(CLNTOKEN) X
WAIT(YES) X
ERROR(ERRORCL) X
END-EXEC.

```

IPNETRAN:
1.5E ok.

Ändern in:

1.5F WAIT (YES) darf nicht verwendet werden! Dann ok.

```

EXEC TCP CLOSE X
RESULTAREA(RSLTAREA) X
DESCRIPTOR(CLNTOKEN) X
ERROR(ERRORCL) X
END-EXEC.

```

IPNETPRE:
1.5F ok.



TCP/IP 1.5.F Programming Styles



Mit IPNETPRE umgewandelte Programme enthalten keine Socket DSECTS mehr. Deshalb können die Socket Befehle nicht mehr einfach verwendet werden. Beispiel ändern der SYSID bei Anwendungen:

```
CLC SOGNL+2+6(2),TSYSID
BE SKIPSS
SOCKET SET_SYSID,TSYSID
SKIPSS DS 0H
```

Ändern in:

```
EXEC TCP OPEN FOREIGNPORT(0) X
FOREIGNIP(0) X
LOCALPORT(SERVPORT) X
SYSID(TSYSID) X
RESULTAREA(RSLTAREA) X
DESCRIPTOR(CLNTOKEN) X
PASSIVE X
WAIT(YES) X
ERROR(ERROROP) X
END-EXEC.
```



TCP/IP 1.5.F Programming Styles



Oder andere nur mit Socket Assembler API codierte Funktionen müssen mit ausgelagerten Programmen / Routinen gelöst werden. Beispiele hierzu:

Command	Input	Output
GETHOSTBYADDR	IPv4 address in EBCDIC text characters.	Blank-delimited domain name
GETHOSTBYNAME	Domain name in EBCDIC text characters.	19 bytes of data containing <ul style="list-style-type: none"> • 4-byte binary IPv4 address • 15-byte text IPv4 domain address (nnn.nnn.nnn.nnn)
GETHOSTNAME	None	Blank-delimited text name of the local VSE system
GETHOSTID	None	19 bytes of data containing <ul style="list-style-type: none"> • 4-byte binary IPv4 address • 15-byte text IPv4 address of the local VSE system



TCP/IP 1.5.F Programming Styles



- Das EXEC Level API ab TCP/IP Version 1.5.F generiert vollkommen neuen Code. Dieser Code ruft intern über Call eine interne neue Schnittstelle zum TCP/IP auf.
- Leider ist der Source Code nicht mehr kompatibel zur Version 1.5E. Deshalb gibt es auch beide Preprozessoren ab der Version 1.5F.
- 6.** Das BSD Socket Interface ist ein Standard der in den 80-er Jahren dem BSD Unix API nachempfunden wurde. Ursprünglich wurde dieses BSD Socket API für die Sprache C entwickelt. Von CSI ist das aber für jede Programmiersprache verfügbar.

Beispiel:

<pre> ** BSD-C SOCKET() CALL INVOKES IPNRSOCK... LA R1,DOMAIN @ OF DOMAIN ST R1,PARM1 STORE IT LA R1,TYPE @ OF TYPE ST R1,PARM2 STORE IT LA R1,PROTOCOL @ OF PROT ST R1,PARM3 STORE IT LA R1,PARMS @ OF PARM L R15,=V(IPNRSOCK) @ SOCKET BALR R14,R15 GO FOR IT </pre>	<pre> Als C Code: int s; char *name; ... /* Get stream socket in Internet domain with default protocol */ s = socket (AF_INET, SOCK_STREAM, 0); ... </pre>
---	---



TCP/IP 1.5.F Programming Styles



- 7.** Es gibt 2 Arten von REXX Unterstützung im TCP/IP for VSE/ESA:
 - REXX Unterstützung innerhalb von TCP/IP for VSE/ESA. (d.h. REXX Socket API und wurde von CSI mit der Version 1.30 unterstützt). Die Dokumentation findet man im TCP/IP for VSE 1.5 Programmer's Reference manual.
 - REXX/VSE Socket API Unterstützung in REXX/VSE wird im einzelnen im *REXX/VSE Reference*, SC33-6642 beschrieben.

Beispiel REXX Socket API:

