

UNIX Systems Services in Today's Mainframe Software

Kevin Wick, CA Technologies

Session 10979

Wednesday, March 14, 2012, 9:30 AM



Abstract

- Been using USS for awhile, but still not comfortable with the environment? Do you know enough to be dangerous, and usually are? More and more vendors are starting to use UNIX filesystems and utilities as the heart of their new software, and we need to get you ready to manage it! In this session, the speaker will give an overview of the base UNIX System Services environment, talk about what you need to prepare so your users can use UNIX, and describe how you can manage the files that your vendors are using. We will cover profile setup, file system management, and even get into a high level overview of common web containers, like Apache Tomcat. It's a whole new world out there!

Legal Notice

- © Copyright CA 2012. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. No unauthorized use, copying or distribution permitted.
- THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised of the possibility of such damages.
- Certain information in this presentation may outline CA’s general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA’s sole discretion.
- Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available (i) for sale to new licensees of such product; and (ii) in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis.

What's in a name?

- Official IBM Naming Progression
 - OpenEdition, UNIX Services
 - OS/390 UNIX System Services, OS/390 UNIX
 - z/OS UNIX System Services, z/OS UNIX
- Common but “Unofficial” Names
 - USS: UNIX System Services
 - z/UNIX?

So, What am I Dealing With Here?

- Shell Interface
 - OMVS, ISHELL (ISH)
 - Two shells, “z/OS” and tcsh
 - Scripting Languages (batch and interactive)
- Application Programming Interface (API)
 - Execute programs in batch
 - Request services (from ASM, TSO, REXX)
- Starting to see products (and programmers!) who want to take advantage of all these features at once

What Do My Users Need?

- Notorious “OMVS Segment”
 - UID
 - Logon Shell
 - Home Directory
- Absolutely required for USS use!
 - If you don’t, strange errors will occur!
- Good idea to assign a default group as well
 - If not, z/OS will use the installation default
- Let’s examine each in detail...

UID

- Must:
 - Be numeric, between 0 and 2147483647
- Recommended:
 - Less than 2097151 (because of tar/pax extraction issues)
 - Unique to each user (ensure file security)
 - Not zero! Give BPX.SUPERUSER instead.
 - Even better, implement sudo! More logging and granularity.
- Security packages have the ability to assign the next free UID and prevent shared UIDs... this is **HIGHLY RECOMMENDED!**
- Almost all of this applies to the GID as well

Logon Shell

- Must:
 - Exist!
 - Be fully qualified
 - Case sensitive
- Called when the user issues a request for a shell
 - Also when using rlogin, otelnetd, su, or newgrp
- Rare to see it set to anything other than `‘/bin/sh’`

Home Directory

- Must:
 - Exist!
 - Be fully qualified
 - Case sensitive
- Recommended:
 - Unique to each user (filesystem too!)
 - Not the root filesystem (/)
 - Not read-only
- This is where the shell points whenever a user requests shell services (interactive or batch). If it's not there, we've got a problem.

Seems Easy, Right?

- Can't I just make up a bunch of paths and numbers?
 - Sure, for the first 40 users or so...
- Have a policy for assigning UIDs (or use auto assign)
- Make individual filesystems for each user
 - Take advantage of automount (and allocuser option)
 - Use standard naming convention (just like datasets)
 - /u/smiths01, /u/doej01, /u/smiths02
 - Careful, must be unique here...
- Don't forget to educate the security team. In most cases, they will be defining new users, not you.

What About File Systems?

- Some common questions:
 - What are all these directories?
 - Where are you going to put stuff?
 - Where should I put stuff?
 - How do I maintain all this?
- Again, we'll discuss each one individually...

Don't Touch Me!

- /bin
 - UNIX executables, shell commands
 - Functions somewhat like LNKLST
- /dev
 - Special character files for shells and c89.
 - Populated during IPL, updated automatically
- /lib
 - C runtime libraries, TCP/IP symbolics
- /var
 - Dynamic data used internally by z/OS
 - Files created during execution/customization

We're Here for You

- /tmp
 - Temporary data
 - Programs/applications use this, you can too!
- /etc
 - Configuration files for all sorts of stuff
 - This is a grey area; IBM puts things here too
- /samples
 - “Un-customized” files (that will live in /etc)
- /u
 - User home directories
 - /u/users is redundant!

Other UNIX Standard Directories

- /opt
 - Software or other “add-on” packages
- /usr
 - Shared user data
 - Usually read-only
 - Not home directories!
 - Might see /usr/lpp
 - LPP: Licensed Program Product
 - Used by TCPIP, BookManager...
- Many others!
 - http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

What Do Vendors Do?

