An Alternative Approach to the Configuration/Upgrade of the OMEGAMON/ITM Family of Products

Cecile C. Day
IBM Corporation

August 9, 2011 (Tuesday @ 11:00 AM-12:15 PM ET)
Session 10033
Portions of this presentation contain animation. Please select the **Slide Show** pull-down, and Select **View Show**
Agenda

- Mission Statement
- Summary – What is “PARMLIB”?
- z/OS Product Families Supported by PARMLIB and ICAT Configuration Modes
- PARMLIB Scope and Schedule
  - Summary of 1H11 Deliverables
- PARMLIB Workflow – Summary of Steps
- PARMLIB User Interface (UI) Roadmap
  - Parameter Generator UI (“PARMGEN”)
- PARMLIB Workflow – Details
  - Parameter On-line Help
  - PARMLIB CONFIG Profiles (Global and LPAR-specific {RTE})
  - Parameter Validation
  - $PARSE* “Create runtime members and jobs” Batch Job
- PARMLIB IVP
- PARMLIB Sample User Stories
- PARMLIB KCIJP* Batch Jobs
- ICAT Cross-Reference & Detailed Description
Mission Statement

• *Reduce our total cost of ownership (TCO) in the areas of installation, configuration, deployment and maintenance.*

  - As the installation and configuration expert of the OMEGAMON/ITM z/OS-based products, I want the configuration software to be simplified so that I can implement and deploy products to my monitoring environments, quickly and with minimal complexity.

  - As the installer of OMEGAMON/ITM z/OS-based products, I want configuration parameters to be simplified or eliminated so that I can install and configure product without needing to make choices unless a unique operational problem arises.

  - As an existing customer of OMEGAMON/ITM z/OS-based products, I want to quickly update (whether this is an upgrade or maintenance of an existing installation), without reconfiguration, so that I can minimize the time to implement and deploy upgrades.
Product-centric (ICAT) vs. Function RTE-centric jobs (PARMLIB)

144 ICAT product-centric jobs to configure 37 components for 1 LPAR RTE

8-10 PARMLIB function-centric jobs to configure components for 1 LPAR RTE regardless how many products!
How do we get there?

- One way is to come up with an alternative to ICAT that aligns itself with other z/OS product installations.
- The z/OS Road Ahead of Configuration - PARMLIB
PARMLIB:

New: Alternative configuration method to the ICAT (a.k.a. z/OS Configuration Tool) for OMEGAMON/ITM-based products.

Intuitive: Update PARMLIB RTE CONFIG profiles to choose the products you want configured into your runtime environment (RTE) and provide overrides.

Fast: Just submit about 10 "RTE-centric" batch jobs.

Convenient: An improved implementation of system variable support aids in faster deployment.

No worries: PARMLIB does not replace ICAT for a while, so use ICAT while you get to know PARMLIB. It is being delivered in phases throughout 2011+ to maximize the opportunity for customers to provide their input and influence the design.

Available: Download the latest PARMLIB PTF and get started!

For more information:

Master PARMLIB Technote

URL: http://www-01.ibm.com/support/docview.wss?uid=swg21417935
Scope of Support:

- The 2010 initial PARMLIB deliverables focused on enabling all 37 components that support ICAT today, to now be configured using the alternative PARMLIB approach, in order to create a brand new RTE. Upcoming phases are planned throughout 2011+ to focus on RTE maintenance & deployment best practices and performance improvements, of PARMLIB-created RTEs.

- The plan is to have the next versions of the products (post-OMEGAMON XE V420s, post-ITM6.2.2, etc.) to be fully supported for the end-to-end life cycle of an RTE (i.e., create/convert new PARMLIB RTE, customize/configure RTE, deploy RTE, maintain RTE and upgrade RTE).

