

# User Experience: CICS Sockets – A Strategic Solution

Steve Ware, **UF**

Session 1050, SHARE 105  
August 24 (Wed.), 2005, 9:30 AM  
Hynes Convention Center,  
2<sup>nd</sup> Floor, Room 207

# Abstract

The **University of Florida** has considered the **CICS Socket Interface** a strategic part of their enterprise IT solutions suite for many years. Now that even IBM has deemed the CICS Socket Interface a strategic CICS connectivity option (as noted in the June 2004 CICS solutions White paper, G224-7324 “Delivering e-business access to CICS: Strategic Options”), it's time to once again take a closer look at the past, present, and future of this very capable component of Communications Server for z/OS.

<mailto:sfw@cns.ufl.edu>

<http://nersp.cns.ufl.edu/~sftware/share105/s1050sw.pdf>

# Disclaimer

Standard disclaimers apply. Any opinions expressed are the opinion of the author only. Any mentioned brand names, trademarks, registered trademarks, service marks, etc., are the exclusive property of their owners. No warranties are either expressed or implied, your mileage may vary, etc.



# Agenda/Topics

- Introduction
- Past/Present/Future Overview
- Strategic Considerations
- Applications
- Security Considerations
- CICS Regions Consolidation
- Miscellaneous Considerations
- Summary and Q&A
- Appendix and Additional Information

# Introduction



- **UF CNS**, University of Florida Computing & Networking Services (formerly known as **NERDC**), is the primary data center at the Gainesville, FL campus.
- Currently utilizing an IBM **zSeries** z800 2066-002 with 16GB, running **z/OS** 1.4, **CICS** TS 2.3, **DB2** 7.1, **RACF**, **JES2**, etc.
- 3 LPARs - 1 **internal** "sysprog sandbox", 1 **test** "alternate", and 1 **production** or "primary".
- We have **12 CICS regions** configured, and run **~1M** prod. transactions/weekday, and **~2M** on **peak** load days.
- **2 internal/test** sandbox, **2 development/test**, **3 test/QA**, and **5 production** CICS regions currently configured.



The powerful IBM  
zSeries 800

# Introduction

- Founded in **1853**, became the **University of Florida** in **1905**.  
(East Florida Seminary -> Florida Agricultural College -> University of Florida)



- **UF** is a member of the **AAU**, the Association of American Universities.

**AAU**

Association of American Universities

1200 New York Avenue, NW, Suite 550 Washington, DC 20005 202.408.7500

*An association of 62 leading research universities in the United States and Canada*

- UF is one of the five largest universities in the U.S., public or private.
- Over **48K enrolled** at the start of Spring 2005 semester, and over **250K alumni**.



# Introduction

- We're considered a "**Classic**" CICS site.  
("Legacy = It Works!")
- Web access to CICS is via the **CICS Socket Interface**, in use at our site since ~1997.
- More than ~**60%** of local CICS tasks utilize **sockets**.
- All locally developed CICS applications are **Assembler** and/or **COBOL**. We have ~8K CICS application load modules, and ~32 have CICS sockets API.
- Several internal CICS application programs written in **C/C++** and **REXX**. Java has only been IVP tested.

# Introduction

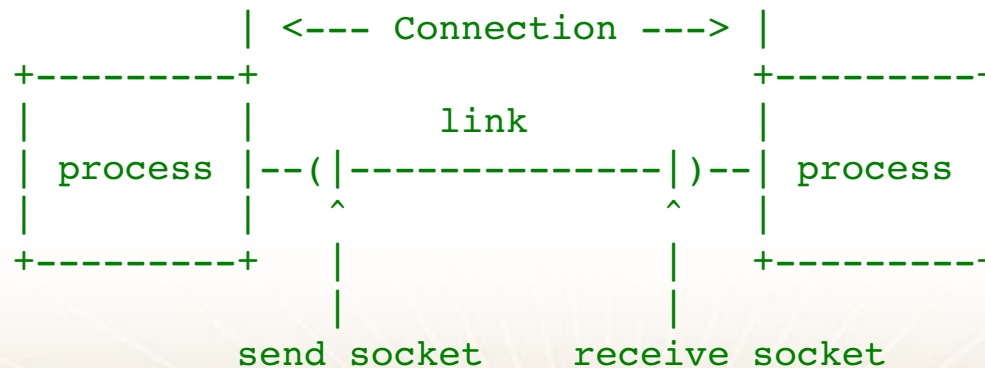
- Separation of **business**, **data** access, and **presentation logic** is very important to easily web-enable CICS applications.
- More recently, IBM (and others) have also recommended that **interface logic**, or **component interfaces**, be separated, to make it easier to integrate CICS applications with other disparate systems.
- CICS has many connectivity options, but the **CICS Socket Interface** allows connectivity to almost **any** system that supports TCP/IP, which is about every known computing system today.

# Introduction

- This presentation is focused on the **CICS Socket Interface** (CSI) component of IBM Communications Server for z/OS (aka TCP/IP for MVS), not the **Sockets Domain** component of CICS TS for z/OS.
- See the presentation **Appendix** for links to prior SHARE presentations, IBM documentation, IBM presentations, etc., that include more overview, implementation, and configuration detail. Links to sample programs and applications are also provided.
- Recall that IBM TCP/IP z/OS development is in **Raleigh, NC**, and IBM CICS development is in **Hursley, UK**.

# Past/Present/Future Overview

- **RFC 33** - New Host-Host Protocol, 12 February 1973.
- "A socket forms one end of a connection, and a connection is fully specified by a pair or sockets."
- RFC 33, Figure 4, The relationship between sockets and processes:



## Past/Present/Future Overview

- **Note:** See the appropriate Appendix section of the IBM CICS Socket Interface Guide and Reference Manual "Related protocol specifications (RFCs)", which lists the related RFCs for TCP/IP.
- "The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the *de facto* standards, on which the TCP/IP protocol suite is built."

## Past/Present/Future Overview

- CICS Socket Interface **optional feature** added to TCP/IP for MVS in 1992, and was additional cost at the time.
- Due to budget considerations, initial requests to purchase this optional/added cost feature were **turned down** by data center management.
- IBM eventually **bundled** the CICS Socket Interface "no additional cost" feature into OS/390 1.x.
- After we installed OS/390 1.3, in **about 1997**, we started taking a look at IBM TCP/IP 3.2 for MVS with "Enhancements Kit", FMID JTCP32C, aka "Release 32C".

