CANADIAN TANDEM USERS GROUP

OSS in the Real World

CTUG Fall 2004 Technical Session

Randall S. Becker
CEO, Nexbridge Inc.

Agenda

- UNIX Introduction
  - Origins
  - File System
- Best Practice Development with OSS
  - OSS vs. NSK, what can I do where
  - Do I have to give up Pathway?
  - Building code under OSS
  - Where's my Coffee (Java)
  - So you want to use ETK
  - Sewing with Threads
  - Fun with MX
- OSS Q&A

CANADIAN TANDEM USERS GROUP

UNIX Introduction

UNIX Introduction

- UNIX is an offshoot of MULTICS
- Originally designed as a single user operating system
- Made popular by university students
- Variants:

XENIX	Linux	System V
SCO UNIX	Solaris	AIX TRU64
BSD UNIX	OSS	and many more

Key Features

- Hierarchical file system
- Consistent command interface (Shell)
- Files are unstructured streams of bytes
- Pipes
- Sockets (origins BSD and System V)
- Common API

Key Misses

- Unstructured File System Only
- Security model is very simplistic
- User interface developed by experts for experts
- Lowest Common Denominator API
- Network security virtually unknown
- Assumed many users many systems

Key Opportunities

- Systems management is painful
- Easy access means anyone can claim UNIX experience
- Scaling applications
- Cluster management is complex
- Network management is simplistic

What is OSS, Really?

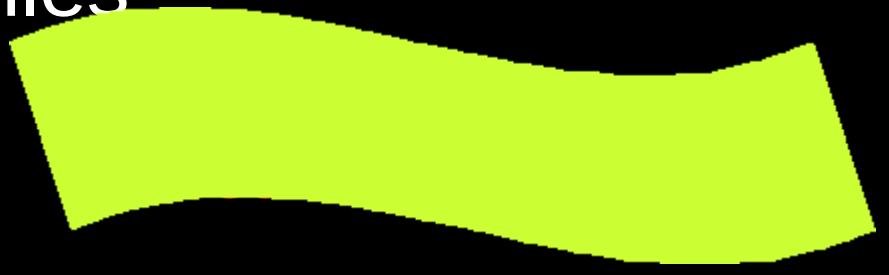
- Originally just a layer on top of Guardian
- OSS requires NonStop Kernel
- File system is built on DP2
- Security is independent of Guardian security, mostly
- Really does implement most of UNIX
- Scalability requires NonStop knowledge

OSS is UNIX

- Regardless of what the word on the street is, OSS looks like UNIX
- UNIX APIs are well supported
- Most applications port cleanly with just a recompile
- However…

OSS is not UNIX

- NonStop is not an SMP
- Use NonStop fundamentals like Pathway to manage scalability
- NSK rules still apply
- File handles can be shared, but not always for /G or /E files



OSS vs. NSK

- Differences in the File Systems
- Differences in API
- Sockets and Threading are not what we remember about NonStop
- Shell and TACL can both be used
- Pathway still manages messaging and scaling
- Security is very different in OSS

Files under OSS

- Hierarchical file system
- Long file names
- Security organized by user, group, other
- Security attributes read, write, execute
- Odd-unstructured files
- Extent limits don’t really apply