```

/* Open a control connection to the TCP/IP for VSE partition*/
rc = SOCKET('CONTROL','OPEN')
if rc \= 0 then say 'rc='rc 'errmsg='errmsg
/* Send the command string. */
rc = SOCKET(handle,'SEND','GETHOSTID')
if rc \= 0 then say 'rc='rc 'errmsg=' errmsg
/* Receive the response from TCP/IP for VSE */
rc = SOCKET(handle,'RECEIVE')
if rc \= 0 then say 'rc='rc 'errmsg='errmsg
/* Close the control connection */
rc = SOCKET(handle,'CLOSE')
if rc \= 0 then say 'rc='rc 'errmsg=' errmsg

```

Beispiel: REXX/VSE Socket API:

```

/* rexx procedure: socket server procedure */
rc = 0
/* initialize socketset */
fc = SOCKET('INITIALIZE','SERVMIRR',,,,'SSLV3','CRYPTO.KEYRING',86400)
parse var fc socket_rc .
if socket_rc ^= 0
then do
    say 'INITIALIZE failed with return info ' fc
    exit 99
end
/* create a TCP socket for client connection requests */
fc = SOCKET('SOCKET','AF_INET','SOCK_STREAM','IPPROTO_TCP')
parse var fc socket_rc newsocketid
if socket_rc ^= 0
then do
    say 'SOCKET failed with return info ' fc
    fc = SOCKET('TERMINATE')
    exit 99
end
end
    