## Past/Present/Future Overview

- Initially, the CICS Socket Interface required at least CICS/ESA 3.3 - we had CICS/ESA 4.1. Note that there were some limitations if you only had CICS/ESA 3.3.
- Note that during this time frame Interlink Computer Sciences, Inc. had a competing MVS IP stack with a CICS socket interface and included development tools.
- Then as now, the primary reference manual is the IBM "CICS Sockets Guide (and Reference)" in the "IBM Communications Server" bookshelf. In z/OS 1.4, it is currently titled "*z/OS V1R4.0 CS: IP CICS Sockets Guide*" and has document number *SC31-8807-##*.

# Past/Present/Future Overview

- Socket applications are **peer-to-peer** with both ends programmable. "P2P" has been in the news recently.
- The CSI is **bi-directional** (inbound and outbound socket calls), and a "**low-level**" interface.
- IBM says: "The advantage of using the socket API directly is the opportunity for developing high-performing applications with a very low software overhead."
- The IBM CSI provides a comprehensive **socket API** (Application Programming Interface).

# Past/Present/Future Overview

- CSI is implemented in CICS as a **TRUE** (Task Related User Exit) similarly to DB2.
- Provides reusable MVS **TCBs**, **listener(s)**, a **security exit** point, etc.
- **Strategic? Yes!**

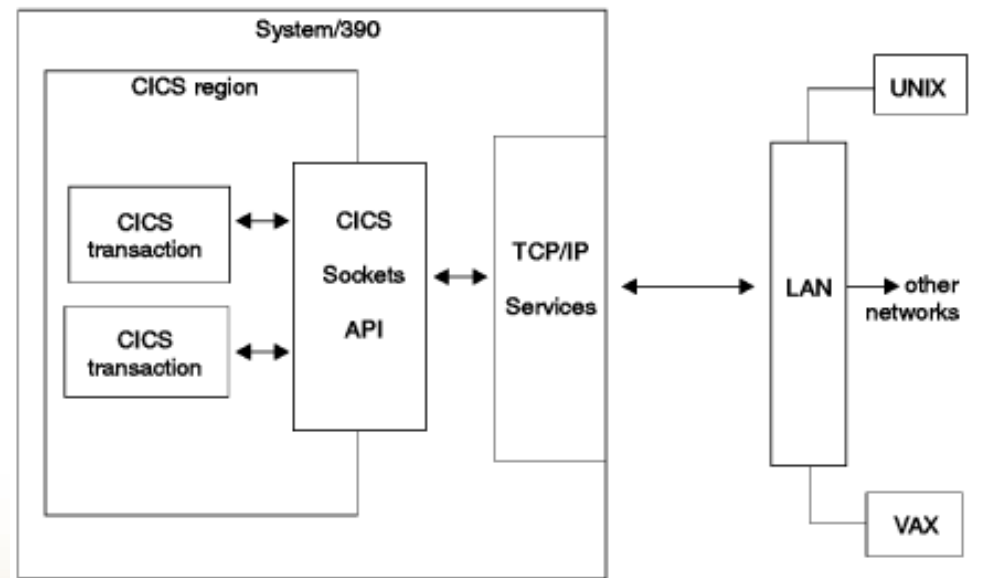


Figure 1. The use of CICS Sockets, from SC31-8807.

