We're STILL extending our CICS applications

Steve Ware, UF (with application info. assistance from Alan Cook)

Session 1068, SHARE 114
March 17 (Wed.) 2010, 1:30 PM
Washington State Convention Center, Room 303

http://nersp.cns.ufl.edu/~sfware/share114/s1068sfw.pdf (Updated: 03-17-2010)
Abstract

In this session, we'll share information on the use of CICS at the University of Florida. Not only are new applications being developed and deployed in CICS, but mission critical applications are made more accessible with CICS's support of Web services.

CICS is the leader in transaction processing, providing extremely fast response time, very high availability, and a very modern and feature-rich application processing environment.

CICS is strategically positioned for enterprise class application serving, whether for new and/or extended mission critical applications.
Abstract

CICS TS for z/OS 4.1 became generally available in late June 2009, and IBM shows no signs of slowing down regarding future releases of CICS TS for z/OS. At UF, we installed CICS TS 4.1 in early July 2009, and have enjoyed 100% scheduled CICS availability. The IBM System z mainframes, running the IBM z/OS operating system, provide one of the most modern, resilient, and capable computing platforms.

The past, present, and future of CICS and the mainframe are bright - bring your sunglasses to this session as we shine a light on the best of breed - CICS and the mainframe!
Disclaimer

Standard disclaimers apply.

Any opinions expressed are the opinion of the author/presenter 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

- Let's start with a quick tour!
- Introduction
- Extending/enhancing CICS applications
- What about mainframe application development?
- Why CICS?
- Why the Mainframe?
- Summary and Q&A
- Appendix and Additional Information
- Abbreviations (and a bit of Glossary)
A quick tour!
A quick tour! (cont.)
A quick tour! (cont.)

Display On-line Messages and Codes

Type the required message identifier, then press Enter.

Component ID. . . . (for example, TC for Terminal Control
FC for File Control, etc.)
This field is required for messages in the form DFHxxyyy, where xx is the Component ID.

Message Number. . . aj05 (for example, 1060, 5718, or Abend Code
such as ASRA, etc.)

F3=Exit to CICS
A quick tour! (cont.)
A quick tour! (cont.)
A quick tour! (cont.)

CICS Web Interface

CICS Web Support 3270 screen emulation

A005

EXPLANATION: An unhandled exception has been caught by the Java environment setup class, Wrapper, as an InvocationTargetException from the user class.

The callUserClass method of Wrapper detects this, sets return code INVOCATION_TARGET_EXCEPTION and invokes native method setAbend to abend the task.

SYSTEM ACTION: The task is abnormally terminated with a CICS transaction dump.

USER RESPONSE: See related messages to determine the reason for the original Exception.

MODULE: DFICICS
A quick tour! (cont.)

EXPLANATION: An unhandled exception has been caught by the Java environment setup class, Wrapper, as an InvocationTargetException from the user class.

The callUserClass method of Wrapper detects this, sets return code INVOCATION_TARGET_EXCEPTION and invokes native method setAbend to abend the task.

SYSTEM ACTION: The task is abnormally terminated with a CICS transaction dump.

USER RESPONSE: See related messages to determine the reason for the original Exception.

MODULE: DFHCICS
A quick tour! (cont.)

**Web Sample 1**

**Inbound Client Request Information:**

- **Method:** GET
- **Version:** HTTP/1.1
- **Path:** /CICS/CWBA/DFJ8JWB1
- **Request Type:** HTTP/YES
- **Query String:** null

**HTTP headers:**

Value for HTTP header User-Agent is "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.16) Gecko/2009120206 Firefox/3.0.16"

**Browse of HTTP Headers started**

- **Name:** Host Value: zhost
- **Name:** User-Agent Value: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.16) Gecko/2009120206 Firefox/3.0.16
- **Name:** Accept Value: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
- **Name:** Accept-Language Value: en-us,en,q=0.5
- **Name:** Accept-Encoding Value: gzip, deflate
- **Name:** Accept-Charset Value: ISO-8859-1,utf-8;q=0.7,*;q=0.7
- **Name:** Keep-Alive Value: 300
- **Name:** Connection Value: keep-alive
- **Name:** Authorization Value: Basic c2ZZJ0IkV5UXJBEQ3

**Browse of HTTP Headers completed**

**TCP/IP Information:**

- **Client Name:** zclient.name
- **Server Name:** zhost
- **Client Address:** ...
- **ClientAddrNo:** ...