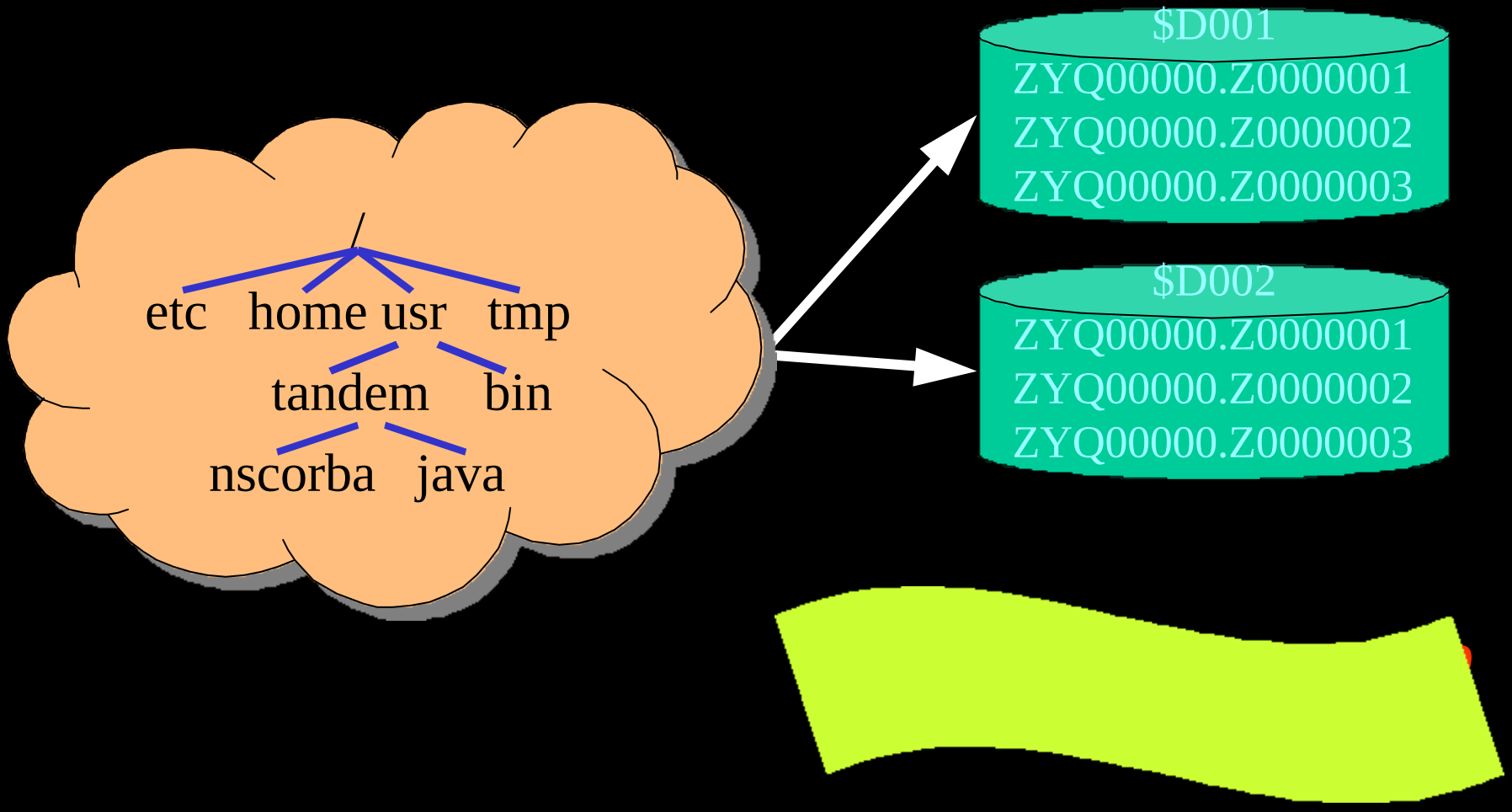
Getting at Guardian Files

- `/G/volume/subvol/filename` allows UNIX APIs and Shells to read Guardian files
- `/E/system/volume/subvol/filename` provides EXPAND access
- Only code 180 files can be written to `/G` and /E directories
- Sub-volumes only appear in directory listings when files are created in them

Do UNIX Names Matter?

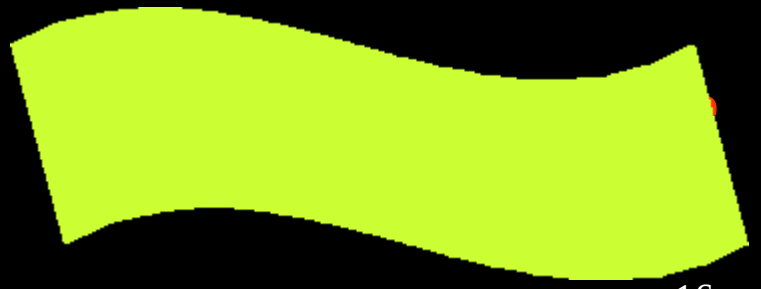
- Only to standard UNIX tools and ported programs
- NSK procedures are still available
- NSK DEFINE support is available
- fopen() knows about UNIX, NSK file and DEFINE names
  - Support both, if you can

Where do OSS files come from?



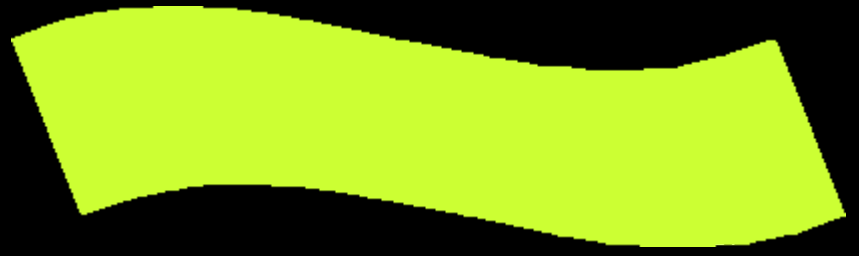
What's in a Name?