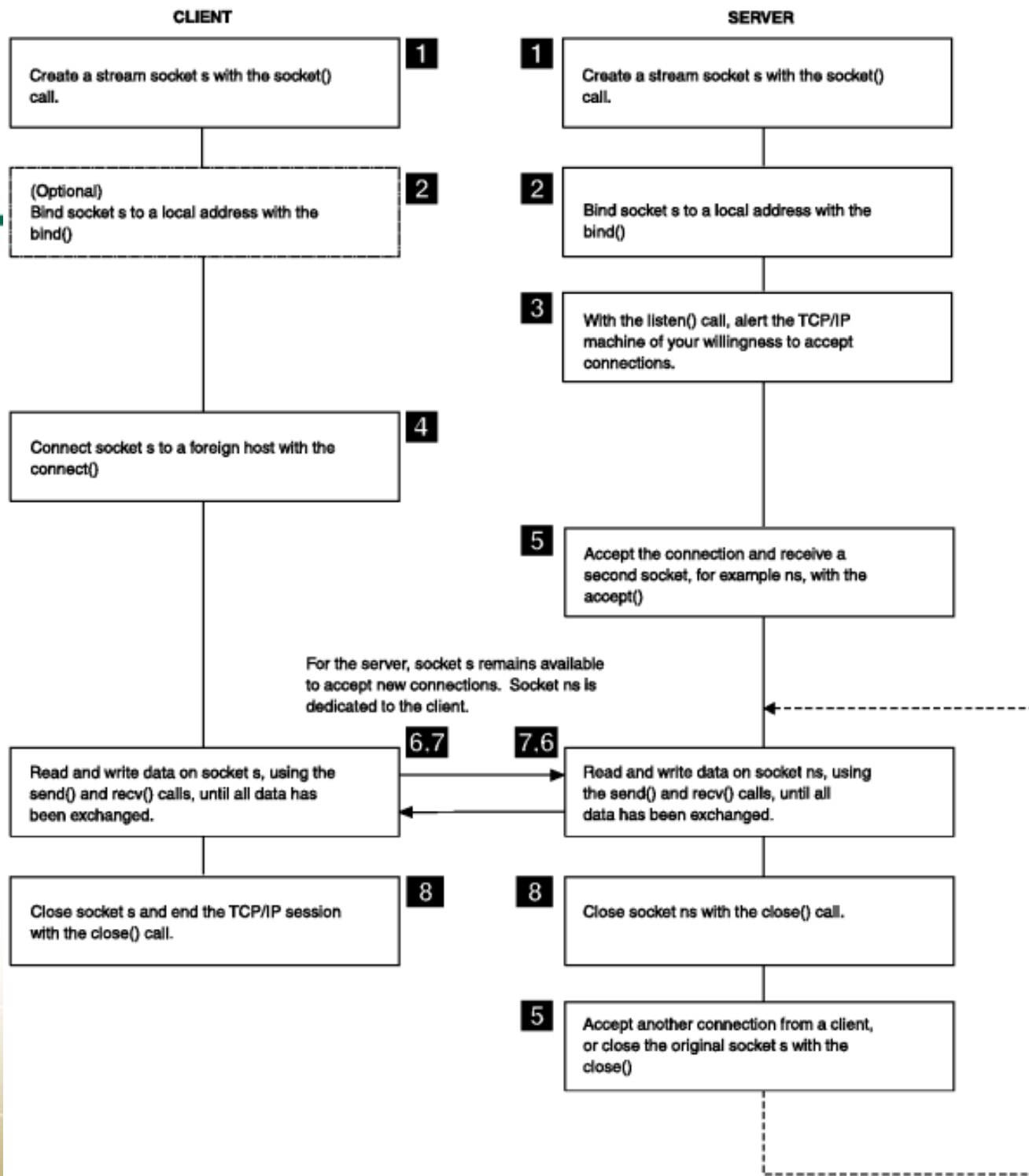


Figure 3., "A typical client server session", from the IBM CICS Socket Interface manual for z/OS 1.4.

# Past/Present/Future Overview

- "An *iterative server* handles both the connection request and the transaction involved in the call itself. Iterative servers are fairly simple and are suitable for transactions that do not last long."
- "However, if the transaction takes more time, queues can build up quickly. In Figure 4, once Client A starts a transaction with the server, Client B cannot make a call until A has finished."

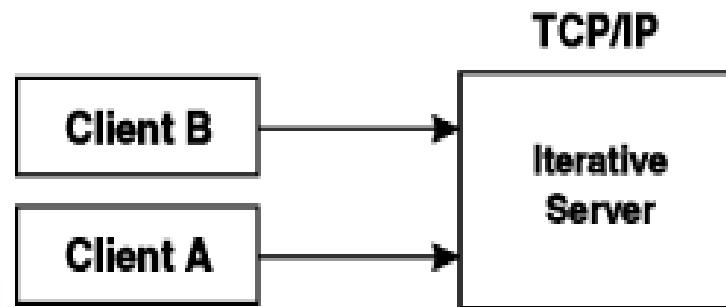


Figure 4. An iterative server, from SC31-8807.

# Past/Present/Future Overview

- "So, for lengthy transactions, a different sort of server is needed -- the *concurrent server*, as shown in Figure 5. Here, Client A has already established a connection with the server, which has then created a **child server process** to handle the transaction. This allows the server to process Client B's request without waiting for A's transaction to complete. More than one child server can be started in this way."
- "TCP/IP provides a concurrent server program called the CICS Listener."

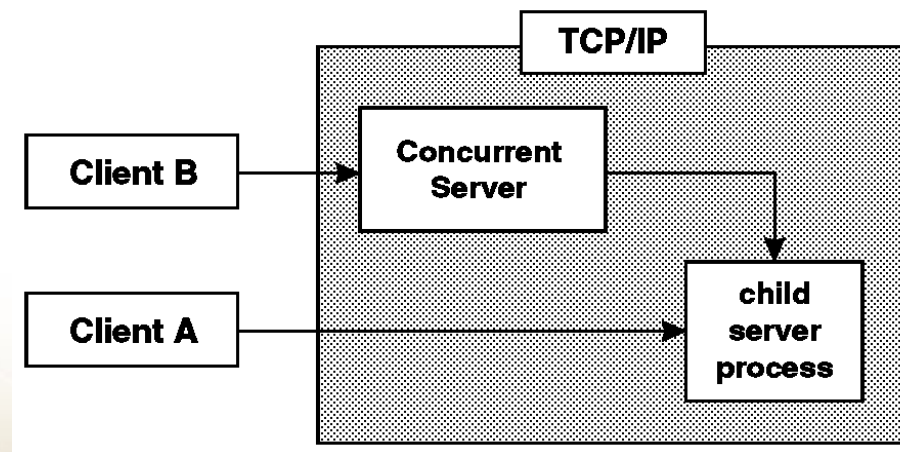
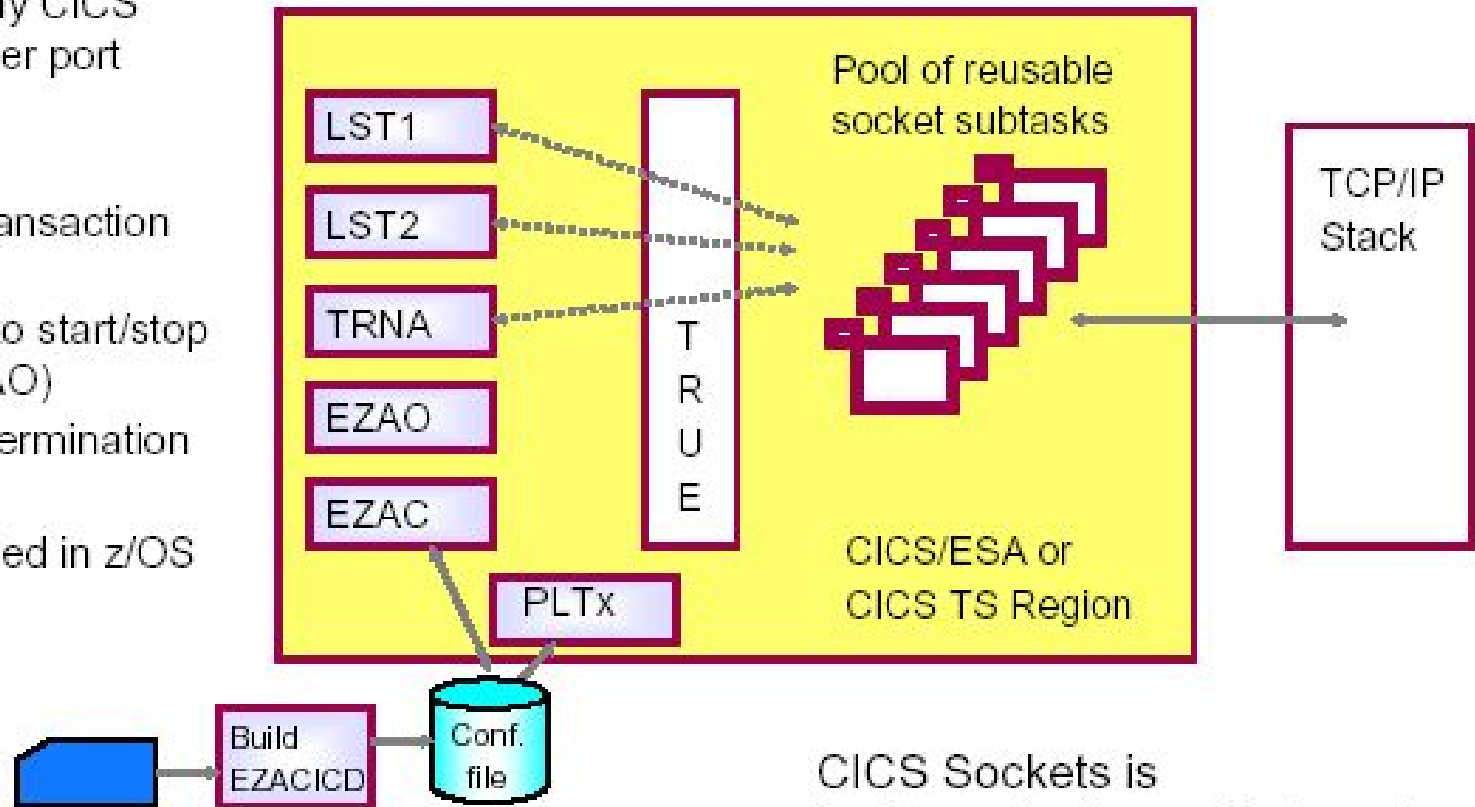


