

SD15 - z/VSE Workshop: Focal Point TCP/IP

Dagmar Kruse (dkruse@de.ibm.com)

IBM Deutschland GmbH

Heinz Peter Maassen (hp.maassen@lattwein.de)

Lattwein GmbH

- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge) → nächste Frühjahrsession
- SOAP und Webservices → nächste Frühjahrsession

Wer hat sich mit TCP/IP for VSE 1.5E beschäftigt?

- Bisher nur testweise im Einsatz?
 - **Erfahrungen: positive/negative**

- Schon produktiv im Einsatz?
 - **Seit wann?**
 - **Erfahrungen: positive/negative**

- Hatten Sie Schwierigkeiten bei der Implementierung?

- Was ist im TCP/IP 15 E besser/schlechter?

- Welche (neueren) Funktionen nutzen Sie?

Seit **Okt 2006** von CSI verfügbar

- ZAPS unter

<http://www.csi-international.com/csi-support/zaps15e.htm>

Inhalt:

- TCP/IP-Stack wurde überarbeitet
- Performanceverbesserungen, speziell im FTP-Bereich
- Security-Erweiterungen
- ...

TCP/IP for VSE/ESA 1.5. Service Pack E



Seit Nov. 2006 von IBM verfügbar:

Verfügbar	APAR	PTFs	Beschreibung
August 2007	PK51399	8TP - UK28397 9TP – UK28398	Sammel-PTF
Juni 2007	PK47545	8TP - UK26361 9TP - UK26374	SIT OPTION RENTPGM=Protect and SSL
März 2007	PK38492	7IP - UK22215 8TP - UK22216 9TP - UK22217	Sammel-PTF
März 2007	PK38365	7IP: UK21991 8TP UK21992 9TP UK21993	Online Mesaage Explanation-File Stand: TCP/IP 150 Service Pak E
Nov. 2006	PQ87041 (+ II14121, II14130)	7IP - UK19329 8TP - UK19330	TCP/IP 150, Service Pack E + ZP15E201 & ZP15E202 (03/2007 z/VSE 4.1 GA-Stand)

ZIP: VSE/ESA 2.7, 8TP:z/VSE3.1, 9TP: z/VSE4.1

Aktuelle CSI-Literatur:

- <http://www.csi-international.com/download.htm>

Überarbeitete Bücher für TCP/IP for VSE 1.5.E: **(Stand 28.9.2007)**

- Command Reference 15E (Beta) (06/2007)
- Installation Guide (08/2007)
- User Guide (09/2007)
- Messages and Codes (02/2007)
- Optional Features (11/2006)
- Programmers Guide (10/2006)
- Email and PDF Delivery Methods (Beta) (11/2006)

- Online Messages (01/2007)
- Online Commands (Beta)(2007)

Zusätzliche IBM-TCP/IP-Literatur:

- <http://www-03.ibm.com/servers/eserver/zseries/zvse/documentation/>
- TCP/IP for VSE/ESA – IBM Program Setup and Supplementary Information, SC33-6601-08, (03/2007)
 - Kapitel 4: Security-Exit BSSTISX
 - Kapitel 10, 11: EZASOKET und EZASMI laufen auch außerhalb des LE Environments
- IBM z/VSE Planning V4.1 (03/2007)
 - Chapter 11. TCP/IP, OSA, and HiperSockets Support

Mindestanpassungen aufgrund der Security-Erweiterungen:

- USER ID und Password wird für alle Prozesse benötigt (explizit /Default)
→ Bei Installation i.d.R. folgende **Definitionen ergänzen** (IPINIT):

```
DEFINE USER,ID=$WEB,PASSWORD=$WEB,WEB=YES
DEFINE USER,ID=$LPR,PASSWORD=$LPR,LPR=YES
DEFINE USER,ID=$EVENT,PASSWORD=$EVENT,LPR=YES
DEFINE USER,ID=$LPD,PASSWORD=$LPD,LPD=YES
DEFINE USER,ID=$EMAIL,PASS=$EMAIL
```