```

Fazit:

Am besten verwendet man die Programmier Techniken, die von IBM zur Verfügung gestellt werden.

EZASMI eignet sich nur für Assembler, ist aber nicht im CICS/TS für z/OS unterstützt.

EZA-Socket Call API für COBOL, PLI, Assembler und C falls man auch Anwendungen schreibt, die ins z/OS portiert werden sollen.

Falls man den IBM C for VSE/ESA Release 1 Compiler zusammen mit dem IBM Language Environment for z/VSE (LE/VSE) 1.4 C Run-time verwendet, so sollte man das LE/VSE 1.4 Socket Interfaces benutzen. Hier wird die Kompatibilität zum OS/390 X/Open (XPG4.2) Compliant Socket Interfaces garantiert.

Weniger geeignet sind der EXEC Level API und das CSI Socket API.



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- **TCP/IP 1.5.G IPV6**
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten



TCP/IP 1.5.G IPV6



- **Download TCP/IP for VSE 1.5G Early Release Program**

Downloads

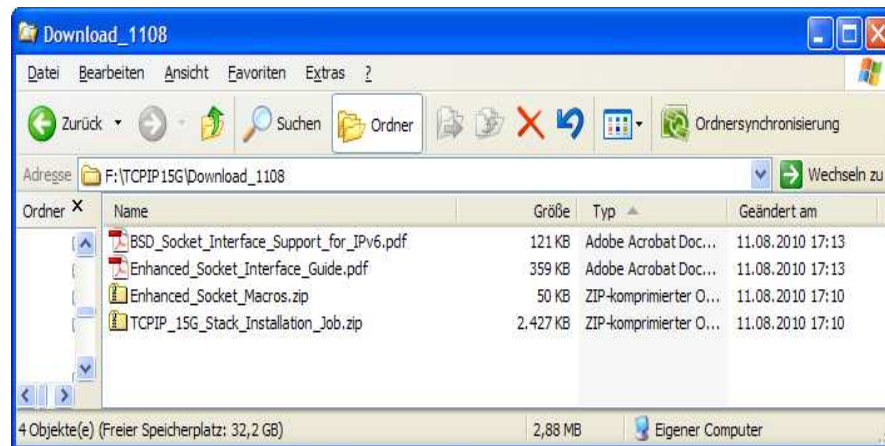
These download links provide access to the current development version of TCP/IP for VSE 1.5G. A major component of this release paves the way for integrated IPv6 support. This includes an Enhanced Socket Interface to support an expanded parameter list, along with many additional services, functions, and diagnostic tools.

While we continue work on the stack itself, we are making the socket API available for customer and developer testing. This means that although you can explore the Enhanced Socket Interface now, most of the stack-based daemons and servers (for example, FTP, GPS, and TN3270) will not be available until later.

Currently, all link adapters are functional, including the use of an IPNET link to connect with 1.5F stacks. We will update the available materials frequently as we progress toward open beta and GA status.

Downloads	Last Updated
1.5G Stack Installation job (ZIP)	6/14/2010
Enhanced Socket Macros (ZIP)	6/14/2010
Enhanced Socket Interface Guide (PDF)	6/14/2010
BSD Socket Interface Support for IPv6 (PDF)	6/14/2010

- Dateien zu TCP/IP for VSE 1.5G Early Release Program



- BSD_Socket_Interface_Support_for_IPv6 enthält die Dokumentation, wie das CSI BSD Socket Interface IPV6 unterstützt.
- Dieses Interface unterstützt C, Assembler und Cobol, sowohl im CICS als auch im Batch.
- Im \$SOCKOPT.PHASE muss der Parameter BSDXPHS = IPNRBSDC (für IPV4) auf BSDXPHS = IPNRBSDI für IPV6 geändert werden. (Partiton weise über LIBDEF steuern!)
- Die neue Phase IPNRBSDI unterstützt sowohl IPV4 wie auch IPV6.
- Eine IPV6 Adresse wird mit 16 Stellen definiert. Eine IPV4 Adresse ist 4 Stellen lang.



TCP/IP 1.5.G IPV6



- Die Struktur der IP Adressen wurde für IPV4 geändert:

SOCKADDR_IN Format für 4.4BSD IPv4

USOCKIN	DSECT	
ULENGTH	DS	XL1
UFAMILY	DS	XL1
UPORT	DS	XL2
UIPADDR	DS	XL4
UNUSED	DS	XL8

Das Feld ULENGTH enthält die Länge der Struktur und das Feld UFAMILY enthält den Hexwert x02 als Kennung für IPV4 Protokoll. Zusätzlich enthält die Struktur SOCKADDR_IN 8 Stellen UNUSED, der leider eine 16-stellige IPV6 Adresse nicht aufnehmen kann. Somit muss es für IPV6 folgende neue Struktur geben:



TCP/IP 1.5.G IPV6



- Die Struktur der IP Adressen wurde für IPV6 erweitert:

SOCKADDR_IN6 Format für IPv6

U6SOCKIN	DSECT		
U6LENGTH	DS	XL1	Length of this structure
U6FAMILY	DS	XL1	Protocol family
U6PORT	DS	XL2	UDP or TCP port number
U6FLOW	DS	XL4	IPv6 flow info(not used)
U6IPV4AD	EQU	U6FLOW	IPv4 address(32-bits)
U6IPV6AD	DS	XL16	IPv6 address(128-bits)
U6SCOPE	DS	XL4	Scoped interfaces(not used)
U6L	EQU	*-U6LENGTH	Length of this area

Die Struktur SOCKADDR_IN6 wurde so designed um mit der Struktur SOCKADDR_IN sowohl zu 4.3BSD aber auch zu 4.4BSD Definitionen kompatibel zu sein.



TCP/IP 1.5.G IPV6



IP V4 Adressen werden automatisch in IPV6 Adressen gemapped. Dazu wird folgende Methode verwendet:

0:0:0:0:FFFF:<IPV4-Adresse> → ::FFFF:c0:a8:64:1

Adress-Konvertierung:

IPv4 Name-to-address Übersetzung erfolgt mit den Funktionen:GETHOSTBYNAME() und GETHOSTBYADDR() und bleiben unverändert. Es gibt zwei neue Funktionen die sowohl für IPV6 als auch für IPV4 verwendet werden können:

Diese Funktionen werden nur für 4 Byte IPV4 Adressen verwendet:

INET_ADDR() übersetzt eine IPv4 Adresse von Text nach Binär.

INET_NTOA() übersetzt eine IPv4 Adresse von Binär nach Text.

2 neue Funktionen die IPV4 als auch IPV6 unterstützen wurden neu geschaffen:

INET_PTON() übersetzt eine IPv4 oder IPv6 Adresse von Text nach Binär.

INET_NTOP() übersetzt eine IPv4 oder IPv6 Adresse von Binär nach Text.



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- **BSI TCPIP IPV4**
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten



- **System Requirements Operating System**
 - VSE/ESA 1.3, 1.4, 2.1 oder aktueller (z/VSE 3, ZVSE 4.x)
- **System GETVIS**
 - TCP/IP-TOOLS Anwendungen brauchen **keinen VSE System GETVIS**. Jeder Daten Transfers und Zugriff zur TCP/IP Stack (BSTTINET) Partition wird über Access Registers realisiert.
 - Weder 24-Bit oder 31-Bit System GETVIS wird von irgendeiner TCP/IP-TOOLS Anwendung benötigt.
- **SVA Phases**
 - Keine TCP/IP-TOOLS Phasen sollten in die SVA geladen werden.
- **Partition Execution**
 - TCP/IP-TOOLS können in jeder Partition laufen, einschließlich der Dynamischen Partitionen.



- **BSI TCP/IP** besteht aus dem Stack BSTTINET und den Tools Partitionen.