Figure 5. A concurrent server, from SC31-8807.

# Past/Present/Future Overview

## IP CICS sockets structural overview

- Multiple listeners – each instance separately configurable
- Multiple listeners in many CICS regions can share listener port number
- User ID security
- Configuration file and transaction (EZAC)
- Operations transaction to start/stop individual listeners (EZA0)
- PLT-enabled start and termination
- Reusable subtasks
- IPv6 (AF\_INET6) enabled in z/OS V1R5



CICS Sockets is implemented as an External Resource Manager in CICS.

# Past/Present/Future Overview

- The socket **API** is available in the C language (but not C++ yet) and in COBOL, PL/I, or assembler. It includes the following socket calls:
- **Basic** calls: SOCKET, BIND, CONNECT, LISTEN, ACCEPT, SHUTDOWN, CLOSE
- **Read/Write** calls: SEND, SENDTO, RECVFROM, READ, WRITE
- **Advanced** calls: GETHOSTNAME, GETPEERNAME, GETSOCKNAME, GETSOCKOPT, IOCTL, SELECT, GETHOSTBYNAME, GETHOSTBYADDR
- **IBM-specific** calls: INITAPI, GETCLIENTID, GIVESOCKET, TAKESOCKET

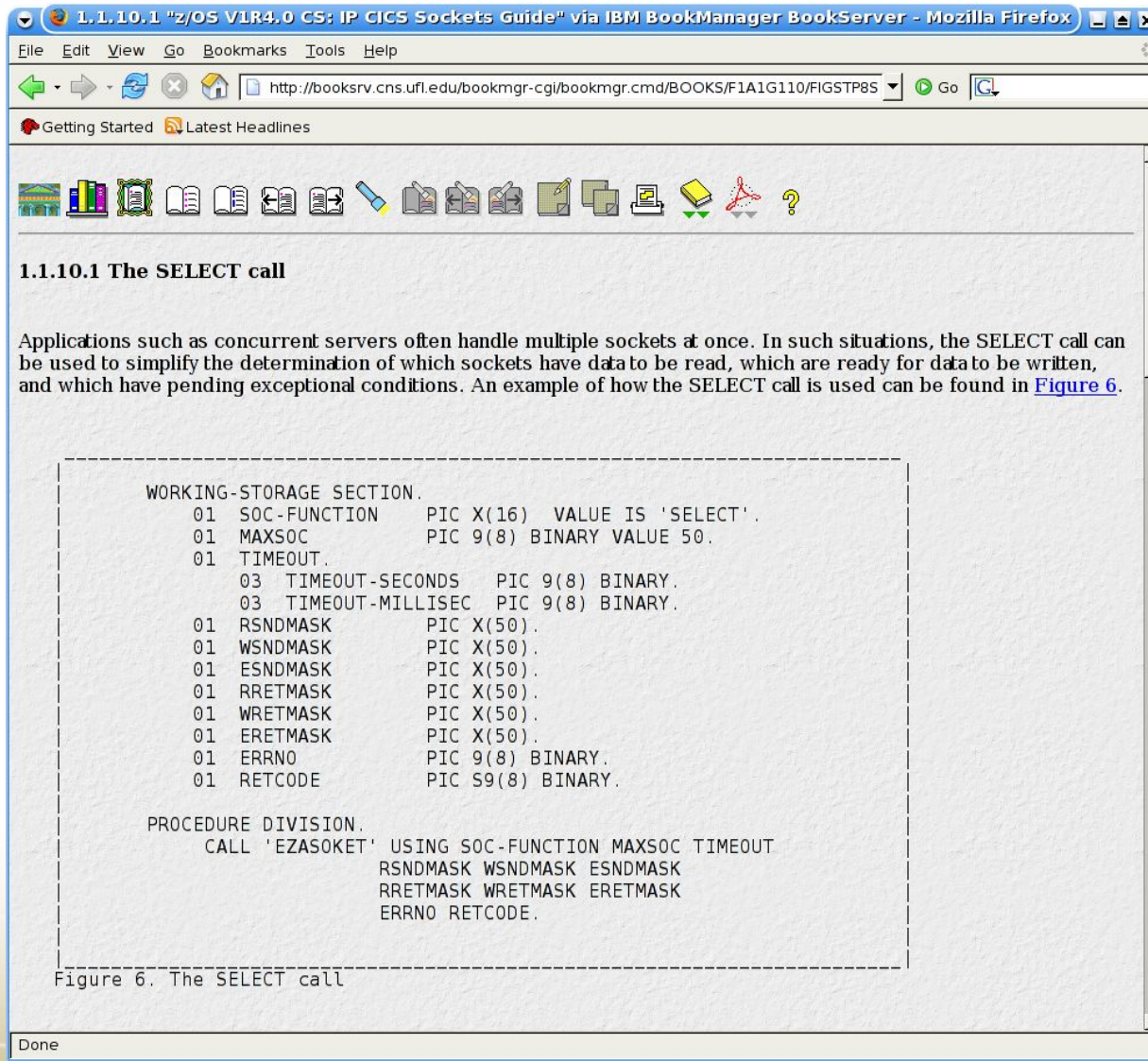
# Past/Present/Future Overview

- The CSI supported API commands for sockets programming (as of z/OS 1.4):