- OSS Name Server
  - Maps OSS to NSK file names
- Domain Name Service
  - Maps host names and IP addresses
- CORBA Name Service
  - Stores references to server classes and processes



Things to Remember

- Always use mirrored volumes for OSS catalogs
- Configure as many disks as you can for OSS file systems
- Minimum useful OSS version is G06.22 with all SPRs



CANADIAN TANDEM USERS GROUP

Pathway and OSS

Pathway

- Pathway manages both types of processes
  - Process Type enables different personalities
  - Nothing is lost
  - OSS processes behave just like NSK processes under Pathway
  - Proper $RECEIVE handling is required

Pathway Process Types

- Two types of processes:
  - `set server processtype OSS`
    - Sets up a server running by OSS rules
    - Allows only OSS server settings
      - stdin, stdout, arglist, ENV
  - `set server processtype guardian`
    - Sets up a server running by NSK rules
    - Allows only NSK server settings
      - IN, OUT, STARTUP, PARAM

Persistence Manager

- This is a new way of managing processes under OSS
- Not tried and true, but Kernel Services use it
- Used for keeping processes running
- Does not manage scaling
- Does not manage communication
- Watch this space!

CANADIAN TANDEM USERS GROUP

Building Code Under OSS

Building Code under OSS

- Same compilers!
  - c89 does C and C++
  - It's time to move off of TAL and COBOL
  - There's always Java
- Make
- GNU tools have been ported

C and C++

- The c89 compiler handles both C and C++ using file extensions:
  - .c files are handled as C modules
  - .cpp files are handled as C++ files
  - .o files are object files
- The linker is automatically invoked by c89

Standard Header Files

- C and C++ programs support standard UNIX library header files
- Standard UNIX headers have .h suffixes:

  stdio.h, unistd.h, string.h
- C++ template libraries do not have suffixes:
  - string, map

Standard Header Files

- Guardian header files ending with ‘h’ can be specified as .h
- Guardian procedure calls are in cextdecs in $system.system

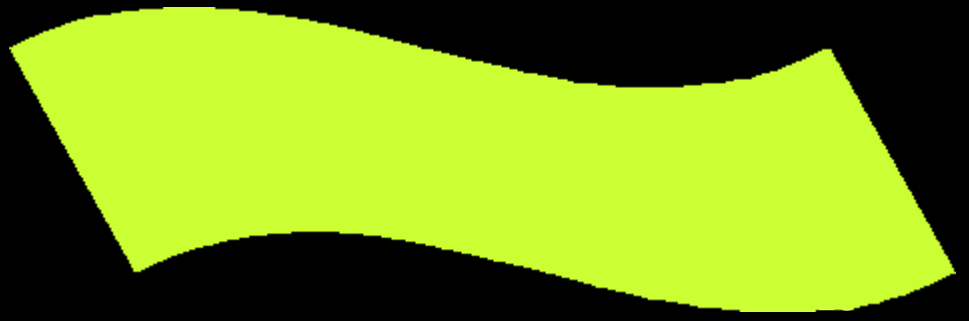
C++ Versions

- There have been 3 revisions to the C++ language standard to date
  - Version1 is essentially obsolete
  - Version2 is compatible with SQL/MX 1.8.5 and CORBA B13-C12
  - Version3 can only be used in isolation for now but is compatible with SQL/MX 2.0 and CORBA C20

CANADIAN TANDEM USERS GROUP

C++ Version1 Compatibility

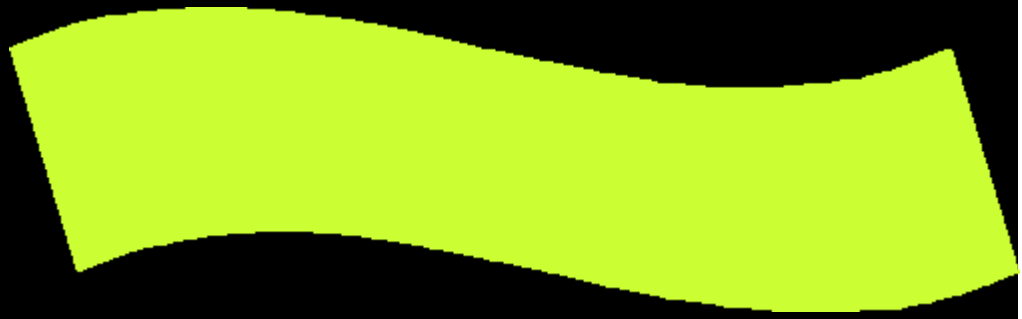
- Can be upgraded to Version2 without change
- Can be upgraded to Version3 without change



CANADIAN TANDEM USERS GROUP

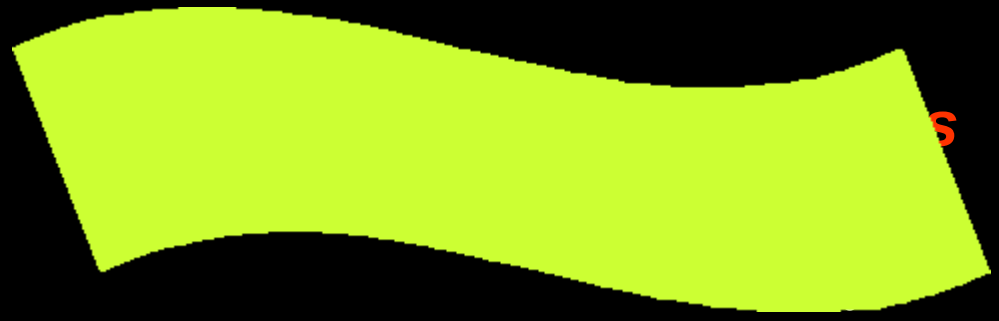
C++ Version2 Compatibility

- Cannot be downgraded to Version1 without substantial change
- Not source compatible with Version3
  - Requires small changes to use of Standard Template Libraries