- Bei **Nutzung des IBM Security Exits BSSTISX**:

In der 'Exception Liste' BSSTIXE ergänzen für analoges Verhalten zum TCP/IP for VSE 15D:

```
DC  XL2'1400'   - SXTYLOGI  allow FTPD LOGIN message
DC  XL2'1A00'   - SXTYCWDL  allow FTP subdirectories
DC  XL2'1D00'   - SXTYCWDL  allow FTP commands
```

Neue FTPDAEM-Phase (reentrant) statt bisherige FTPD-Phase

- Mehrere FTP-Sessions werden pro FTP-Daemon unterstützt.
 - Eine Task wird zum Überwachen des PORTs gestartet
 - Für jede Session wird neue Tasks dynamisch gestartet.
- Bisher nur 1 FTP-Session pro FTP-Daemon möglich

Performanceverbesserung laut CSI:

- 10-20% schneller
- Weniger CPU und Storage-Verbrauch !

DEFine FTPD ,ID=id

```
[,ALLowabort={YES | NO}] [,BUFFERSize=num]
[,COUNT=num] [,DATAPort=num][,DRIVER=string]
[,DYNfiles={YES | NO}] [,EXTtypes={YES | NO}]
[,HESitate=num] [,IDLEtimeout=id] [,IPAddr=x.x.x.x]
[,JOURnal=YES | NO} [,LIPAddr=x.x.x.x] [,MAXACTive=num]
[,PORT=num] [,SENDFast={YES | NO}]
[,SENDWack={YES | NO}][,SSL={YES | NO | YESCLAUTH}]
[,SSLCIPHER=string]
[,SSLDATAconn=string] [,SSLKEY=string]
[,SSLVERsion={0300 | 0301 | SSLV3 | TLSV1}]
[,TIMEOut=num] [,TRANslate=string] [,UNIX={YES| NO}]
[,UPPERcase={YES | NO}]
[, WELcome=string] [,ZEROerror={YES | NO}]
```

TCP/IP for VSE 1.5.D: FTP- Daemon



IPINIT00.L

DEFINE FTPD, ID=FTP, PORT=21, COUNT=2

Im TCP/IP 15D:

88 query ftpds

R1 0086 IPN435I	ID: FTP01	Port: 21	Driver: FTPD	SSL: NO
R1 0086 IPN874I	Buffers: 3 Buffer size: 131072 bytes			
R1 0086 IPN350I	Current Status: Inactive			
R1 0086 IPN351I	Current IPaddr: 0.0.0.0			
R1 0086 IPN355I	Current Userid:			
R1 0086 IPN435I	ID: FTP02	Port: 21	Driver: FTPD	SSL: NO
R1 0086 IPN874I	Buffers: 3 Buffer size: 131072 bytes			
R1 0086 IPN350I	Current Status: Inactive			
R1 0086 IPN351I	Current IPaddr: 0.0.0.0			
R1 0086 IPN355I	Current Userid:			

IPINIT00.L

DEFINE FTPD,ID=FTP,PORT=21,COUNT=2

Im TCP/IP 15E:

88 query ftpds

R1 0085 IPN253I << TCP/IP FTP Daemons >>

R1 0085 IPN435I ID: FTP Port: 21 Driver: FTPDAEMN SSL: No

R1 0085 IPN874I Buffers: 2 Buffer size: 65536 bytes

R1 0085 IPN350I Current Status: Listen

R1 0085 IPN362I Maximum sessions: 3, Current: 0

Option COUNT= num (1 through 99)

- Aus Kompatibilitätsgründen noch unterstützt
- Synonym für Option "MAXACTIVE=" (falls COUNT > 3)
- Bisher gab COUNT die Anzahl der zu definierenden FTP Daemons an
- Jetzt gibt COUNT die Anzahl der zu unterstützenden FTP Sessions dieses FTP-Daemons an

Option MAXACTIVE= num (Default=3)

- Anzahl der gleichzeitig unterstützten FTP-Sessions durch diesen FTP-Daemon
- Vereinfacht FTP-Konfiguration und reduziert Overhead
- Achtung:
Jeder gleichzeitig stattfindende
"AutoFTP"-Event benötigt eine FTP-Session.