- Folgende Adapter werden von BSTTINET unterstützt:
 - 3172 LAN Channel Station (LCS) Devices
 - OSA and OSA/2 Devices
 - OSA Express (OSAX) Devices (requires VSE/ESA 2.6 or higher)
 - CLAW devices (RS/6000 and 3172 CLAW mode)
 - CTCA and Virtual CTCA Devices
 - IUCV linkage to VM TCP/IP and LINUX/390
- Die BSTTINET TCP/IP Stack Partition braucht als Minimum eine **16MB Partition**. Das reicht für wenige Benutzer und Test ist aber für die Produktion zu klein.
- **Nach der Formel:** 16MB + 84K pro TCP Verbindung kann man die Größe berechnen: Hat man 100 Telnet und 10 FTP Verbindungen so ergibt sich eine Größe von $16\text{Mb} + 110 \cdot 84\text{Kb} = \text{ca. } 25 \text{ MB}$. Maximal werden 4095 Verbindungen unterstützt was einer Größe von 352 MB entspricht.



BSI TCPIP IPV4



- **BSI TCP/IP** Tools Partitionen sind im einzelnen:
 - Externe FTP Partition
 - GZIP und BSTTGZIP zum Packen und entpacken. Kompatibel zu jedem ZIP Programm
 - External Simple Network Time Protocol (SNTP) Client
 - External Simple Network Time Protocol (SNTP) Server
 - External TN3270E Server
 - Unterstützung von Telnet 3270E
 - Auch Drucker werden unterstützt (Kein GPS erforderlich)
 - Drucker können auch über LPR, LPD, IPP und Direct angesprochen werden.
 - Es gibt auch eine USSTAB = Menu um bestimmte Applikationen auszuwählen.



BSI TCPIP IPV4 Programmierung



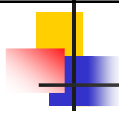
- **Kompatibilität:**
 - Der BSI TCP/IP Stack wurde so designed, dass eine möglichst große Kompatibilität zum IBM/CSI TCP/IP Stack erreicht wird. Anwendungen die mit dem IBM/CSI API erstellt wurden sollten mit dem BSI TCP/IP Stack ohne Änderung laufen, sogar ohne neu kompiliert zu werden. Programme die das IBM/CSI API und das BSI TCP/IP Stack benutzen müssen in der LIBDEF Anweisung die TCP/IPTOOLS Library vor den IBM/CSI Libraries definieren.
- **Programmierung:**
 - Die beste Methode BSI TCP/IP zu programmieren ist es das EZASOCKET Call API zu verwenden.
 - Für COBOL Programme, die das CSI-EXEC Interface verwenden gibt es eine REXX Prozedur, die den COBOL Code auf EZASOCKET CALL umsetzt.
- Wird das LE/C API verwendet so muss die IBM Phase \$EDCTCPV.PHASE gegen die BSI Phase \$EDCTCPV.PHASE ausgetauscht werden. Dieses Interface wird verwendet im CICS/TS, MQSeries, VSE Connector Server und DB2.



Übersicht



- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- **BSI TCPIP IPV6**
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten



BSI TCPIP IPV6



- **VSE/ESA and z/VSE Requirements**
- *Der BSTT6NET IPv6 TCP/IP Stack setzt z/VSE 4.2 (oder grösser) voraus. Nur OSA Express Adapters im QDIO Mode oder Hipersockets werden unterstützt. Der IBM z/VSE IJBOSA Driver muss den APAR Level DY46973 oder grösser haben.*
- *BSI's IPv6/VSE beinhaltet den TCP/IP Communication Stack. BSTT6NET TCP/IP Stack läuft in einer statischen oder dynamischen Partition. Keine anderen TCP/IP Anwendungen (FTP, TN3270E, etc.) laufen in dieser Partition.*
- *BSTT6NET TCP/IP Stack läuft im 31-Bit Mode und nutzt intensiv Access Registers, optimiert für dynamische Partitionen. Beste Performance erhält man, wenn alle anderen Anwendungen auch in dynamischen Partitionen laufen.*
- *Alle anderen Voraussetzungen gelten wie bei IPV4.*



BSI TCPIP IPV6



- Die BSTIT6NET TCP/IP Stack Partition braucht als Minimum eine **20 MB Partition**. Das reicht für wenige Benutzer und Test, ist aber für die Produktion zu klein.
- **Nach der Formel:** 20MB + 124K pro TCP Verbindung kann man die Größe der Partition berechnen: Hat man 100 Telnet und 10 FTP Verbindungen so ergibt sich eine Größe von 20 MB + 110*124Kb = ca. 34 MB. Maximal werden 4095 Verbindungen unterstützt was einer Größe von 516 MB entspricht.




BSI TCPIP IPV6



- **Link Local IPv6 Adresse**
 - Bei IPv6 hat jedes Netzwerk Interface 2 IP Adressen. Die Link Local IPv6 Adresse und eine zugewiesene IPv6 Adresse.
 - Die Link Local IPv6 Adresse kann nur auf dem lokalen Link verwendet werden.
 - Heißt eine zugewiesene IPv6 Adresse 806::1:2, so wird die Link Local IPv6 Adresse dynamisch erstellt. Diese wird aus der MAC Adresse des Interfaces gebildet. Falls die MAC Adresse 0200 0000 0008 ist wird die Link Local IPv6 Adresse fe80::0200:0000:0100:0008 gebildet. Die 0100 wird mitten in der IPV6 Adresse eingefügt und enthält die Interface Nummer (01) + 00.
 - BSI verwendet die gleiche Logik für den Link Local IPv6 Adress- Generierung wie z/VM TCP/IP.




Übersicht





- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- **SSL mit TCP/IP ?**
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- Power PCOPY mit Segmenten

41 4th European GSE / IBM TU for z/VSE, z/VM and Linux on System z 26.10.2010



SSL mit TCP/IP



- **Fragen an alle:**
- Wer setzt SSL ein ?
- Für welche Dienste ? (Telnet, FTP, EMAIL)
- Verschlüsselung nur auf PC Ebene ?
- Wer nutzt den LDAP Sign On

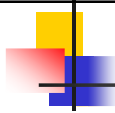
42 4th European GSE / IBM TU for z/VSE, z/VM and Linux on System z 26.10.2010



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- **Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit**
- Power PCOPY mit Segmenten



Zeitumstellung im z/VSE mit IPL (unter z/VM)



HW-Clock ist auf Greenwich Mean Time eingestellt!

Im z/VM sind Zeitzonen vordefiniert:

```
Timezone_Definition EDT East 01.00.00 (Sommerzeit)
Timezone_Definition EST East 02.00.00 (Winterzeit)

Timezone_boundary on 2010-03-28 at 02:00:00 to EDT
Timezone_boundary on 2010-10-31 at 02:00:00 to EST
```

Diese Definitionen stehen in der System Config, auf cf1,cf2,cf3 beim User MAINT.

Online-Änderung im z/VM per: `SET TIMEZONE est`

IPL z/VSE:

Kein SET ZONE- Befehl beim IPL → es wird die virtuelle Zeit vom z/VM im z/VSE übernommen

SET ZONE- Befehl beim IPL → es wird bei IPL der Offset zur HW-Clock errechnet



Zeitumstellung: Sommerzeit auf Winterzeit



Probleme für Subsysteme, wie CICS, DLI, DB2, Sceduler, TMON, und andere langlaufende Vendorprodukte oder eigene Langläufer . . . !

Doppelte Zeiteinträge bei Recovery, Journaling, Logging, Accounting müssen vermieden werden

Herunterfahren der kritischen Subsysteme (Kundenverantwortung)!

Ansonsten kann es zu unvorhersehbaren Problemen kommen !

Zwingend notwendige Vorgehensweise:

Alle problematischen Subsysteme wie CICS, DB2, ... herunterfahren
Zeitumstellung aktivieren mit TIME-Befehl

Mindestens 1 Stunde warten (sollte auch bei IPL gewartet werden)
Restart der Subsysteme

IPL nicht zwingend erforderlich, aber empfehlenswert



Zeitumstellung: Sommerzeit auf Winterzeit



Umstellung ohne IPL ? Unter z/VM - ja, liegt in Kundenverantwortung!

Auch wenn DB2 im z/VSE aktiv ist, kann z/VSE auf Winterzeit umgestellt werden - ohne ein IPL auszuführen.

Wird im CICS Logging und Recovery verwendet, sollten die CICS Systeme beendet werden.

Folgende Schritte sind durchzuführen:

Im z/VM müssen Zeitzonen in der SYSTEM CONFIG wie folgt definiert sein:

```
Timezone_boundary on 2010-03-28 at 02:00:00 to EDT  
Timezone_boundary on 2010-10-31 at 02:00:00 to EST
```

Unter z/VM wird die Zeitzone umgestellt mit dem Befehl:

```
set timezone est
```



Umstellung auf Winterzeit



Die Abfrage, welche Zeitzone aktiv ist geht mit:

```
query timezone
Zone Direction Offset Status
UTC ---- 00.00.00 Inactive
GMT ---- 00.00.00 Inactive
EDT East 01.00.00 Inactive
EST East 02.00.00 Active
```

Im z/VSE auf der Systemconsole den Befehl -

```
time zone=vm
```

 eingeben.

Für **alle** aktiven CICS/TS Systeme muss der folgende CEMT Befehl gestartet werden:

```
CEMT PERFORM RESET
```

Dann DB2 Systeme und CICS/TS mit Logging/ Recovery eventuell neu starten.



Übersicht



- Kurz notiert
- TCP/IP 1.5.F Programming Styles
- TCP/IP 1.5.G IPV6
- BSI TCPIP IPV4
- BSI TCPIP IPV6
- SSL mit TCP/IP ?
 - LDAP und Sign-On Security ?
- Zeitumstellung: Am 31.10.2010 beginnt die Winterzeit
- **Power PCOPY mit Segmenten**



PCOPY - Kopieren von Power Queue Einträgen:

d lst,ams*,full=yes