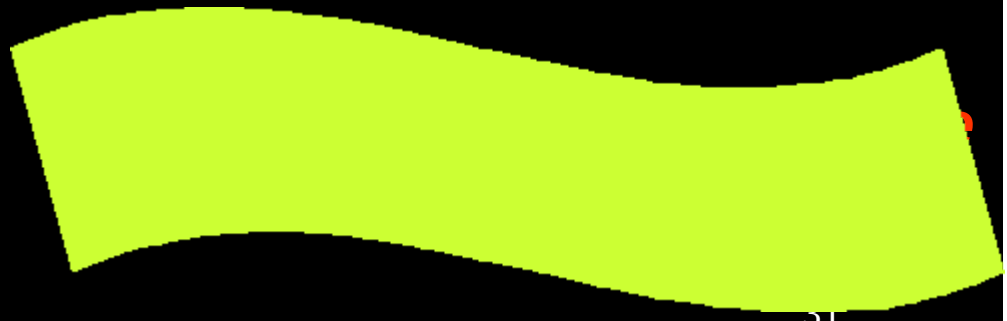
C++ Version3 Compatibility

- Cannot be downgraded to Version1 without change
- Cannot be downgraded to Version2 without change
- Most compatible with other platforms



Make

- Make is a standard UNIX tool for building applications
- Easy to set up compiles
- Expensive to maintain as your application evolves



Revision Management

- Consider an industrial strength SCM solution of some kind
- Don’t minimize the cost of “Freeware”
- Ensure you are corruption-safe and have backups
- Ensure your procedures for building releases are solid

Backup Limitations

- OSS Catalog cannot be incrementally restored
  - Once a file is gone, it's gone
- Mixing Volume and Incremental backups doesn't always work for OSS
- PAK/UNPAK need to be part of your procedures

Backups Practices

- Take frequent backups
- Always mirror your OSS Catalog disks
- Backup development drives at least once a day
- Compress unused files to cut down backup time

CANADIAN TANDEM USERS GROUP

Where’s my Coffee (Java)

Java and NonStop

- Java is here
- Some ZLE infrastructure components have substantial Java elements
- Java resources are widely available
- Java resources need to learn NonStop

The Java Runtime

- The Java run-time is an OSS program that executes Java code
- Object file is called `java`
- Java can be run through shell scripts or as a Pathway server class
- Environment variables (ENV) in Pathway and Shell no longer available for controlling the Java applications
- Command line arguments are normally used to run Java programs

Java Versions

- Key advantage to NonStop Java is the ability to run multiple versions
- Keep copies of java objects for migration and fallback purposes
- Don’t use the standard location since DSM/SCM will replace it

Getting to NSK from Java

- Unless there is an HP-supplied class for Java to get at desired resources, use JNI
- JNI is the Java Native Interface
- Good samples exist in the java samples directory /usr/tandem/java/samples
- JNI modules are linked with the java binary

Getting from Java to servers

- Structured messaging is difficult due to lack of data control
- CORBA + Java provide a portable message structure that scales well and is easy to use on NonStop
- CORBA automatically handles local and off-platform message translation
- CORBA + Java is a naturally open pair

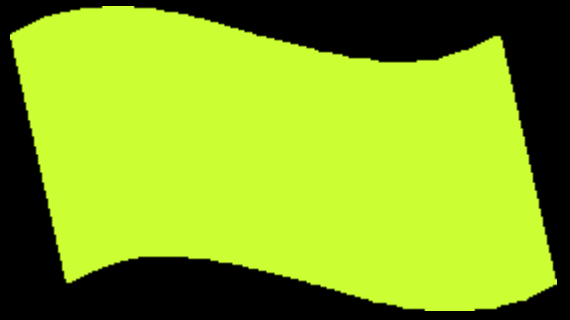
What happened to Ddl2Java?

- Ddl2Java builds classes that format messages for use on NonStop only
- Off-platform client access is not supported, unlike CORBA
- Requires knowledge of proprietary Enscribe DDL
- Only solid alternative to existing legacy servers

CANADIAN TANDEM USERS GROUP

Too Much Caffeine?