- ?
- IBM might use /opt, /etc, or recommend /usr/lpp
- /<VendorName>/<ProductName>
 - Keeps all related things together
- Almost everyone uses /tmp to some degree
- There is almost always an option to customize
 - Don't feel forced to take the default
 - However, default sometimes does make the install easier
- Careful changing the directory after install!

Possible Vendor Example

- `/etc/<productname>/version`
 - Product Code
- `/usr/lpp/<productname>/version`
 - Shared Product Modules
- `/etc/<productname>`
 - User Configuration Files
- `/var/<productname>/logs`
 - Log Files
- `/var/<productname>/data`
 - Product Database
- `/tmp`
 - Temporary Files

Recommendations for Directories

- Use /u for home directories
 - Don't fill it with other "stuff"
- Get in the habit of /tmp for temporary files
 - Remember to clean up if not TFS
- Common files go in common places!
 - Put Java in /usr/lpp as recommended
 - Don't fill it with other "stuff"
- Feel free to make your own directories
 - Make a standard and enforce it!
 - Apply DSN qualifiers to promote consistency
 - /sys2/<ProductName> for vendor products

Other File System Recommendations

- Give every user their own (small) file system for their home directory
 - Prevents out of space conditions due to other users
 - Encourage creation/addition of more filesystems as necessary
 - Means keeping track, mount permissions, etc...
- Remember that filesystems should be mounted to empty directories!
 - Don't "cover up" files
 - This is a very, very common issue

More File System Recommendations

- Use multiple BPXPRMxx members for mounts
 - Don't forget order of mount points
 - Remember to document!
- New filesystems as logical groupings
 - Again, avoid out of space errors from others
 - Easier to manage and control (unmount and move)
 - More organized, less confusing to new people
- Always new mount points on root, never files
 - Specifically recommended by IBM
 - Easier to upgrade z/OS
 - Usually read-only

Troubleshooting Problems

- Display contents of everything
 - `D OMVS,FILE,O`
- Display contents of specific directories and filesystems
 - `ls -la /directory/name`
 - `df -p NAME.OF.ZFS`
- Display Mount Failures
 - `D OMVS,MF=A`
- If you have to, automate these commands to catch intermittent problems

Recommendations for Java

- Always have at least 3 versions installed and mounted
 - Current (v6.0.1), previous (v6.0.0), bleeding edge (v7)
 - Both 31 and 64 bit
- Use the directories structure IBM provides
 - `/usr/lpp/java/J<version>` and `/usr/lpp/java/J<version>_64`
- Give users access to Java by default
 - Set JAVA_HOME, CLASSPATH, LIBPATH in /etc/profile
- Install optional elements (JZOS)
- Update on Service Refreshes

What's Wrong With This Picture?

- /sys/java31bt/v5r0m0/usr/lpp/java/J5.0
- /sys/java31bt/v6r0m0/usr/lpp/java/J6.0
- /sys/java31bt/v6r0m1/usr/lpp/java/J6.0.1
- /sys/java31bt/v7r0m0/usr/lpp/java/J7
- /sys/java64bt/v1r4m2/usr/lpp/java/J1.4_64
- /sys/java64bt/v6r0m0/usr/lpp/java/J6.0_64
- /sys/java64bt/v6r0m1/usr/lpp/java/J6.0.1_64
- /sys/java64bt/v7r0m0/usr/lpp/java/J7.0_64

Much Better!

- `/usr/lpp/java/J6.0`
 - `/usr/lpp/java/J6.0_64`
 - `/usr/lpp/java/J6.0.1`
 - `/usr/lpp/java/J6.0.1_64`
 - `/usr/lpp/java/J7`
 - `/usr/lpp/java/J7.0_64`
-
- IBM Standard, as delivered, in documentation
 - Clean, easy to find, easy to understand
 - No wasted/empty directories
 - Takes a bit more work, but your users will thank you.

The Actual Work!

- Traditional ways to “work” with USS
 - Interactive (OMVS, ISHELL)
 - Batch (Scripts and programs with BPXBATCCH)
- Other ways
 - API (ASM, TSO, REXX, etc...)
 - JZOS Batch Launcher
- Use of JZOS is popping up more and more often
 - Easy way to get Java running in a started task
 - Provides additional Java methods to access z/OS services

What is JZOS?