---

*Session 1068, SHARE 114, Seattle, WA, Steve Ware, UF.*
A quick tour! (cont.)

Hello from HelloCICS.php running in CICS TS!
A quick tour! (cont.)
A quick tour! (cont.)
A quick tour! (cont.)
A quick tour! (cont.)
A quick tour! (cont.)
Introduction

- UF CNS, University of Florida Computing & Networking Services (formerly known as NERDC), manages the primary data centers in and around the Gainesville, FL campus.

- Currently utilizing an IBM z9 BC 2096-S02 with 16GB, running z/OS 1.9 (1.10, 1.11), CICS TS 4.1, DB2 V8, RACF, JES2, etc.

- 3 LPARs - 1 internal "sysprog sandbox", 1 test "alternate", and 1 production or "primary".

- We have 9 CICS regions configured, and run ~.5M prod. transactions/weekday, and ~1.25M on peak load days.

- 2 internal/test sandbox, 2 development/test, 3 test/QA, and 2 production CICS regions currently configured.
Introduction (cont.)

- 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

- UF is one of the largest universities in the U.S., public or private

- ~50K enrolled and ~250K alumni
Introduction (cont.)

• 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

• ~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 (for file/data transfer, email, web enablement, etc.)

• Several internal CICS applications written in C/C++ and REXX. Java in CICS has been thoroughly IVP tested, occasionally utilized in CICS for Web service validation, and we're considering Java in CICS applications, possibly including PHP in CICS
Introduction (cont.)

• Are we now considered a **Nouveau** CICS site?

• CICS Web Services in production since Sept. 2006: http://docweb.cns.ufl.edu/update/u0610cics/u0610cics.html

• CNS & UF Registrar Implement "MyStudentBody.com" Requirement using CICS Web Services.

• "On Friday, September 22nd, 2006, UF CNS CICS systems staff and UF Office of the University Registrar application staff implemented a new, secure (https) CICS Web service, with CICS acting as the service requester, for the MyStudentBody.com UF health requirement. The new capabilities introduced to support this initiative pave the way for implementation of encrypted Web services accessing real-time student data, making applications more accurate, serving the UF community better."
Extending/enhancing CICS applications

- CICS Socket Interface
- CICS Web Services
- CICS Web Support
- CICS Event Processing (non-invasive)
- The CICS Explorer (includes event binding editor)
- Atom feeds, RESTful interfaces, mashups, Web 2.0 (initially in SupportPac CA8K, now integrated into CICS TS V4.1)
- Service Component Architecture support via RDz tooling
- CICS Service Flow Runtime for CICS business services (or service flows)
Extending/enhancing CICS applications (cont.)

- Java (Java 6 in CICS TS V4.1), PHP, REXX
- CICS Channels and Containers (eliminate 32K COMMAREA limitations)
- CICS Document API
- ID Propagation (distributed identities) enhancements (in CICS TS V4.1 with recent service and enhancements starting in z/OS 1.11)
- CICS Management Client Interface API using RESTful principles for HTTP client applications, including CICS Explorer
- Home grown tools, exits, and ????
Extending/enhancing CICS applications - some UF examples

- Presentation logic separated from business logic, as IBM has recommended (for decades?)
- TeleGator developed for initial student access to 3270/BMS applications via telephone ("screen scraping" - no longer in use)
- Web enabled 3270/BMS applications with the CICS Socket Interface for student, faculty and staff access - ISIS
- Original "C" cgi code in AIX recently converted to PHP in Linux (Intel) with Apache web servers
- CICS Socket Interface also used for email, CICS as http client (CICS Web Support would be used today), XML messaging, and more
Extending/enhancing CICS applications - some UF examples (cont.)

- Local modular "tooling" and CICS application generation with web based EAGLE
- EAGLE has "topics", is table driven, has "ESP" scripting language, and supports dynamic DB2
- Authentication via DB2 PIN (no longer in use), RACF userid, GatorLink id (kerberos based via DB2 stored procedures), and most recently, Shibboleth
- Authorization via "roles" (via DB2 tables)
- Uses a decision tree with a realtime business logic engine, for applications such as student fees and admissions decisions
- Includes VSAM query tools utilizing batch, CICS and DB2
Extending/enhancing CICS applications - some UF examples (cont.)

• CICS Web Services initially used for MSB (mystudentbody.com mentioned earlier), but no longer in use - why?