→ Überdenken Sie noch mal Ihre DEFINE FTPD-Definitionen!

Option HESITATE = Numeric, 0 through 99999.(Default=0)

0 bedeutet:

- Keine Verzögerung beim Versenden von Daten
- Kein Warten auf ein Acknowledge von der empfangenden Seite

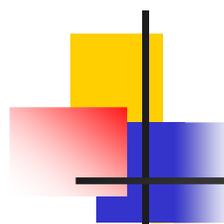
n = n/300th Sekunde

- Gibt das max. Zeitintervall an, wie lange auf ein Acknowledge der empfangenden Seite gewartet wird, bevor der nächste Transfer-Puffer gebildet und gesendet wird.
- Kann zu langsamen Datenübertragungen führen
- **"WAIT-for-ACK" Performance-Problem**

NEU: Jetzt für einzelnen FTP-Daemon einstellbar !!!

- Sind Ihnen ähnliche Änderungen im TCP/IP for VSE 1.5.E aufgefallen?
(s. DEFINE FTPD option COUNT)
- Hatten Sie Schwierigkeiten bei der Implementierung?
- Was ist im TCP/IP 15 E besser/schlechter?
- Welche (neueren) Funktionen nutzen Sie?

- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge)
- SOAP und Webservices



CHECKSUM Hardware

CHECKSum {SOftware|HARdware|OFF}

- SOftware – The checksums of incoming datagrams are computed and validated using an **internal software algorithm**. This is the **default**.
- Off – **Sollte im Produktionssystem NIE benutzt werden !**
- HARdware – The checksums of incoming and outgoing datagrams are computed and validated using the **hardware CHECKSUM instruction**.

Speziell bei der z990, z890 oder z9 BC/EC:

**Der interne Service Prozessor verarbeitet CHECKSUM Instruktionen,
NICHT der Standard Prozessor der Betriebssysteme**

→ Nutzen Sie **CHECKSUM HA !!**

Werden z.T. noch akzeptiert, aber nicht ausgeführt:

Dispatching Algorithmus im TCP/IP for VSE 15D geändert:

- **SET DISPATCH_TIME=nnn**
- **SET REDISPATCH=nnn**

und

- **SET TRANSfer_buffers=num**

Größe des 'Buffer-Pools' der FTP Daemon

(Im TCP/IP 15D wird 0 angezeigt bei Query sets)

Einzelne Zaps/Fixes auf CSI-Homepage verfügbar

- Auch Einspielen, wenn TCP/IP for VSE über IBM bezogen wird
- Bei PMR aber dann genauen Service-Stand angeben
 - Alle Fixes werden bei ‚Query version‘ aufgelistet

Bessere Fehlersuche mit dem **IP-Trace-Tool**

- Unter Windows mit JRE 1.4.0 oder höher
- Konvertierung des CSI TCP/IP for VSE Trace-Output in CAP-Format
- Lesbar mit **Wireshark** (Vorgänger **Ethereal**) oder **Packetyzer**.
- **Kostenfrei** verfügbar auf VSE-Homepage:

<http://www-03.ibm.com/servers/eserver/zseries/zvse/downloads/tools.html#iptracetool>



TRACEVSE.TEXT.cap - Ethereal

File Edit Capture Display Tools Help

No.	Time	Source	Destination	Protocol	Info
48	26.042382	192.168.1.34	192.168.1.32	TCP	20 > 1028 [SYN] Seq=539
49	26.052086	192.168.1.32	192.168.1.34	TCP	1028 > 20 [SYN, ACK] Seq=625381125
50	26.053439	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
51	26.053443	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
52	26.079721	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
53	26.079722	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
54	26.079945	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
55	26.079948	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
56	26.093056	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
57	26.093058	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
58	26.093268	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
59	26.093272	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539
60	26.105987	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
61	26.105989	192.168.1.32	192.168.1.34	FTP-DATA	FTP Data: 4056 bytes
62	26.106197	192.168.1.34	192.168.1.32	TCP	20 > 1028 [ACK] Seq=539