```
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R46I LIST QUEUE P D C S PAGES CC FORM BU
F1 0001 1R46I AMSPRINT 25871 3 D Y 5000 1 S=015 FROM=(MGOY)
F1 0001 D=14/10/2010 S=015 DBGP=000144 L=00190000
F1 0001 QNUM=01007 T=07:54:39
```

pcopy lst,amsprint,25871,cqnum=1007,class=x

```
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1QK4I PCOPY ATTEMPT REJECTED, RC=0014
```

PF9 auf 1QK4I gibt die Erklärung für PC=0014



Explain 1(3)

```
1QK4I PCOPY ATTEMPT REJECTED, RC=nnnn
```

Explanation: The PCOPY command had to be rejected. The reason code (RC=nnnn) can be one of the following:

```
RC=0001 There is no VSE/POWER partition Getvis storage available
          for a new queue record
RC=0002 The queue record specified by CQNUM= is a 'FREE' queue
          record
RC=0003 The queue record specified by CQNUM= is a 'BAD' queue
          record
RC=0004 The queue record specified by CQNUM= is an 'excluded'
          queue record
RC=0005 The queue record specified by CQNUM= is a 'reader' type
          queue record
RC=0006 The queue record specified by CQNUM= is not in the 'XMT'
          queue as selected by the PCOPY command
RC=0007 The queue record specified by CQNUM= is not in the 'DEL'
          queue as selected by the PCOPY command
RC=0008 The queue record specified by CQNUM= is not in the 'LST'
          queue as selected by the PCOPY command
RC=0009 The queue record specified by CQNUM= is not in the 'PUN'
          queue as selected by the PCOPY command
```



Explain 2(3)

```
1QK4I      PCOPY ATTEMPT REJECTED, RC=nnnn

RC=0010    The queue record specified by CQNUM= has a different
           jobname than selected by the PCOPY command
RC=0011    The queue record specified by CQNUM= has a different
           jobnumber than selected by the PCOPY command
RC=0012    The queue record specified by CQNUM= is a master queue
           record with the maximum of 99 duplicates
RC=0013    The queue record specified by CQNUM= is a duplicate queue
           record and its master has already the maximum of 99
           duplicates
RC=0014    The queue record specified by CQNUM= is an output segment
           with suffix number
RC=0015    There is no more VSE/POWER queue record (-number)
           available for the queue record of the new copy
RC=0016    The queue record specified by CQNUM= is a duplicate queue
           record, but the pointer to its master is outside the queue
           file, i.e. is less than zero or is greater than MRQ#MAX -
           as documented by the preceding 1Q2JI Idump
RC=0017    The queue record specified by CQNUM= is a duplicate queue
           record, but its master has a 'count-of-duplicates' equal
           to zero (which should at least be one or more) - as
           documented by the preceding 1Q2JI Idump
```



Explain 3(3)

```
1QK4I      PCOPY ATTEMPT REJECTED, RC=nnnn

RC=0018    The queue record specified by CQNUM= is a duplicate queue
           record, but its master does not have the same spooling
           values (QRNB, QRDF, or QRLDF) - as documented by the
           preceding 1Q2JI Idump
RC=0019    The queue record specified by CQNUM= has an appendable
           (DISP=A) disposition
RC=0020    The operator or the programmed interface is not authorized
           to access the queue entry (specified by CQNUM=) for a
           PCOPY operation (similar to a PALTER modification)

System Action: The command is ignored.

System Programmer Response: None.

Operator Response: Reissue the corrected command or notify your
system programmer, when system shortages have occurred.
```

Haben Sie noch Fragen?



**Themenvorschläge für die
nächste Tagung ?**

26.10.2010

**Vielen Dank
für Ihre
Aufmerksamkeit**

26.10.2010