ACCEPT, BIND, CLOSE, CONNECT, FCNTL, GETCLIENTID, GETHOSTBYADDR, GETHOSTBYNAME, GETHOSTID, GETHOSTNAME, GETPEERNAME, GETSOCKNAME, GETSOCKOPT, GIVESOCKET, INITAPI and INITAPIX, IOCTL, LISTEN, READ, READV, RECV, RECVFROM, RECVMSG, SELECT, SELECTEX, SEND, SENDMSG, SENDTO, SETSOCKOPT, SHUTDOWN, SOCKET, TAKESOCKET, TERMAPI, WRITE, WRITEV

# Past/Present/Future Overview

A COBOL example of a SELECT call, Figure 6, from the IBM IP CICS Sockets Guide, for z/OS 1.4.



The screenshot shows a Mozilla Firefox browser window displaying a document titled "1.1.10.1 'z/OS VIR4.0 CS: IP CICS Sockets Guide' via IBM BookManager BookServer". The address bar shows the URL: http://booksrv.cns.ufl.edu/bookmgr-cgi/bookmgr.cmd/BOOKS/F1A1G110/FIGSTP8S. The page content includes a section header "1.1.10.1 The SELECT call" and a paragraph explaining the use of the SELECT call in concurrent servers. Below the text is a code block enclosed in a dashed box, showing COBOL code for a WORKING-STORAGE SECTION and a PROCEDURE DIVISION.

```

WORKING-STORAGE SECTION.
  01 SOC-FUNCTION      PIC X(16)  VALUE IS 'SELECT'.
  01 MAXSOC            PIC 9(8)   BINARY VALUE 50.
  01 TIMEOUT.
     03 TIMEOUT-SECONDS PIC 9(8)  BINARY.
     03 TIMEOUT-MILLISEC PIC 9(8) BINARY.
  01 RSNDMASK          PIC X(50).
  01 WSNDMASK          PIC X(50).
  01 ESNDMASK          PIC X(50).
  01 RRETMASK          PIC X(50).
  01 WRETMASK          PIC X(50).
  01 ERETMASK          PIC X(50).
  01 ERRNO             PIC 9(8)  BINARY.
  01 RETCODE           PIC S9(8) BINARY.

PROCEDURE DIVISION.
  CALL 'EZASOKET' USING SOC-FUNCTION MAXSOC TIMEOUT
                    RSNDMASK WSNDMASK ESNDMASK
                    RRETMASK WRETMASK ERETMASK
                    ERRNO RETCODE.
  
```

Figure 6. The SELECT call

# Past/Present/Future Overview

- The CSI supported API functions for C sockets programming (as of z/OS 1.4):

accept(), bind(), close(), connect(), fcntl(), getclientid(),  
gethostbyaddr(), gethostbyname(), gethostid(), gethostname(),  
getpeername(), getsockname(), getsockopt(), givesocket(),  
initapi(), ioctl(), listen(), read(), recv(), recvfrom(), select(),  
send(), sendto(), setsockopt(), shutdown(), socket(), takesocket(),  
write()

# Past/Present/Future Overview

A server issues the `accept()` call to accept a connection request from a client. The call uses a socket already created with a `socket()` call and marked by a `listen()` call. Here's an example of the format of an `accept()` C socket call:

```
#include <manifest.h> (non-reentrant programs only)
#include <cmanifes.h> (reentrant programs only)
#include <bsdtypes.h>
#include <in.h>
#include <socket.h>
int accept(int s, struct sockaddr *name, int *namelen)
```

## Past/Present/Future Overview

- Critical to our success was the September 1993 **IBM Redbook**, "CICS/ESA and TCP/IP for MVS Sockets Interface", **GG24-4026**, which included a diskette with sample application code. One of the reviewers/editors of this Redbook was Bob Yelavich, from the Dallas Systems Center (at the time). Note that the IBM Redbook web site says the publish date is 04 October 1993.
- The IBM GG24-4026 Redbook is still available in PDF and HTML formats at <http://www.redbooks.ibm.com/>. (Search for gg244026.) The downloads at <ftp://www.redbooks.ibm.com/redbooks> does not currently have the diskette material from this Redbook.

# Past/Present/Future Overview

- **Standard** and **enhanced** listener. We utilize the standard listener, and it has met all of our varied needs, from both a system and application perspective.
- Support for **IPv4** (32-bit addresses) and **IPv6/IPng** (128-bit addresses) in standard and enhanced listener, along with multicast options, starting with z/OS 1.5.
- The IBM Listener program, **EZACIC02**, allows for WLM registration and deregistration in support of connection balancing.

# Past/Present/Future Overview

- IBM has said that the focus with the CICS Socket Interface in z/OS 1.6 is "**reliability**". Note that we have not had a z/OS 1.4 TCP/IP stack nor a CICS Socket Interface unscheduled outage in several years. zSeries z800, z/OS 1.4, CICS TS 2.3, and the CICS Socket Interface have been **very reliable** at our site.
- Almost all "**issues**" with local CICS Socket Interface availability or performance have been related to "external" components, such as network hardware/software, MS IIS "issues", AIX Apache WebSphere plugin "problems", AIX DB2 Kerberos authentication "hangs", or various CICS application "opportunities".

# Past/Present/Future Overview

- The CICS Socket Interface **enhanced** listener, also known as "the enhanced version of the CICS listener", does not expect the standard "header" information from the client, such as CICS transid. IBM recognized that many sites were writing their own listeners, due to a desire to not have to have clients send this header info.
- "**Note:** For the **enhanced** listener, more temporary storage is needed to support passing a larger amount of data to the security/transaction exit and to the child server. Depending upon the size of the data defined in the Listener configuration, temporary storage should be adjusted accordingly."

