

Using System Symbols User Experience

J.O. Skip Robinson
Southern California Edison Co.

Monday 6 Aug 2012
Session 11839



Suggestions for installation defined system symbols

1. Use all eight characters in every name
 - a. Cannot substitute more characters than name length
 - b. Allow for maximum substitution just in case
2. Prefix every name with your installation identifier
 - a. Use SHARE code or other unique character string
 - b. Installation symbols will stand out
 - c. Installation symbols will group together in D SYMBOLS
3. Establish patterns for analogous names
 - a. Easier to recognize names used for each purpose
 - b. Easier to create new names in any schema

Examples of System Symbol Use

- 1. SYSNAME control**
- 2. CTC device management**
- 3. Disaster Recovery (DR) scenarios**

Generic Enterprise Configuration

CEC

C1

C2

C3

LPAR

L1 L2 L3

L1 L2 L3

L1 L2 L3

CEC/LPAR

C1/L1

C1/L2

C1/L3

C2/L1

C2/L2

C2/L3

C3/L1

C3/L2

C3/L3

Sys Name

E1

F1

G1

E2

F2

F4

E3

H1

F3

Plex ID

E

F

G (monoplex)

E

F

F

E

H (monoplex)

F

Determining **SYSNAME** based on CEC/LPAR

- Assumption
 - Each LPAR on each CEC hosts a specific system
 - **SYSNAME** should always be the same for a CEC/LPAR
- Problem
 - Making sure it's the right system on each CEC/LPAR
 - System workload may need to match CEC/LPAR size
 - Some software licensed only on certain CEC(s)
 - Wrong system can kill the right system
- Modern z/OS blocks rogue system takeovers
 - But only if the right system is currently running
- Solution: Derive **SYSNAME** from CEC/LPAR

Deriving **SYSNAME** from CEC/LPAR

IEASYMxx for sysplex 'F' (F1, F2, F3, F4)

```
SYSDEF
  SYSNAME(86)                /* Actual SYSNAME MUST BE SET LATER */

SYSDEF HWNAME(C1) LPARNAME(L2) /* FOR THIS CEC/LPAR */
  SYSNAME(F1)                /* USE THIS SYSNAME */

SYSDEF HWNAME(C2) LPARNAME(L2) /* FOR THIS CEC/LPAR */
  SYSNAME(F2)                /* USE THIS SYSNAME */

SYSDEF HWNAME(C3) LPARNAME(L3) /* FOR THIS CEC/LPAR */
  SYSNAME(F3)                /* USE THIS SYSNAME */

SYSDEF HWNAME(C2) LPARNAME(L3) /* FOR THIS CEC/LPAR */
  SYSNAME(F4)                /* USE THIS SYSNAME */

SYSDEF SYSCLONE(&SYSNAME(-2:2)) /* DERIVE SYSCLONE FROM SYSNAME */
```

Determining **SYSNAME**

- IPL takes place on a specific CEC in a specific LPAR
 - HMC icons are associated with a specific combination
 - Operator initiates IPL according to ICON selected
- During IPL, IEASYMxx is processed in sequence
- When CEC/LPAR matches, **SYSNAME** is set
- If no CEC/LPAR matches, **SYSNAME** remains as '86'
 - **&SYSCLONE** is also set to '86'
- **&SYSCLONE** is used to name system data sets
- LOGREC86, PAGE86, etc. do not exist
- **IPL fails rather than trying to grab wrong data sets**

Setting CTC Addresses by CEC/LPAR

- Problem: CTC addresses can be a quagmire to manage
 - Every LPAR should be able talk to every other LPAR...
 - ...on the same CEC
 - ...on another CEC on the same floor
 - ...on a CEC in a different data center via 'extender'
- Solution
 1. Define a schema for CTC device addresses by CEC/LPAR
 2. Use system symbols to define connections
- Connect appropriate CTC addresses in IODF
- **CTCs in all LPARs on all CECs connect to each other**

CTC Device Address Schema: **xyz**

<u>CEC</u>	<u>LPAR</u>		
C1	L1	L2	L3
C2	L1	L2	L3
C3	L1	L2	L3

x: e.g. 4 or 5 (for XCF PATHIN/PATHOUT)

z: e.g. 0 - 3 by exploiter, e.g. 0=XCF, 1=VTAM, ...

yy: uniquely assigned by CEC/LPAR (see below)

<u>CEC-LPAR</u>	<u>Sys Name</u>	<u>Plex ID</u>	<u>'To' CTC Device</u>
C1/L1	E1	E	x 11 z
C1/L2	F1	F	x 12 z
C1/L3	G1	G (monoplex)	x 13 z
C2/L1	E2	E	x 21 z
C2/L2	F2	F	x 22 z
C2/L3	F4	F	x 23 z
C3/L1	E3	E	x 31 z
C3/L2	H1	H (monoplex)	x 32 z
C3/L3	F3	F	x 33 z