- Java allows rapid program development
- Memory model is highly dynamic
- Garbage collection is CPU intensive
- Process start-up is expensive
- Monitor CPU resource usage
- Load-test your application as early as possible
- Messaging is not built-in

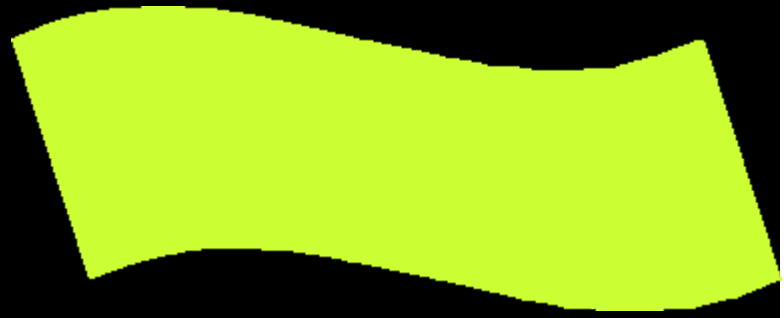


Deploying Java Solutions

- Dedicate resources to Java/BEA WebLogic
- Java is Thread-intensive
- Migrate from Ddl2Java to new offerings including CORBA
- Continuously test off-platform
- Deploy different JDBC SQL drivers

Messaging Options

- Ddl2Java provides access to legacy Pathway servers
- SOAP provides web access into the NonStop
- CORBA provides efficient on-box and LAN messaging and transaction services



CANADIAN TANDEM USERS GROUP

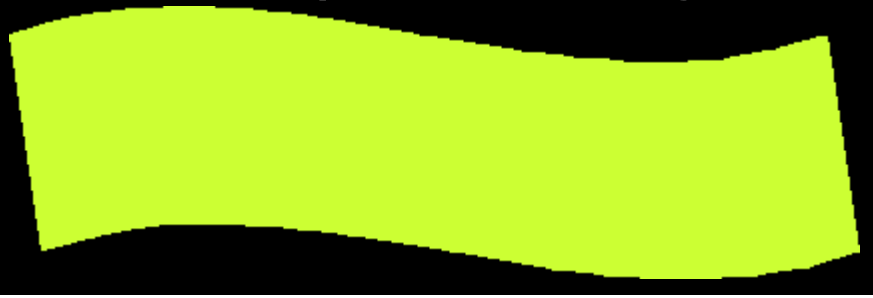
So You Want to Use ETK

The Enterprise Toolkit

- Built on MS Visual Studio .NET 2003
- Allows developers to build code on desktops
- Uses cross-compilers run on workstations to build object code for C, C++, pTAL, COBOL, etc.
- Supports SQL/MP and SQL/MX compiles

What ETK Doesn't Do

- Does not generate code that executes on workstations
  - Use Windows projects with the same source files as your NonStop projects
- Does not support Java
  - Use a third-party development environment and then deploy to the NonStop for testing



What the ETK Does Do

- Uses workstation horsepower to compile
- Developers are much more productive
- OSS disks usage is substantially reduced
- Allows solutions to be built to include NSK and non-NSK projects
- Allows builds for different operation system versions

ETK Slip-ups

- Building for the wrong operating system can result in invalid objects
- Make sure SRL and header files are compatible with the target system
- Functionality to build of all of your solutions is not supplied
- Functionality to manage and convert groups of solutions is not supplied

Best ETK Practices

- Make sure you have a way of compiling everything
- Have separate Windows and NSK variants of projects to test portable code on the workstation
- Manage the build versions very carefully
- Investment in infrastructure pays back in saved developer time