# Past/Present/Future Overview

- **EXEC TCP** calls CICS command **API** available in TCP/IP for VSE from CSI International:  
*<http://www.e-vse.com/>*
- What does the **future** hold for CICS TS for z/OS and the CICS Socket Interface?

# Past/Present/Future Overview

- Allow users to suppress CICS Tracing. Reference IBM **REQ00076494** from early 2000. (IMASPZAP today?) IBM says that this requirement is met in z/OS 1.7 (Sept. 2005).
- Dynamically **disable** CICS Monitor calls when not coded in the MCT. (z/OS 1.7)
- Allow IP CICS Sockets to exploit CICS **OTE**.
- Allow IP CICS Sockets to exploit **SSL/TLS**. (z/OS 1.7)
- Allow the Listener to **restart** if/when the stack is recycled. See the **Appendix** for a possible solution today.
- What has been "previewed" and "announced" by IBM?

# Past/Present/Future Overview

IBM Announcement Letter 205-034, "Preview: z/OS V1.7 and z/OS.e V1.7...", from Feb. 15, 2005 says:

- "Greater scalability with support for 32 processors in a single image, support for sequential and EXCP data sets larger than 65,535 tracks, more than 255 extents per VSAM component, and CICS® socket enhancements."
- "CICS Sockets and Application Transparent TLS: Applications that use IP CICS Sockets will be able to take advantage of the Application Transparent TLS support. A new CICS Listener configuration option is planned to be provided that can allow listeners using this support to obtain the user ID associated with the connecting client, using the client's digital certificate and a security manager, such as Security Server (RACF). This is designed to allow remote clients to be authenticated without the need for passing user IDs and passwords in application data streams. In addition, the client's user ID can be associated with the child server transactions started by the CICS Listener, allowing access control to CICS resources to be performed based on the client's user ID."

# Past/Present/Future Overview

## IBM Announcement Letter 205-167, "z/OS V1.7 delivers..."

- Networking: "New functions for IPv6 reinforce IBM's commitment to the next generation of IP networks, and enhancements for SNA/EE, CICS® Sockets, and key TCP/IP applications provide new functions that can help improve usability and performance."
- Security: "CICS® Sockets and Application Transparent TLS"
- Availability: "CICS sockets enhancements"
- Planned availability date: September 30, 2005.

# Strategic Considerations

- "**Strategic**", adj., according to <http://www.dictionary.com/>:

1. Of or relating to strategy.



2. a. Important or essential in relation to a plan of action:  
*a strategic withdrawal.*
- b. Essential to the effective conduct of war:  
*strategic materials.*
- c. Highly important to an intended objective:  
*The staff discussed strategic marketing factors.*
3. Intended to destroy the military potential of an enemy:  
*strategic bombing.*

# Strategic Considerations

- Almost 10 years ago, to keep the mainframe strategic, we needed to **quickly web-enable** CICS applications.
- Solution must be very **fast, reliable, secure, inexpensive**, etc.
- **Exploit** the many **strengths** of IBM zSeries hardware, z/OS, Communications Server for z/OS (TCP/IP for MVS), CICS, and the CICS Socket Interface.
- **Leverage** existing CICS applications, and **exploit** the **CSI** to build new/enhanced applications.

# Strategic Considerations

- Continue to provide **sub-second response time** applications, even from the web.
- Continue to provide **very reliable** applications, even from the web.
- Provide **viable alternative** to other campus presentation bullet points which tout "eliminate expensive mainframe technologies".

# Strategic Considerations

- Based upon "**Open Standards**".
- **Low-level** interface.
- Good **performance** and **WLM** enabled.
- Very **few limitations**.
- **Knowledge base** wide and large.
- **Sample code** and applications available.
- **Security?** See "Security Considerations" section.

# Applications

- **Data/file transfer** to/from CICS.
- **CGI/Web** access to CICS.
- **Credit card** payments.
- **Email** from CICS.
- **DNS** lookup.
- Web client and/or server? FTP client and/or server?
- Limited to your **imagination!**

# Applications - Data/file transfer to/from CICS

- First "**proof of concept**" CICS Socket Interface application at our site, and first to be installed in production CICS.
- Worked with CICS application developers at FCLA, the Florida Center for Library Automation.
- **IND\$FILE** "replacement".
- Desire to **own and maintain** the application **source code**, due to IND\$FILE "issues".
- **Visual Basic** client program, and CICS **assembler** child server program.

# Applications - Data/file transfer to/from CICS

- UF Registrar "**Grade-A-Gator**" utilizes HTML form text upload.
- AIX "C" client CGI program, called "uploader", and CICS Assembler child server program.
- Locally written CICS Socket Interface security exit, EZACICSE, coded to verify RACF userids/passwords. (The presentation Appendix has a link to a skeleton sample, ezacicse.asm.)
- UF CICS Socket Interface applications can also verify "GatorLink" ids via DB2 stored procedures. (GatorLink ids are Kerberos based.)

# Applications - CGI/Web access to CICS

- CICS Socket Interface web based **check utility** was written to verify availability of the interface.
- Useful for system and application programmers, and operations staff.
- CGI client code written in Perl (called **sokt.pl**), and CICS child server code written in IBM Assembler (called **DCSOKT00**). CGI code runs in AIX, but is developed and tested in Linux.
- See Appendix for links to [sokt.pl](#), [dcsokt00.asm](#), etc.

# Applications - CGI/Web access to CICS

- **UF ISIS** - UF Integrated Student Information System. Register for classes, get grades, check financial aid, do degree shopping, and much more.
- CGI code written in ANSI C (called "nirvana"), and CICS code written in IBM Assembler. CGI code runs in AIX via Apache web server(s).
- **EAGLE** - Enhanced Application Generation Language for the Enterprise.
- See Appendix for links to ISIS and EAGLE web sites.
- Why is the CGI source code called "**nirvana**"?

# Applications - CGI/Web access to CICS

- **FCLA WebLUIS** - Web Library User Information System. Access to all State of Florida university library catalogs and databases.
- **LU6.2** to CICS Socket Interface application "**conversion**". (LU6.2 no longer in use at our site.)
- CGI code written in C, and CICS code written in IBM Assembler. CGI code runs in AIX via Apache web server(s).
- WebLUIS is being replaced with (non-CICS) "Ex Libris **Aleph** 500 library management system and Web Catalog", which is AIX (COBOL) based.