- A load module! There's a sample PROC too.
- Basically, JZOS is a program that allows Java to run as a started task
 - DD statement support
 - Return code handling
 - Console support (WTOs, Start/Stop, etc...)
 - Access to system variables
- How do I get it?
 - Free offering from IBM
 - Option on z/OS Java or can be downloaded separately

JZOS In Action

```
/u/wick/java> cat HelloWorld.java
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
/u/wick/java> javac HelloWorld.java
/u/wick/java> ls HelloWorld.*
HelloWorld.class  HelloWorld.java
/u/wick/java>
```

```
000008 //STEP01 EXEC PGM=JVMLDM60,REGION=0M,PARM='HelloWorld +D'
000009 //STEPLIB DD DISP=SHR,DSN=SYS2.JZOS.LOADLIB
000010 //SYSPRINT DD SYSOUT=*
000011 //SYSOUT DD SYSOUT=*
000012 //STDOUT DD SYSOUT=*
000013 //STDERR DD SYSOUT=*
000014 //STDENV DD *
000015 APP_HOME=/u/wick/java
000016 export JAVA_HOME=/usr/lpp/java/J6.0
000017
000018 export PATH=/bin:${JAVA_HOME}/bin
000019 LIBPATH=/lib:/usr/lib:${JAVA_HOME}/bin
000020 LIBPATH="${LIBPATH}:"${JAVA_HOME}/lib/s390
000021 LIBPATH="${LIBPATH}:"${JAVA_HOME}/lib/s390/j9vm
000022 LIBPATH="${LIBPATH}:"${JAVA_HOME}/bin/classic
000023 export LIBPATH="${LIBPATH}:"
000024 export CLASSPATH=$APP_HOME:${JAVA_HOME}/lib:${JAVA_HOME}/lib/ext
000025
000026 IJO="-Xms16m -Xmx128m"
000027 export IBM_JAVA_OPTIONS="$IJO "
000028 /*
```

JZOS Output

```
...+....10.|.+....20...+.  
Hello World!
```

```
...+....10...+....20...+....30...+....40...+....50...+....60...+....70...+....80...+....90...+....100...+....1|  
JVMJZBL1001N JZOS batch Launcher Version: 2.3.0 2009-10-08  
JVMJZBL1002N Copyright (C) IBM Corp. 2005. All rights reserved.  
java version "1.6.0"  
Java(TM) SE Runtime Environment (build pmz3160sr9-20101125_01 (SR9))  
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 z/OS s390-31 jvmmz3160sr9-20101124_69295 (JIT enabled, AOT enabled)  
J9VM - 20101124_069295  
JIT - r9_20101028_17488ifx2  
GC - 20101027_AA)  
JVMJZBL1023N Invoking HelloWorld.main()...  
JVMJZBL1024N HelloWorld.main() completed.  
JVMJZBL1021N JZOS batch launcher completed, return code=0
```

- Wonderful, more Java. Why do I care about JZOS?

Starting Point for Serious Java!

- Provides a MVS environment for web containers
 - Open-source projects: Jetty, Apache Tomcat, JBoss
- Popular among vendors who don't have time/resources to write their own web containers
- With a little work, you can run these too!
 - Dovetailed Technologies' Quickstart for Tomcat
- We'll use this as a sample for the rest of the discussion
 - Vendor web containers are very similar, with more customization

Quickstart for Tomcat

- Apache Tomcat, optimized for z/OS

```

000001 <?xml version='1.0' encoding='IBM-1047'?>
000002 <!--
000003   This is an XML fragment containing entity declarations (variables)
000004   that can be used in the Tomcat SERVXML
000005   -->
000006
000007 <!ENTITY httpPort "9376">          <!-- the Tomcat HTTP port          -->
000008 <!ENTITY sslPort  "8443">          <!-- if SERVXML configured for SSL/TLS -->
000009 <!ENTITY hostname "localhost">    <!-- normally dont need to change -->
000010
000011 <!-- The following used for DB2 type-2 JDBC only (see SERVXML)          -->
000012 <!-- db2Loc specifies the DB2 database aka "location" name.              -->
000013 <!-- Note: DB2_SSID must also be set in TCENV script for type-2          -->
000014 <!ENTITY db2Loc   "LOC1">
000015

```

```

Aug 10, 2011 10:55:42 AM org.apache.coyote.http11.Http11Protocol init      (see SERVXML)      -->
INFO: Initializing Coyote HTTP/1.1 on http-8080
Aug 10, 2011 10:55:42 AM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 877 ms
Aug 10, 2011 10:55:42 AM org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
Aug 10, 2011 10:55:42 AM org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.18
Aug 10, 2011 10:55:43 AM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Aug 10, 2011 10:55:43 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 885 ms