• MSB vendor made Web Services interface (WSDL) changes without notification to customers

• MSB vendor made changes to their infrastructure that caused very slow response time, including transmission timeouts (CICS PIPELINE resource "respwait" option)

• UF now using echeckuptogo.com (San Diego State Research University Foundation)
Extending/enhancing CICS applications - some UF examples (cont.)

- CICS Web Services used for CICS proof of concept to securely access VSAM files, and now for CICS Web Services health checks
- CICS Web Services used for integration between ISIS (CICS) and Oracle PeopleSoft Student Financials (BizTalk)
- Alan Cook wrote a CICS Web Services "wrapper" (subroutine) program (Assembler), using CICS Channels and Containers, with the endpoint URIs stored in DB2 tables
- Errors handled via one of Alan's email subroutines (SMTP in CICS via the CICS Socket Interface)
- UF now has 10 CICS Web Services enabled in production (4 providers and 6 requesters)
Extending/enhancing CICS applications - some UF examples (cont.)

- Some AJAX functionality now being used by UF CICS student information application developers (Javascript and DB2 utilized)

- Alan is looking at utilizing CICS Channels and Containers to "objectize" some of his application storage management, in addition to use for CICS Web Services

- A planned new CICS application for graduate admissions processing with electronic referrals and record matching

- A fairly new CICS mailroom application has dramatically improved processing time, reduced costs, and improved business processes - handles items such as transcript requests, admissions form processing, package logging, check handling and redistribution, and more
What about Mainframe application development?

- Locally written at UF: EAGLE
- Other "home grown" - a recommended option? (Owning the source code to your core business applications is "priceless").
- IBM EGL - Enterprise Generation Language
- IBM Rational Developer for System z
- IBM Rational Business Developer
- Eclipse
- Sun NetBeans
- Other?
Why CICS?

• IBM's CICS is the planet's premier OLTP (On-Line Transaction Processing) system. CICS Rocks! Stick With CICS! Anyway...

• Enterprise caliber, high performance, fast response time, exceptional throughput and reliability, resilient, large installed base, capabilities continually enhanced, large variety of applications and tools, superior technical support from IBM, expansive API, feature-rich application processing environment - and so much more!

• CICS is sometimes referred to as an "Application Server" and/or "Middleware". IBM has positioned CICS Transaction Server in the WebSphere "application and transaction infrastructure".

• IBM says "CICS Transaction Server for z/OS v4.1 is a modern, dependable, and cost effective application platform."
Why CICS? (cont.)

- Support for "heritage" technologies such as VTAM and SNA networking and 3270 devices. Some things are still better via a CHUI (CHaracter User Interface) than a GUI (Graphical User Interface). Data entry and scripting are examples where a CHUI shines.

- Support for "modern" technologies, including Web services, SOA, SOAP, Java, JVMs, EJBs, C/C++, SSL, XML, and much more.

- Support for most operating systems - our focus will be z/OS.
Why CICS? (cont.)

- CICS offers flexible intercommunication facilities allowing it to be supported and configured across a variety of networks.

- CICS data management is comprehensive and includes support for major database management systems such as DB2 - this also includes a high performance CICS DB2-Attach facility. Data can be in databases, OS datasets, datatables, or even within CICS itself. Other database support includes Oracle, IMS, etc.

- CICS and VSAM continue to be enhanced, such as with the VSAM RLS function of DFSMS, and DFSMStvs (Transactional VSAM Services), for CICS and batch. Note that a coupling facility is required for these optional VSAM components.
Why CICS? (cont.)

• What about CICS for Linux?

• IBM Statement of Direction: "IBM recognizes the significance and benefits of the Linux operating system to CICS customers who have chosen the TXSeries for their applications. It is IBM's intention to release a CICS offering on the Linux platform in 2005..."

• Note that CICS for zLinux is not mentioned - only TXSeries.

• IBM iPRPQ 7J0468 announced Linux for xSeries availability of TXSeries for Multiplatforms v6 on 2005-12-20. Contact your IBM software business partner - an order requires IBM (Hursley) Lab approval.
Why CICS? (cont.)

• Programming language support includes Java, C/C++, Enterprise COBOL, Assembler, PL/1, REXX. Object oriented programming support, even in Assembler, with IBM's High Level Assembler. High speed XML parsers and CICS translator integration available with some compilers. What about PHP? See IBM SupportPac CA1S: REST support in CICS using PHP.
Why CICS? (cont.)

- Expansive API (Application Programming Interface), SPI (System Programming Interface), and XPI (eXit Programming Interface). You can truly "make CICS dance" anyway you'd like.