# Applications - CGI/Web access to CICS

- **UNF Osprey Online** - University of North Florida (in Jacksonville, FL) students can register for classes, get grades, check financial aid, do degree shopping, and much more.
- **UNF Administrative Web Applications** - UNF Faculty and Staff can perform administrative work.
- CGI code written in C, and CICS code written in IBM COBOL. CGI code runs in NT via IIS web server(s).
- UNF is migrating to "UNF Wings", an SCT Banner (non-CICS) ERP solution (utilizing Sun Solaris OS and Oracle DB - Sun Solaris 10 to be open source, btw).

# Applications - Credit Card Payments

- Students can pay fees (etc.) via credit card from CICS, utilizing CASHNet. The CASHNet web site says "CASHNet is the **only** vendor 100% focused on higher ed commerce solutions.". Prior to CASHNet, UF utilized a local "**IPAY**" (NT) system, which connected to IC**VERIFY** system(s).
- CICS client code for this application is written in IBM **COBOL**. **C/C++** and **Java** are utilized on the NT system, running as socket servers for this CICS client.

# Applications - Email from CICS

- IBM CICS SupportPac **CA1H**, "CICS to SMTP Samples":  
*<http://www-306.ibm.com/software/htp/cics/support/supportpacs/individual/ca1h.html>*
- "This **SupportPac** provides sample programs that show methods of interfacing a CICS application with SMTP. The programs are intended to demonstrate various methods of sending EMail from within a CICS application. This is accomplished by utilizing either the CICS SPOOL interface or the CICS TCPIP Sockets interface. The SPOOL Interface requires the use of the OS/390 SMTP server. The Sockets Interface may utilize any SMTP server accessible on the TCPIP network. There is also an example CICS based mail server that could be used for receiving EMail in CICS. Examples are provided in Assembler, COBOL and C/370."

# Applications - DNS Lookup

- Local DNS testing/verification COBOL program, DCSO2500:

*<http://nersp.cns.ufl.edu/~sftware/share97/dcso2500.cob>*

- This program utilizes a LINK to IBM supplied program EZACIC25. This is similar to the "nslookup" command found on many platforms. Here's an example of what this program does from a clear screen in CICS (3270-based):

```
so25 www.ufl.edu (results in)
SO25 WWW.UFL.EDU           = 128.227.74.67
```

# Security Considerations

- CICS Socket Interface **security exit point** is available for local customization. Default name is **EZACICSE**, but this can be modified. See Appendix for a sample EZACICSE.
- z/OS 1.5 includes DSECT **EZACICSX**, in OS dataset prefix.SEZACMAC. This DSECT maps the CICS Socket Interface COMMAREA. (Roll your own if not z/OS 1.5+.)
- Through at least z/OS 1.6, the CICS Socket Interface does not have support for **SSL/TLS**. See the IBM z/OS 1.7 announcement letter, 205-167, for details on what has been announced in this regard.

# Security Considerations

- Until **SSL/TLS** is available in the CICS Socket Interface, you could "roll your own" encode/decode (encrypt/decrypt) solution, utilizing the security exit point.
- Another option might be an **encrypted VLAN** (Virtual Local Area Network) solution.
- For web access, we utilize SSL enabled web servers for CGI access on an **internal intranet**. Clients must utilize SSL enabled web browsers.
- If complete "**end-to-end**" **SSL** is required today, consider the existing **CICS Web Support** and **SOAP for CICS** capabilities (aka CICS Web Services).

# CICS Regions Consolidation

- Our site recently **consolidated CICS regions**, for a variety of reasons. We reduced our total configured regions by 2 - 2 test regions were combined into 1, and 2 production regions were combined into 1.
- MRO/Function Shipping **CPU consumption** reduction considerations.
- Changes in **workload** and application mix.
- Now using **2 listeners** in several CICS regions.
- Required **minor modifications** to the CICS Socket Interface configuration (VSAM) file, EZACONFG.

# CICS Regions Consolidation

- Required minor modifications to local CICS Socket Interface Security exit, **EZACICSE**, and local health checker program, **DCNEIP00**.
- Utilized **same port numbers**, requiring no changes to CICS application development group CGI or CICS application code.

# Miscellaneous Considerations

- **ASCII/EBCDIC** character conversion "opportunities", including CPU overhead, and "fixed" conversion table module(s). Review APAR PQ70462 - PROBLEM CONCLUSION: New parts EZACIC14 and EZACIC15 have been created and can now be used as alternatives to EZACIC04 and EZACIC05. They use the standard "IBM 1047 to ISO 10646" mapping table.
- **CICS Monitoring and Trace Table** "opportunities".
- **Debugging** and packet trace "opportunities".
- TSO **netstat** can be your friend:  
*netstat sockets*  
*netstat allconn (client cicstest*

# Miscellaneous Considerations

- Useful **TCPM/TCPCICS** log search strings:  
*abend, auth, detach, disab, err, expect, fail, inv, max, null, timeout*
- We utilize the IBM Tivoli **Omegamon** for CICS TCPT command. Check with your CICS monitor vendor for specific details.
- Several vendors also offer z/OS **TCP/IP monitors**. Useful with CSI, and CICS Sockets Domain applications, in addition to other z/OS TCP/IP applications.

# Miscellaneous Considerations

- IBM Tivoli Omegamon for CICS TCPT display:

```

x3270 <4>
File Options
-----
> PF1 Help      PF3 Back      PF4 Main Menu  PF7 Up        PF8 Down
-----
>
TCP/IP SOCKET INFORMATION
TCPT
+
+          General Information
+  TCP/IP address space . : TCPIPP      Global work area . . . : 0004B334
+  User exit module . . . : EZACIC01    User exit status . . . : Enabled
+  Number of listeners . . : 2          Error messages queue . : TCPM
+
+          Applications Information
+  Tran      Task      Current      Return      Appl
+  ID        Number    Socket      Request     Code      State
+  -----
+  CSK2      32         0  SELECTEX
+  CSKL      31         0  SELECTEX
+
+          Listeners Information
+  Tran      Listener
+  ID        State
+  -----
+  CSKL      Select
+  CSK2      Select
-----
TEDVI013  :00.0 001/002
  
```

# Summary

- IBM zSeries z800, z/OS 1.4, CICS TS 2.3, and the CICS Socket Interface have been **very reliable** at our site.
- The **CICS Socket Interface** is easy to install, configure, and utilize, continues to be enhanced by IBM, and is now even considered **strategic!**
- Our local CICS application developers continue to **exploit** the CICS Socket Interface, with new and enhanced applications.
- Many of our CICS application development groups **prefer** the "**low-level**" features of assembler and the CICS Socket Interface.