Note: **xyz** used by **anyone** talking **to** that image

Setting System Symbols for CTC devices

IEASYMXX for symbols **&insC\$Sxx** on ***all*** systems

ins - 2 or 3 character installation prefix

C - 'CTC device marker'

\$S - 'system name marker'

xx - system identifier or **SYSLONE**

```
SYMDEF(&insC$SE1='11')
```

```
SYMDEF(&insC$SF1='12')
```

```
SYMDEF(&insC$SG1='13')
```

```
SYMDEF(&insC$SE2='21')
```

```
SYMDEF(&insC$SF2='22')
```

```
SYMDEF(&insC$SF4='23')
```

```
SYMDEF(&insC$SE3='31')
```

```
SYMDEF(&insC$SH1='32')
```

```
SYMDEF(&insC$SF3='33')
```

COUPLEXX Pathing for Sysplex 'E'

- PATHIN DEVICE(4&insC\$SE1.0) --> 4110
- PATHIN DEVICE(4&insC\$SE2.0) --> 4120
- PATHIN DEVICE(4&insC\$SE3.0) --> 4130

- PATHOUT DEVICE(5&insC\$SE1.0) --> 5110
- PATHOUT DEVICE(5&insC\$SE2.0) --> 5120
- PATHOUT DEVICE(5&insC\$SE3.0) --> 5130

CTC Device Addresses within a Sysplex

- A CTC cannot talk to itself
- Going from CTC-addr to same CTC-addr gets XCF error
 - IXC355I DEVICE (LOCAL/REMOTE): xyz/????
 - '????' indicates that device is unreachable
- Could limit each system's definitions to only 'valid' addrs
 - Requires either separate COUPLExx for each system...
 - Or more elaborate management of system symbols
- Or just tolerate the error message
 - Treat it as informational noise
 - But don't ignore real error conditions!

Managing Disaster Recovery Scenarios

- Normal system operations require many ‘values’
 - JES (devices for example)
 - VTAM (APPLs for example)
 - Tape management (volser ranges for example)
- DR operations require different ‘values’
 - Different CEC/LPAR
 - Different accessible devices
 - Different notion of local vs. remote
- Problem: how to set appropriate ‘values’ automatically?
- Solution: use system symbols to establish environment
- **AutoOps is guided by system symbol settings**

Three Modes or Phases in Disaster Recovery

1. Basic recovery phase (DR)
 - a. Requires manual recovery for many components
 - b. E.g. log streams, CICS, DB2, MQ
 - c. AutoOps starts up only basic z/OS and network functions
2. Network test mode (DN)
 - a. Basic recovery phase previously completed
 - b. AutoOps starts up limited system functions
 - c. DR environment exactly mimics production
 - d. Great care required not to allow 'accidental production'
3. Full production in DR environment (DP)
 - a. Used for all production IPLs after basic recovery is complete
 - b. AutoOps starts up all system functions tailored for DR
 - c. **DR environment now is production**

System Symbols to Control DR - 1

- **&insPLXID** is the first character of **SYSNAME**
- Normal Operation mode IPLs with [LOAD]00 on HMC
 - Contains line: IEASYM (01,02,03,L)
 - These members set symbols with normal values
 - SYMDEF(**&insPLXDR**='&insPLXID')
 - SYMDEF(**&insTYPDR**='&insPLXID')
- DR Recovery Mode IPLs with [LOAD]**DR** on HMC
 - Contains line: IEASYM (R1,R2,**DR**,L)
 - These members set symbols with DR environment values
 - SYMDEF(**&insPLXDR**='R') /* INDICATES DR MODE */
 - SYMDEF(**&insTYPDR**='R') /* RECOVERY PHASE */

System Symbols to Control DR - 2

- DR Network Test mode IPLs with [LOAD]**DN** on HMC
 - Contains: IEASYM (R1,R2,**DN**,L)
 - These members set symbols with DR network test values
 - SYMDEF(&insPLXDR='R') /* INDICATES DR MODE */
 - SYMDEF(&insTYPDR='N') /* NETWORK TEST MODE */
- DR Production Mode IPLs with [LOAD]**DP** on HMC
 - Contains: IEASYM (R1,R2,**DP**,L)
 - These members set symbols with DR production values
 - SYMDEF(&insPLXDR='R') /* INDICATES DR MODE */
 - SYMDEF(&insTYPDR='P') /* DR PRODUCTION MODE */

AutoOps Starts Tasks Based on System Symbols Determined by LOAD Parm



- **HSM**

- //HSM PROC CMD=&SYSCLONE
- // PARM=('CMD=&CMD')
- Normal/DR ops: S HSM → S HSM,CMD=DR

- **TCP/IP**

- //TCPIP PROC TYPE=""
- //PROFILE DD DSN=dsn(PROF&SYSCLONE&TYPE)
- Normal/DR ops: S TCPIP → S TCPIP,TYPE=DR

- **VTAM**

- Normal ops: S VTAM,,,(LIST=0A)
- DR ops: S VTAM,,,(LIST=F0)

17 ops

Complete your sessions evaluation online at SHARE.org/AnaheimEval