- The CICS API helps isolate the CICS application programmer from the operating system, allowing application programmers to focus on application development and business solutions.

- The CICS SPI allows the CICS system programmer to enhance the local CICS environment with SET and INQUIRE system capabilities.

- The CICS XPI allows the CICS system programmer to enhance and expand upon the delivered CICS capabilities in the many supplied CICS exit points.
Why CICS? (cont.)

- Additional programming APIs for CPSM, Java (including JCICS), etc.

- CICS ships with integrated debugging (CEDF/CEDX) and support tools (CEBR, CECI, CEDA, CEMT, CETR, CICSPlex SM, etc.), a large sample library, and sample applications with source code.

- IBM and other vendors also have very capable optional CICS testing, debugging, and monitoring tools.
Why CICS? (cont.)

- Many IBM CICS SupportPacs, vendor products, freely available source code, and helpful web sites and discussion lists. Speaking of SupportPacs, SOAP for CICS is a good example of how quickly IBM added this support to CICS:

  1. A free download was made available by IBM. The download included code, documentation, and samples. IBM also provided a good SOAP for CICS discussion list.

  2. An optional no-charge feature was added to CICS TS 2.2 and CICS TS 2.3, using a CALLable interface.

  3. It's fully integrated into CICS, starting with CICS TS 3.1, via new and enhanced EXEC CICS API commands, etc.
Why CICS? (cont.)

- CICS is designed and developed for high performance, availability, and capability. Examples include:
  1. CICS storage protection and transaction isolation
  2. CICS Language Environment (LE)
  3. CICS and the MVS Logger
  4. CICS-DB2 Attach Facility
  5. CICS-WMQ connection and WebSphere MQ group attach
  6. CICS Java Support and CICS JVMs
  7. CICS domain architecture
  8. CICS Web Services and CICS Web Support
Why CICS? (cont.)

- CICS TS 4.1, CICS Transaction Server for z/OS V4.1, is the latest release from IBM.

- The CICS TS 4.1 announcement letter even mentions "As part of the multi-release IP interconnectivity (IPIC) initiative", which implies there's still lots more to come from IBM.

- Our experience with CICS TS 4.1 so far has been 100% scheduled availability. Exceptional reliability, with sub-second response time from the web! What's not to like?

- The latest CICS Information Centers for CICS TS are (open standards) Eclipse based, with support for Windows and Linux. This Linux support has been thoroughly tested and utilized by the presenter (thanks, IBM!).
Why CICS? (cont.)

In the CICS TS 4.1 Announcement letter, IBM describes these three "themes":

1. **Compete**: Making it easier to create, extend, and reuse applications quickly, to meet changing business needs.

2. **Comply**: Helping to ensure and demonstrate effective management control over business applications and IT facilities.

3. **Control**: Helping IT staff to perform their tasks more effectively, while assuming a mixture of skill levels.
Why CICS? (cont.)

IBM says:

"The key functions provided by CICS TS V4.1 support two or more of the themes described earlier, simultaneously. These key functions are:

- Support for event processing
- Atom feeds from CICS
- The CICS Explorer

For example, in support of multiple themes in this version of CICS TS, the ability to generate business events without changing application programs both reduces cost and complexity and delivers compliant flexible business solutions."
Why the Mainframe?

• A personal description of an IBM System z mainframe: An enterprise class of modern, flexible, scalable, and resilient computing servers.

• Or in more detail: Enterprise computing system, with lots of processing power, continually being enhanced, with very wide I/O bandwidth, comprehensive instruction set, efficient resource sharing, very capable resource management capabilities, 64-bit architecture, supporting many simultaneous processes/programs, all efficiently managed with an enterprise caliber OS and enterprise class subsystems, such as CICS.
Why the Mainframe? (cont.)

• Latest mainframes from IBM include:

  System z10 BC and EC. The "zero downtime" and "a to z" enterprise servers, up to 64 processor units (general purpose, specialty and/or spare), *many* LPARs, and all 64-bit enabled (with 24-bit and 31-bit still supported).

• Do some enterprises not disclose use of the mainframe, due to "competitive advantage"?

• Have you seen recent discussions about "transactions per watt hour", or "transactions per joule", or "transactions per BTU"?
Why the Mainframe?

• "Mainframe renaissance" (once again?) in recent years. "The legacy lives on!" "Big Iron Staying Power."