Frame 52 (4112 on wire, 4112 captured)
 Raw packet data
 Internet Protocol, Src Addr: 192.168.1.32 (192.168.1.32), Dst Addr: 192.168.1.34 (192.168.1.34)
 Transmission Control Protocol, Src Port: 1028 (1028), Dst Port: 20 (20), seq: 625381125, len: 4056
 FTP Data

0000	45 00 10 00 65 2e 00 00	3c 06 86 37 c0 a8 01 20	E...e... <..7...
0010	c0 a8 01 22 04 04 00 14	25 46 8f 05 00 51 ba 3e	...". %F...Q.>
0020	50 10 20 00 cc 31 00 00	61 61 40 d1 d6 c2 40 e4	P. ..1.. aa@...@.
0030	c4 f5 f2 f4 f0 f3 40 40	40 40 40 40 40 40 40 40@@ @@@@@@@@
0040	40 40 40 40 40 40 40 40	40 40 40 40 40 40 40 40	@@@@@@@@ @@@@@@@@

Filter: File: TRACEVSE.TEXT.cap

- Welche allgemeinen Tipps können Sie aus Ihrer Erfahrung geben?
 - Hier und Heute ?
 - Beim nächsten Mal ?

- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge)
- SOAP und Webservices

- TCP/IP ...
- Macht z/VSE zum Member der Open Community
- Moderne Anwendungsentwicklung geht über alle Plattformen
- Weiterverwenden von vorhandenem Wissen
- Neue Technologien integrieren in bestehende Anwendungen.

- Früher war SNA die Sprache um Host-Anwendungen mit anderen Plattformen zu verbinden.
- Mit APPC Programming API konnten Anwendungen erstellt werden die sich mit LU6.1 oder LU6.2 an ein CPI-C Programm auf PC Basis unterhalten haben.
- Aber: SNA Verbindungen waren aufwendig in der Konfiguration und ziemlich sensitiv in der Anpassung an die vorhandene Hardware.

■ LU6.2 API

```
EXEC CICS ALLOCATE SYSID(CONNID)
EXEC CICS CONNECT PROCESS CONVID(CONNAM) SYNCLEVEL(0) *
          PROCNAME(PS2TRA) PROCLENGTH(PS2TRL)
EXEC CICS WAIT CONVID(CONNAM)
EXEC CICS SEND CONVID(CONNAM) FROM(OUTBER) LENGTH(OUTLEN) WAIT
EXEC CICS FREE CONVID(CONNAM)
```

■ TCP/IP EXEC Level Interface

```
EXEC TCP OPEN FOREIGNPORT(COMPORT) FOREIGNIP(COMIPADR) LOCALPORT(0)*
          RESULTAREA(RSLTAREA) DESCRIPTOR(CLNTOKEN) ACTIVE WAIT(YES)*
          ERROR(ERRORO)
EXEC TCP SEND FROM(OUTBUF) LENGTH(OUTLENF) RESULTAREA(RSLTAREA) *
          DESCRIPTOR(CLNTOKEN) WAIT(YES) ERROR(ERRORS)
EXEC TCP CLOSE RESULTAREA(RSLTAREA) DESCRIPTOR(CLNTOKEN) *
          WAIT(YES) ERROR(ERRORCL)
```