```

The Tomcat File System

```
drwxrwxrwx  8 WICKE01 OMVSDFG      544 Aug 10 10:48 .
drwxrwxrwx  9 WICKE01 FRAMEWKG    8192 Aug 10 10:49 ..
-rw-r----- 1 WICKE01 OMVSDFG    37950 Dec  4 2009 LICENSE
-rw-r----- 1 WICKE01 OMVSDFG      556 Dec  4 2009 NOTICE
-rw-r----- 1 WICKE01 OMVSDFG    7318 Dec  4 2009 RELEASE-NOTES
drwxr-xr-x  2 WICKE01 OMVSDFG      704 Dec  4 2009 bin
drwxr-xr-x  2 WICKE01 OMVSDFG      576 Dec  4 2009 conf
drwxr-xr-x  2 WICKE01 OMVSDFG      992 Dec  4 2009 lib
drwxr-xr-x  2 WICKE01 OMVSDFG      256 Dec  4 2009 temp
drwxr-xr-x  5 WICKE01 OMVSDFG      352 Dec  4 2009 webapps
drwxr-xr-x  3 WICKE01 OMVSDFG      288 Aug 10 10:55 work
```

- Looks familiar?
- Very similar to root
- Another set of directories to manage and understand
- Need to be aware of what they contain and how they are used and maintained
- We'll take a look at each directory in detail...

The Heart of the Matter

- /bin
 - Common or shared executables, shell scripts
- /webapps
 - Java classes and JAR files
 - User web applications, WAR files
- /lib
 - Web container runtime libraries

Moving Targets

- /log
 - Log files
- /conf
 - Configuration files to setup the server
- /tmp
 - Temporary data
 - Usually only for duration of the STC
- /work
 - Another temporary place
 - Stays around after STC stops

Recommendations

- Watch sizes of /tmp, and /work
 - Should be documented
 - Consider TFS
- Logging can quickly get out of control
 - Ask about different log levels
 - More logs could be in other directories than just /log
- Be aware of aggrgrow mount option
- Permanently mount STC-related file systems
 - Use BPXPRMxx
 - Careful about order

SMP/E Considerations

- Vendors handle things differently
 - Each file is an element
 - Could use symbolic links if long filenames
 - Multiple files as one element
- SMP/E work on different LPARs?
 - File system(s) have to be mounted on the SMP/E system
 - Use ADRDSSU to DUMP/RESTORE
- Make copies of customized files
 - SMP/E updates might overwrite

SMP/E Element Example

- `++HFS (ZTJNN2CT)`
- `DISTLIB (AETJHFS) SYSLIB (CETJFL5Q)`
- `BINARY PARM (PATHMODE (0,7,7,5) ,PROGCTL)`
- `LINK (' ../jboss-common-core.jar')`
- `RELFIL (1) .`

- `ADD DDDEF (AETJHFS) SHR`
- `DATASET (SYS1.CHORUS.DIST.AETJHFS) .`
- `ADD DDDEF (CETJFL5Q)`
- `PATH (' /cai/chorus/jboss/client/TPV/') .`

- `/cai/chorus/jboss/client/activation.jar`
- `/cai/chorus/jboss/client/commons-logging.jar`
- `/cai/chorus/jboss/client/jboss-common-core.jar`
- `/cai/chorus/jboss/client/TPV/ZTJ9FQPO`
- `/cai/chorus/jboss/client/TPV/ZTJOQ01E`
- `/cai/chorus/jboss/client/TPV/ZTJNN2CT`

More Information

- ABCs of z/OS System Programming, Volume 9
 - <http://www.redbooks.ibm.com/abstracts/sg246989.html>
- z/OS v1r12 UNIX System Services Bookshelf
 - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/bpxzshb0
- JZOS Java Launcher and Toolkit
 - <http://www-03.ibm.com/systems/z/os/zos/tools/java/products/jzos/overview.html>
- Dovetailed Technologies Quickstart for Tomcat
 - <http://dovetail.com/products/tomcat.html>

Related Sessions

- Atlanta 10653: z/OSMF 1.13 Implementation and Configuration, Greg Daynes
 - z/OSMF architecture, you can apply what you learned here
- Atlanta 10980: Killing Zombies, Breaking Pipes, and other UNIX Shenanigans, Brandon Tweed
- Anaheim 9040: USS Filesystems, Ann Totten
 - Commands, Automount, Security
- Seattle 2285: Shell Commands, Ahilan Rajadeva
 - Great USS command introduction
- Austin 8368: JZOS, Hilon Potter

Questions?

- Kevin.Wick@ca.com

- Online Session Survey

- <https://share.confex.com/share/118/sessionevaluation.cgi>
- This was “UNIX Systems Services in Today’s Mainframe Software”, Wednesday, 9:30 AM