• We have 2 processors with 16GB main memory in our z9 BC, which has simultaneously run all of UF and UNF financial and student administration, all of the State of Florida universities LUIS (Library User Information System) and FACTS (Florida Academic Counseling and Tracking for Students). Compare this with the non-mainframe "solutions" with literally hundreds of processors, near terabytes of main memory, many times more DASD, power consumption, floor space, system administrators, etc. Add it up - which is less expensive? Which consistently provides better response time? Which is easier to recover in a disaster recovery scenario?
Why the Mainframe? (cont.)

• Have you priced ERP software and other non-mainframe "enterprise" software costs recently? Do you have enough cooling, floor space, and power to attempt to run non-mainframe system hardware? Would you like to pay "per-seat" software license costs, when per-seat includes ~50K students and ~12K faculty and staff?

• Have you ever heard about "Re-boot Hill"? http://actscorp.com/reboothill.htm

• Some organizations gleefully talk about their non-mainframe initiatives, but tend to "clam up" when these initiatives go way over budget and/or can't deliver as promised and/or fail miserably. Let's call these "successful failures".
Why is it ok to spend *more* money on non-mainframe solutions, and in addition, provide *poorer* service? What follows is a recent personal example of such nonsense...

A recent letter addressed to "Dear ... Participant" that I received from a large "Financial Services" company, included the text:

"To bring you these and other benefits aligned to your needs, we have been transforming virtually all aspects of our organization. Along the way, we have occasionally and inadvertently inconvenienced some participants with processing problems and long wait times to speak with our consultants on the phone. We regret these problems, and we are working day and night to resolve them and prevent their reoccurrence..."
Why the Mainframe? (cont.)

- "Never trust a computer you can lift ;-)

http://linux390.marist.edu

Linux for S/390
Why CICS and the Mainframe - Now?

• The newest capabilities and enhancements to both CICS and the mainframe can be utilized, while more "mature" programs and applications continue to run. For example, newer 64-bit features can be utilized, while older 24-bit and 31-bit programs continue running along fine. Assembler, COBOL, C/C++, and PL/1 CICS applications can be utilized or enhanced, and/or Java can be added to the application mix. Lots of choices and flexibility.

• Multiple releases of CICS can be run simultaneously, making for smooth and phased release migration (see the IBM CICS Installation Guide for additional details).
Why CICS and the Mainframe - Now?

- CICS has provided upward compatibility for most system and application code for many years.

- z/OS and mainframe upward compatibility is also remarkable. We have code from the 70's still running in the latest releases of z/OS. This is not always recommended, but sure is handy in many cases.

- CICS and z/OS migration is made much easier, because backing off an upgrade (or other system level change) is much easier when compared to other computing environments. z/OS can easily be re-IPLed off of the prior SYSRES, or CICS can easily be reloaded with prior run datasets. We haven't had to back off of a CICS or z/OS migration for many many years, btw.
Why CICS and the Mainframe - Now?

• Both CICS and mainframe capabilities and features continue to be enhanced, seemingly faster than some customers can keep up! In my opinion, this is all good news.

• Almost any good IT professional can be taught about almost any platform. Why not teach them about the best of breed - CICS and the mainframe?

• Why not invest in CICS and the mainframe? Remember, it's not an expense, it's an investment! And when choosing, be sure to "choose wisely"!

• Why not now?
Summary

• Computing pioneer Seymour Cray once said, "What would you rather have to plow a field - two strong oxen or 1,024 chickens?"

• Billions and billions of transactions processed daily - CICS is truly a software "star" ;-).

• UF continues to exploit the many inherent advantages of CICS, z/OS, and the mainframe.

• Students especially like the sub-second response time, even from the web! Faculty and staff also appreciate this good response time, but they might be a bit more patient than the students ;-).
Summary

• **Do all the math** when making computing decisions - you just might calculate that the mainframe costs are very competitive, and that the capabilities of the mainframe are much better than other platforms. **TCO** and **TCU** are both important.

• Personally, some business relationships are based upon whether or not the business utilizes a mainframe. Ask, and let the business know what your preferences are!
Summary (cont.)

• I really like things that work, and **work well** - like **CICS** and the **mainframe** (ok, **Linux**, too ;-).

• IT work should be productive, enjoyable, and **fun**. Ok, it doesn't always work out that way! My experience is that working with CICS and the mainframe, from a Linux workstation, is productive, enjoyable, and yes, even fun (most of the time ;-)).

• The **future** looks very **bright** for **CICS** and the **mainframe**! (Sunglasses are optional ;-)
Summary (cont.)