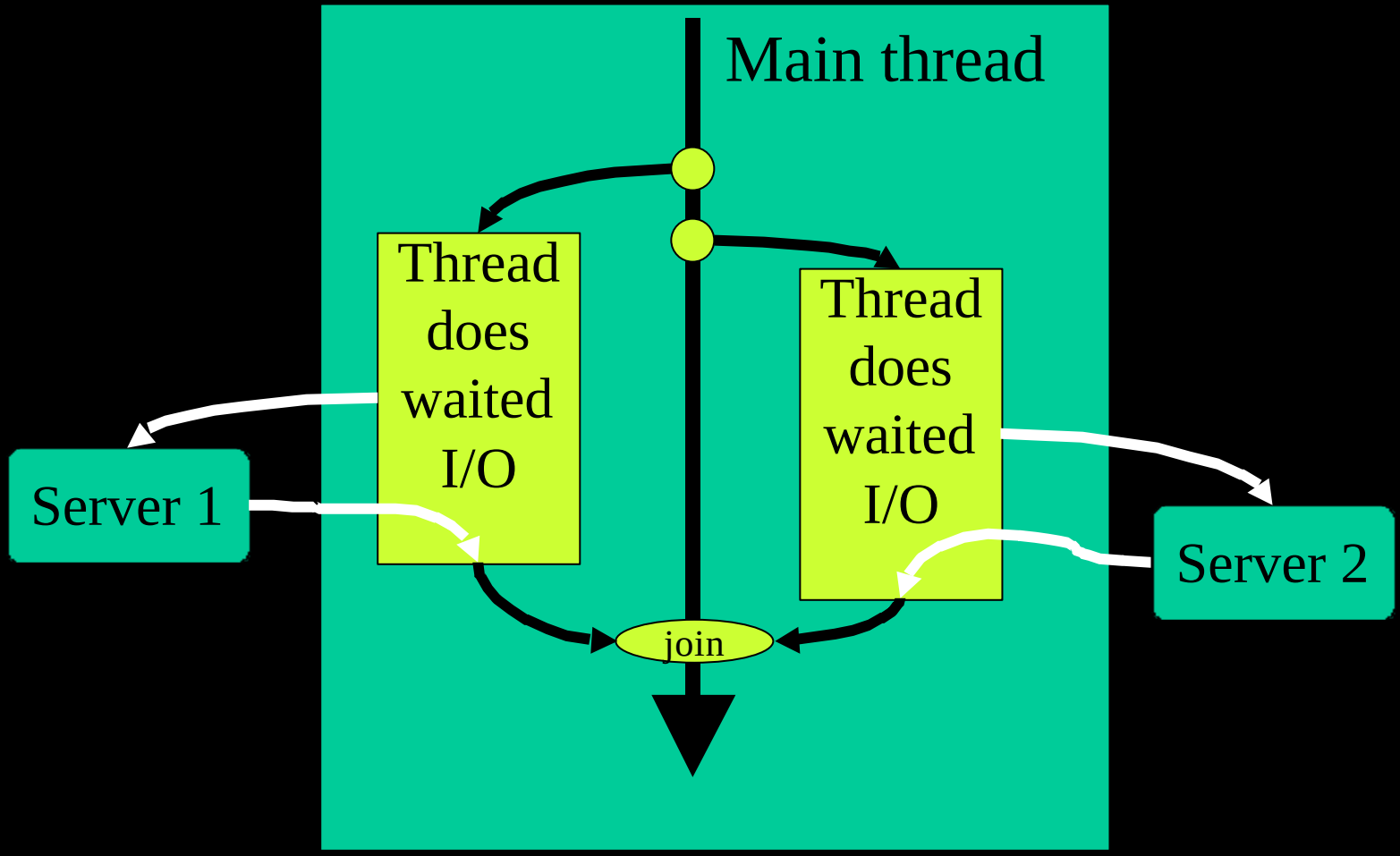
CANADIAN TANDEM USERS GROUP

Sewing with Threads

Threads are Good

- Threads allow developers to implement simple asynchronous operations
- Tags are a thing of the past
- Pathsend supports threads
- Guardian File IO now supports threads
- Socket IO supports threads automatically
- CORBA is threaded