# Summary

- If starting to web enable our CICS applications today, we'd be looking closely at utilizing **CICS Web Support** and **SOAP for CICS**.
- For performance improvements in the CSI, we'd like **REQ00076494** implemented, and serious consideration given to utilizing EXEC CICS RETURN IMMEDIATE, instead of EXEC CICS START, for child server tasks.
- **Thanks**, and have a great time for the remainder of the conference, and have a safe trip home.
- **Questions?** Comments? *Random thoughts?*

# Appendix

- **IBM CICS:**  
*<http://www.ibm.com/cics> (which recently resolved to)  
<http://www-306.ibm.com/software/htp/cics/>*
- **IBM White Paper G224-7324: "Delivering e-business access to CICS: strategic options":**  
*<http://www-306.ibm.com/software/htp/cics/library/indexes/whitepapers.html>*
- Any of **Alfred B Christensen's** great SHARE presentations - he's presenting 10 times at this SHARE!  
*<http://www.share.org/> (check the proceedings)*
- Ken Porter's (IBM z/OS Communications Server Development) presentation PDF, "Delivering e-business access to CICS - **IP CICS Sockets**", available at:  
*<http://www-306.ibm.com/software/htp/cics/library/cicstsforzos31.html>*

# Appendix

- **IBM Redbooks:**  
*<http://www.redbooks.ibm.com/>*
- **IBM Architecting e-business Access to CICS Update:**  
*<http://publib-b.boulder.ibm.com/redpieces/abstracts/sg245466.html?Open>*
- **IBM Mainframe Servers - zSeries:**  
*<http://www.ibm.com/zseries/>* (which recently resolved to)  
*<http://www-1.ibm.com/servers/eserver/zseries/>*
- **IBM USA Announcement Letters:**  
*<http://www.ibm.com/news/usalet>*

# Appendix

- **IBM z/OS Communications Server:**  
*<http://www-306.ibm.com/software/network/commserver/zos/>*
- **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."

# Appendix

- The **University of Florida (UF)**:  
*<http://www.ufl.edu/>*
- The **Association of American Universities (AAU)**:  
*<http://www.aau.edu/>*
- **UF Computing & Networking Services (CNS)**:  
*<http://www.cns.ufl.edu/>*
- **CICS at UF**:  
*<http://cics.ufl.edu/>*
- The **University of North Florida (UNF)**:  
*<http://www.unf.edu/>*

# Appendix

- **UF ISIS** (Integrated Student Information System):  
*<http://www.isis.ufl.edu/>*
- **EAGLE** (Enhanced Application Generation Language for the Enterprise):  
*<http://eagle.ufl.edu/>*
- **FCLA** (Florida Center for Library Automation):  
*<http://www.fcla.edu/>*
- **WebLUIS** (Web Library User Information System):  
*<http://webluis.fcla.edu/>*

# Appendix

- **HostBridge** (CICS Integration) and free downloads:  
*<http://www.hostbridge.com/downloads>*
- **SAS** Sample Code:  
*<http://ftp.sas.com/techsup/download/SASC/sample/>*
- **CASHNet** Higher Ed Commerce Solutions:  
*<http://www.cashnet.com/>*
- **ICVERIFY** Payment Processing:  
*<http://www.icverify.com/>*
- **TCP/IP for VSE** from CSI International:  
*<http://www.e-vse.com/>*

# Appendix

- Request for Comments (**RFCs**) Internet Archive:  
<http://www.faqs.org/rfcs/>
- IBM CICS SupportPac **CA1H**, "CICS to SMTP Samples":  
<http://www-306.ibm.com/software/hfp/cics/support/supportpacs/individual/ca1h.html>
- SHARE CICS Project:  
<http://www.share.org/cics>



# Appendix

- **EZACICSE** sample security exit skeleton, **DCNEIP00** sample CICS Socket Interface "health checker" (and register equates macro), **sokt.pl** Perl CGI client and **DCSOKT00** assembler child server web-based "check utility", **DCSO2500** COBOL DNS lookup program, and notes, are available at:  
<http://nersp.cns.ufl.edu/~sftware/share97/appendix.html>  
<http://nersp.cns.ufl.edu/~sftware/share97/appendix2.html>

# Appendix

- **SHARE 104** session "User Experience: CICS Sockets - A Strategic Solution" (Feb./March 2005):  
*<http://nersp.cns.ufl.edu/~sftware/share104/s1054sw.pdf>*
- **SHARE 97** session "Exploiting the CICS Socket Interface" (July 2001):  
*<http://nersp.cns.ufl.edu/~sftware/share97/>*
- **SHARE 96** session "Exploiting the CICS Socket Interface" (February 2001):  
*<http://nersp.cns.ufl.edu/~sftware/share96/>*
- **SHARE 93** session "CICS Socket Interface - Tips and Techniques" (August 1999):  
*<http://nersp.cns.ufl.edu/~sftware/share93/>*

# Appendix

- **SHARE 92** session "CICS Socket Interface - Tips and Techniques" (February 1999):  
*<http://nersp.cns.ufl.edu/~sftware/share92/>*
- **SHARE 91** session "CICS TCP/IP Sockets - User Experience" (August 1998):  
*<http://nersp.cns.ufl.edu/~sftware/share91/>*

# Presentation Information

- The **Slackware Linux Project**:  
<http://www.slackware.com/>
- **OpenOffice.org 1.1.4 "Impress"**:  
<http://www.openoffice.org/>  
(File -> Export as PDF)  
(Used SHARE provided PowerPoint template.)
- **IBM (Lenovo) ThinkPad T40 2379-D5U**:  
<http://www.lenovo.com/think/us/en/>