■ EZASMI API

```
EZASMI TYPE=INITAPI,MAXSOC=MAXSOC,MAXSNO=MAXSNO, X  
ERRNO=ERRNO,ASYNC='NO',RETCODE=RETCODE, X  
SUBTASK=NSTASK
```

```
EZASMI TYPE=SOCKET,AF='INET',SOCTYPE='STREAM',PROTO=PROTOCOL, X  
ERRNO=ERRNO,RETCODE=RETCODE
```

```
EZASMI TYPE=CONNECT,S=SOCKET1+2,NAME=SAMPSEV,ERRNO=ERRNO, X  
RETCODE=RETCODE
```

```
EZASMI TYPE=WRITE,S=SOCKET1+2,NBYTE=MSG1L,BUF=MSG1, X  
ERRNO=ERRNO,RETCODE=RETCODE
```

```
EZASMI TYPE=CLOSE,S=SOCKET1+2,ERRNO=ERRNO,RETCODE=RETCODE
```

```
EZASMI TYPE=TERMAPI , * TERMINATE API
```

- LU6.2 Programmierung und TCP/IP EXEC Level Interface sind ähnlich in der Verwendung.
- Unterschiede gibt es allerdings reichlich:
 - Bei einem RECEIVE werden bei LU6.2 alle Daten gelesen die bei LENGTH angegeben wurden.
 - Bei TCP/IP gibt LENGTH lediglich die maximale Länge des Buffers an in den die Daten gelesen werden.
 - Die tatsächlich übertragene Anzahl wird in einem internen Feld SRCOUNT bereitgestellt.
 - Man muss selber das Ende einer Übertragung erkennen (Länge mitteilen oder CRLF am Schluss).

- Gravierender Unterschied zwischen APPC LU6.2 und TCP/IP liegt im Aufruf der Server Task.
 - Bei LU6.2 wird der Name dieser Task beim CONNECT gesetzt und automatisch gestartet. Hierzu muss nichts im voraus gestartet werden. Lediglich die SYSID muß im Status INSERVICE ACQuired stehen.
 - Bei TCP/IP muss auf dem Server eine Task gestartet werden, die auf einen vorher ausgehandelten Port lauscht und wartet. Der Befehl dazu lautet LISTEN. Wird von der Client Seite der Connect gegeben, so erhält der LISTEN Befehl die Kontrolle und das Programm läuft weiter.

- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge)
- SOAP und Webservices

- Bei TCP/IP gibt es mehrere Programmier Schnittstellen.
 - Das Socket Macro Interface nur für Assembler aber nicht portabel
 - Das Preprocessor (EXEC Level) Interface für Assembler / Cobol / PLI - Allerdings auch nur für z/VSE verfügbar. Einfach im Handling.
 - Das BSC-C Interface – nur für C/390
 - Das LE/C Interface für C/390. Weitgehend z/OS kompatibel
 - EZASMI Interface – für Assembler basierend auf LE/C Interface und weitgehend z/OS kompatibel.
 - EZASOCKET CALL Interface für Assembler/ Cobol / PLI. Wie EZASMI weitgehend z/OS kompatibel.

- Welche API's für die Socket Programmierung gibt es?

API	Sprachen	Von	Bemerkung	Dok.
SOCKET macro	Assembler	CSI	-Sehr effizient -Kein Standard -Ausführungsumgebung muss beim Assembly angegeben werden	a)
Pre-Processor Interface (EXEC TCP FTP )	COBOL PL/I Assembler	CSI	-Kein Standard -Ausführungsumgebung muss beim Compile angegeben werden	a)
BSD-C Interface	C	CSI		a)
LE/C Interface	C	IBM	-Weitgehend zOS kompatibel -Kein re-compile notwendig bei TCP/IP Service	b)
EZASMI macro Interface	Assembler	IBM	-weitgehend zOS kompatibel -kein re-compile der Applikation notwendig -basiert auf dem LE/C interface	b)

... weiter auf der nächsten Seite

API	Sprachen	Von	Bemerkung	Dokum
EZASOKET call	Assembler COBOL PL/I	IBM	-weitgehend zOS kompatibel -kein re-compile der Applikation notwendig -basiert auf dem LE/C interface	b)
REXX SOCKET	REXX	CSI IBM		a) c)

■ Dokumentation der API's

- a) TCP/IP for VSE: Programmer's Guide
- b) TCP/IP for VSE/ESA – IBM Program Setup and Supplementary Information (SC33-6601-07)
- c) REXX/VSE Reference (SC33-6642)

➤ Eine vergleichende Übersicht der API's befindet sich in b).