• **Thanks!** Have a great time for the remainder of the conference, and have a safe trip home.

• **Questions?** Comments? *Random thoughts*?
Appendix and Additional Information

- IBM CICS (Customer Information Control System):
  http://www.ibm.com/cics

- IBM CICS Transaction Server for z/OS V4.1 Announcement Letter:
  http://www.ibm.com/common/ssi/cgi-bin/ssialias?
  infotype=an&subtype=ca&supplier=897&letternum=ENUS209-135

- IBM CICS Portfolio Brochure (G224-7571-00):
Appendix (cont.)

- IBM Family - News:

- IBM CICS SupportPacs:

- IBM Mainframe Servers - System z:
Appendix (cont.)

- IBM Academic Initiative:  
  http://www.ibm.com/university/

- IBM Redbooks:  
  http://www.redbooks.ibm.com/

- Shibboleth:  
  http://shibboleth.internet2.edu/
Appendix (cont.)

• SHARE ("It's not an acronym, it's what we do.")
  http://www.share.org/

• SHARE CICS Project:
  http://www.share.org/cics
Appendix (cont.)

• The University of Florida (UF):
  http://www.ufl.edu/

• UF Computing & Networking Services (CNS):
  http://www.cns.ufl.edu/

• CICS at UF:
  http://cics.ufl.edu/

• EAGLE at UF:
  http://eagle.ufl.edu/
Abbreviations

- **AJAX**: Asynchronous Javascript and XML
- **CICS**: Customer Information Control System
- **CIS**: Computer and Information Sciences
- **CISE**: Computer and Information Science and Engineering
- **CMF**: CICS Measurement Facility (via SMF)
- **CNS**: Computing & Networking Services (formerly NERDC)
- **COD**: Computing On Demand
- **CP**: Central Processor - see CPU
- **CPU**: Central Processing Unit
Abbreviations (cont.)

- **DFSMS**: Data Facility Storage Management Subsystem
- **EAGLE**: UF Enhanced Application Generation Language for the Enterprise
- **EGL**: IBM Enterprise Generation Language
- **ESM**: External Security Manager
- **FACTS**: Florida Academic Counseling and Tracking for Students
- **GA**: General Availability
- **Heritage**: See Legacy
Abbreviations (cont.)

- IBM: International Business Machines, Inc.
- I/O: Input/Output
- ICF: Integrated Coupling Facility
- ICSF: Integrated Cryptographic Service Facility
- IFL: Integrated Facility for Linux
- IT: Information Technology
- ISIS: UF Integrated Student Information System
- Java: Programming language and computing platform released by Sun Microsystems in 1995
Abbreviations (cont.)

- LE: Language Environment
- Legacy: It Works!
- LPAR: Logical Partition
- MTBF: Mean Time Between Failures
- MVS: Multiple Virtual Storage
Abbreviations (cont.)

• NERDC: Northeast Regional Data Center (now CNS)
• OS: Operating System
• OTE: Open Transaction Environment
• PDF: Portable Document Format
• PHP: Hypertext Preprocessor
• PR/SM: IBM Processor Resource/Systems Manager
• RMF: Resource Monitoring Facility
• RLS: Record Level Sharing
Abbreviations (cont.)

- **SAML**: Security Assertion Markup Language
- **SCRT**: Sub-Capacity Reporting Tool
- **SMF**: System Monitoring Facility
- **SNA**: Systems Network Architecture
- **SOA**: Service Oriented Architecture
- **SOAP**: Simple Object Access Protocol (a component of web services)
Abbreviations (cont.)

- TCO: Total Cost of Ownership
- TCU: Total Cost per User
- TS: Transaction Server
- UF: University of Florida
- UNF: University of North Florida
- VSAM: Virtual Storage Access Method
- VTAM: Virtual Telecommunications Access Method
- VM: Virtual Machine
- WLM: Workload Manager
Abbreviations (cont.)

- **z/OS**: The "zero downtime" and "a to z" Operating System
- **zSeries**: The "zero downtime" and "a to z" Enterprise Servers
- **zAAP**: zSeries Application Assist Processor (for Java)
- **zIIP**: zSeries Integrated Information Processor (for DB2)
- **zNALC**: System z New Application License Charges
Presentation Information

• The Slackware Linux Project:  
  http://www.slackware.com/

• OpenOffice.org 3.1.0 "Impress":  
  http://www.openoffice.org/  
  (File -> Export as PDF)  
  (Used SHARE supplied PowerPoint template.)

• Samsung N110-12PBK:  