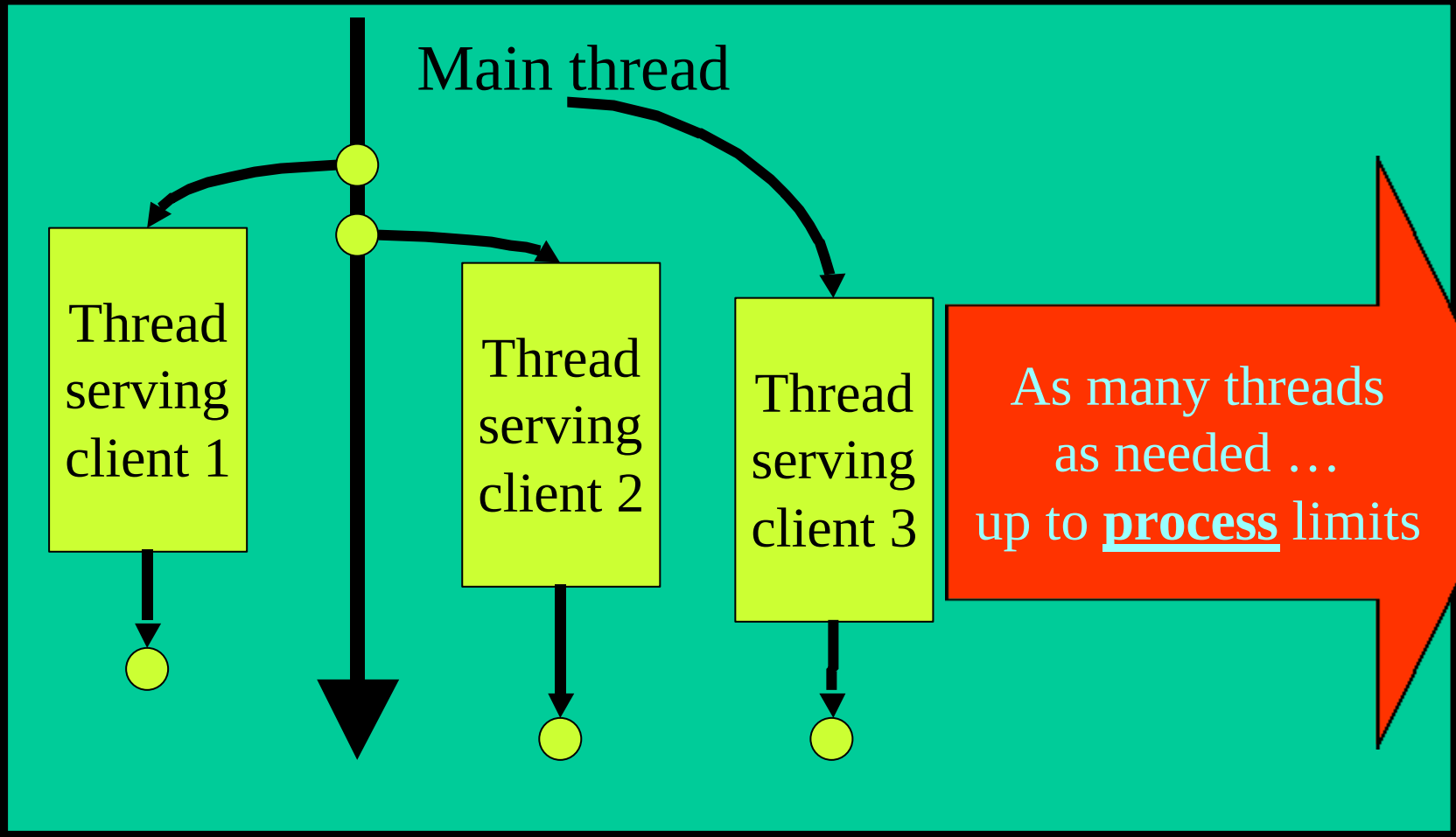
Threaded Client



Threads can be Painful

- Threads use FILE_COMPLETION_ instead of AWAITIOX
- The pThread library implements threads uses polling and can increase CPU load with no apparent work
- Sound thread management requires higher-than-average developers

Threaded Server



Thread Summary

- Multi-threaded clients are useful to permit parallel access to servers
- Multi-threaded servers are useful when doing an initial port from another platform
- Multi-threaded servers DO NOT SCALE well!

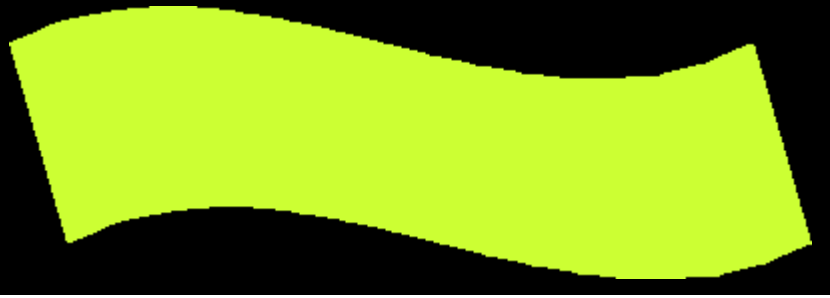
CANADIAN TANDEM USERS GROUP

Fun with SQL/MX

Version 1.8.5 SPR ABI

SQL/MX Ties to OSS

- OSS is required for SQL/MX
- SQL/MP can also be used from OSS
- Module files are stored in OSS directories instead of being embedded
- SQL/MP tables use Guardian and Safeguard security
- Modules use OSS security



SQL/MX: The Good

- Queries are fast
- Advanced executor builds more predicable plans
- ANSI compliance:
  - SQL syntax
  - Record locking
  - Table names
- Objects can reference tables in multiple catalogs

SQL/MX: The Bad

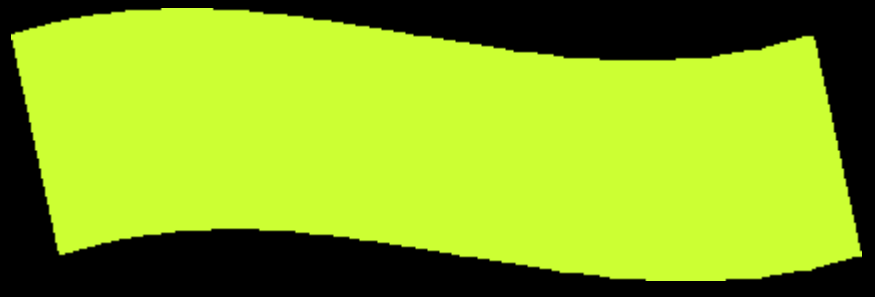
- Inserts and Updates are slow
- Diagnostics are thin compared to MP
  - Difficulty in resolving security-related problems
  - File errors are not visible to programs
- Can’t tie objects to table usage
- Aggressive similarity checking
  - Ongoing changes in this area