- Für EZASMI, EZASOCKET CALL, aber auch für das EXEC TCP API gilt:
 - Daten werden als Datenstrom ohne Trennzeichen verarbeitet.
 - Wenn Anwendung A und B verbunden sind und Anwendung A sendet 1000 Bytes an Anwendung B, so kann jeder SEND(*) Aufruf 1, 10 oder die ganzen 1000 Bytes übertragen.
 - Die Anzahl der übertragenen Bytes wird im Feld RETCODE bei EZASMI oder SRCOUNT bei EXEC TCP API bereitgestellt.
 - Deshalb sollte der READ CALL und der SEND CALL(*) in einer Programm Schleife abgearbeitet werden, bis alle Daten gesendet oder gelesen wurden.
- * Bei SEND werden entweder alle Zeichen oder keins übertragen, da die Aufteilung in die einzelnen Pakete im TCP/IP Stack geschieht.

- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge)
- SOAP und Webservices

- Wichtige Hinweise zum EZASMI Interface.
 - Nur für CICS TS **nicht für CICS/VSE**
 - Nicht in ICCF Pseudo Partitions
 - INITAPI / TERMAPI innerhalb der gleichen TASK !
 - Alle Programme müssen mit EXECKEY(CICS) definiert sein, wenn CICS mit Storage Protection(YES) läuft.
 - AMODE muss 31 sein

- Wichtige Hinweise zum EZASMI Interface.
 - Obwohl es das EZASMI Interface für Assembler Programme gibt, muss das Programm als Unterprogramm von einem LE Programm aufgerufen werden. **(Nicht mehr ab z/VSE 4.1!)**
 - Alle EZASMI Calls laufen asynchron. Im INITAPI kann der Parameter ASYNC=ECB eingetragen werden, dann muss der WAIT selbst programmiert werden.
 - Jeder Call verweist auf einen 160 Byte Storage, von dem das 1 Vollwort der ECB ist.