SQL/MX: The Ugly

- Module Management is awkward
- Table naming is confusing:
  - ANSI Names
  - NSK Names
  - DEFINE Names
- Prototypes

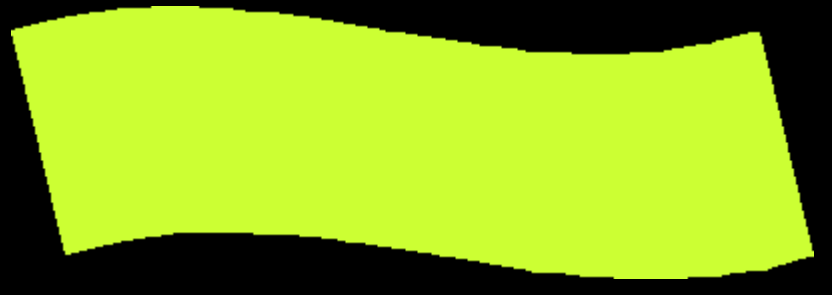
SQL/MX: Module Management

- Global modules are stored in `/usr/tandem/sqlmx/USERMODULES`
- Local modules are stored in the object's directory
- ETK has settings for where to put modules
- Modules sometimes contain EXPAND references



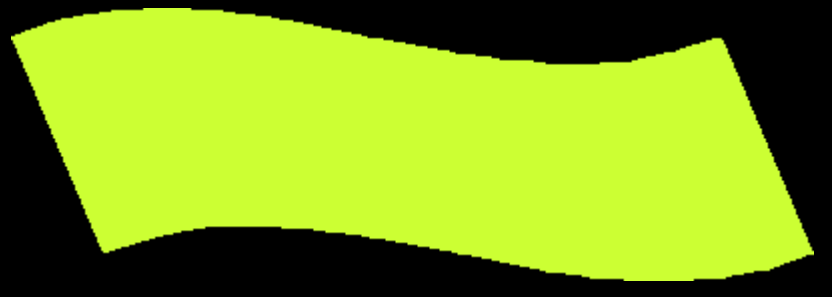
SQL/MX: Table Naming

- ANSI names have three levels:
  - Catalog
  - Schema
  - Table
- ANSI names are best for Java since off-platform drivers only know about ANSI
- Catalog and Schema are arbitrary values



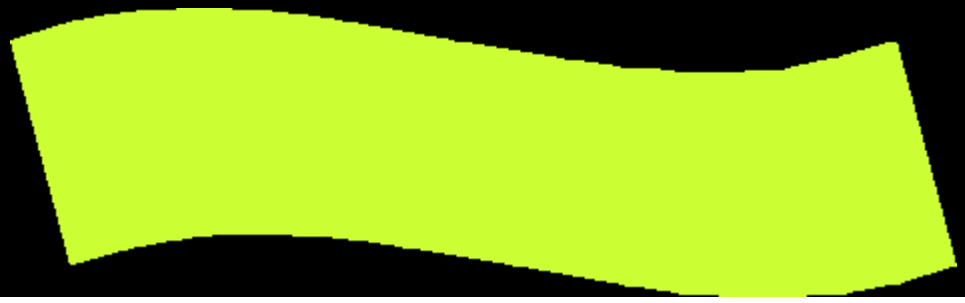
SQL/MX: Table Names

- NSK File Names use standard format:
  - System
  - Volume
  - Sub-volume
  - File
- This form is only useful for Dynamic SQL:
  - The file name in the Module cannot be changed at run-time



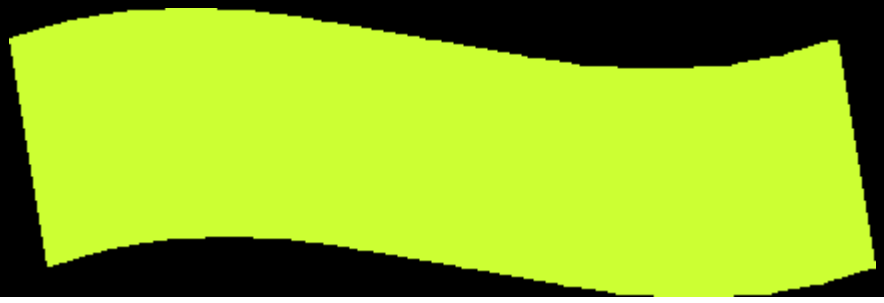
SQL/MX: Table Names

- DEFINE names use standard format
- Names can be redefined:
  - Specify Shell defines and use the mxcmp compiler
  - Specify DEFINES at run-time



SQL/MX: Prototypes

- Prototypes allow developers to:
  - Use a model of the table for compiles
  - Change the name of a table at run-time through a host variable
- For similarity checks, prototypes need the original model table
- Security rules often preclude the use of prototypes



Where to Go For More

- NonStop Technical Library
  - Guardian Programming Manual
  - OSS Programming Manual
  - Open Systems Services Management and Operations Guide