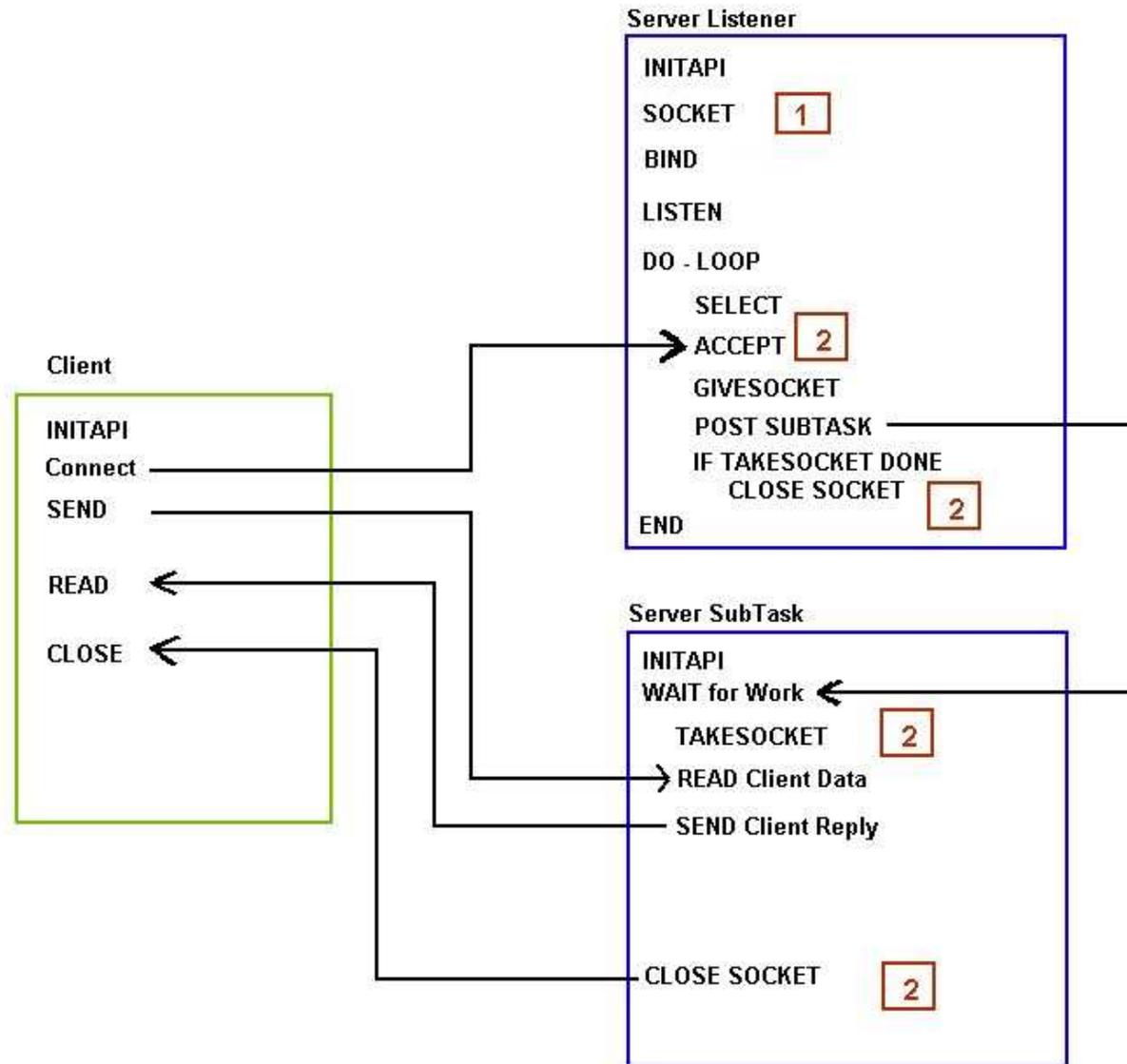
■ Neu mit z/VSE 4.1 !

- Bevor das neue EZA API (EZASMI Makro und EZASOKET Call Interface) im einem CICS TS Umfeld benutzt werden können, muss die "task-related-user-exit" (TRUE) Routine gestartet werden. Sie heißt: EZATTRUE.
- Diese Routine ist verantwortlich Speicher für den Task Arbeitsbereich anzufordern bzw. Bei Task Ende diese Speicherbereiche wieder freizugeben.
- EZATTRUE wird gestartet mit dem Programm EZASTRUE entweder über die Transaktion EZAT START bzw. STOP.
- Alternativ kann in der PLTPI ein Eintrag für EZASTRUE für CICS Start und in der PLTSD für CICS Ende vorgenommen werden.
- Die 3. Möglichkeit ist der Aufruf über EXEC CICS LINK des Programms EZASTRUE mit der COMMAREA Parameter Liste:
- "EZASTRUES " (Start) oder "EZASTRUET " (Ende).
- Siehe hierzu: TCP/IP for VSE/ESA – IBM Program Setup and Supplementary Information

- **Wichtige Tipps:**

- 1. Neu sind die Error Codes 157, 1110, 20111-20114 !
- 2. Wenn das EZATRUE nicht geladen wurde gibt es den Fehlercode 20112
- 3. In der Standard PLTPI / PLTSD ist der Eintrag für EZASTRUE **nicht** Default! Das heißt mit der Umstellung auf z/VSE 4.1 laufen evtl. bestehende Programme erst nach Änderung der PLTPI.

Logic Flow mit EZASMI Interface



- TCP/IP for VSE/ESA 1.5. Service Pack E
- Allgemeine Tipps zum TCP/IP for VSE
- TCP/IP und VTAM SNA
- TCP/IP API Vergleich
- Hinweise zum EZASMI API
- Ein Beispiel (Programmauszüge)
- SOAP und Webservices