- Interoperability support extends only to setting up a new PARMLIB environment based on existing ICAT RTE values to quickly set-up the PARMLIB CONFIG user profiles. Maintenance of the new RTE will only be done via PARMLIB mode exclusively; i.e., ICAT→PARMLIB but not PARMLIB→ICAT.
z/OS Product Families Supported by PARMLIB and ICAT Configuration Modes
**z/OS Product Families that PARMLIB & ICAT Support**

- **z/OS TMS family**
  - TEMS

- **OMEGAMON family**
  - z/OS, CICS, DB2, IMS, Storage, OMEGAVIEW, Management Console, Mainframe Networks, Messaging

- **z/OS ITCAM family**
  - SOA, WebSphere (Appl. Diagnostics), File Transfer Enabler

- **NetView family**
  - NetView for z/OS Agent

- **Rocket family**
  - Advanced Audit for DFSMS/SHsm, Advanced Catalog Management, Allocation Optimizer, Advanced Reporting, Automated Tape Allocation Manager, Tape Optimizer

- **System Automation family**
  - System Automation for z/OS Agent

- **TDS family**
  - Tivoli Decision Support Agent

**ICAT & PARMLIB Configuration Framework**
2011+ z/OS Configuration Roadmap
Project Scope and Schedule – Details
PARMLIB Phase 1.n 2010 Contents

Base Contents:
- Provide PARMLIB samples, KCIJP* batch jobs and KCIPLE PARSE utility to support a brand new runtime environment (RTE) set-up.
- Support a PARMLIB “RTE-centric” approach to the sample KCIJP* PARMLIB jobs (jobs for allocation, load, Persistent Datastore, TEMS registration, security, system set-up, etc.)
- Provide IVP function and IVP job logger/SUPERC report for KCIJP* PARMLIB jobs.
- Provide parameter on-line help utility and parameter validation (Iteration 1).
- For existing ICAT users, support interoperability:
  1. reuse existing RTE Batch Mode members to convert to PARMLIB CONFIG profiles.
  2. convert most commonly-updated parameters not externalized in the current ICAT to be supported for externalization / preserved customization in PARMLIB mode.
  3. convert a number of product-specific system library parameters into global parameters.
  4. rename parameter names to clearer, more self-describing parameter names.

Enablement Support:
- **Common Infrastructure:**
  - **Phase 1.1:** APAR#: OA30575 (CI) for PTF HKCI310 UA52371 (February 28, 2010)
  - **Phase 1.2:** APAR#: OA32122 (CI) for PTF HKCI310 UA53118 (May 31, 2010)

Documentation:
- Self-documenting PARMLIB members
- “PARMLIB - Alternative Configuration Mode for Pilot OMEGAMON z/OS Products” Newflash Technote #1417935
- New “IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS: Parameter Reference”
- New “Chapter 15. Using the PARMLIB method to set parameter values” in the Tivoli® OMEGAMON XE and Tivoli Management Services on z/OS V6.2.2 Common Planning and Configuration Guide
2011 Project Base Contents

2H10 PARMLIB Base Contents:
- Provide all previous phases’ PARMLIB Base Contents for all 37 ICAT-supported components.
- Provide System Variables support – Phase 1.
- Provide parameter validation.
- Provide PARMLIB configuration support/function exploitation of any new Interim Features of PARMLIB-enabled products in the base versions supported.

Enablement Support:
- Common Infrastructure:
  - Phase 1.3: APAR#: OA32126 (CI) for PTF HKCI310 UA53981 (August 31, 2010)
  - 4Q10: APAR#: OA34091 (CI) for PTF HKCI310 UA56531
  - Jan.’11: APAR#: OA35009 (CI) for PTF HKCI310 UA58103 (January 31, 2011)
  - Jul.’11: APAR#: OA35415 (CI) for PTF HKCI310 UA58791 (July 31, 2011)

2011/Future ICAT/PARMLIB Schedule:
2011+: ETA GA: TBD – several iterations being planned
- 3Q11: APAR#: OA37159 (CI) for PTF HKCI310 UA61621
- 4Q11: APAR#: OAnnrrrr (CI)
- 1Q12: APAR#: OAnnrrrr (CI)
- 2Q12: APAR#: OAnnrrrr (CI)
- 3Q12: APAR#: OAnnrrrr (CI)

2011/Future Base Contents:
- Provide additional RTE cloning and deployment improvements and time-to-value (TTV) ease-of-use enhancements.
- Integrate with Install Job Generator and future z/OS Management Facility (z/OSMF) Configuration Workflow UI.
- Provide PARMLIB configuration support/function exploitation of any new versions of PARMLIB-enabled products.
### Summary of 1H11 Deliverables

**A. PARMLIB Configuration Framework Enhancements:**

- Support Parameter Generator User Interface (PARMGEN) - Phase 1.
- Integrate Install Job Generator (JOBGEN) facility with the PARMGEN PARMLIB configuration process by sharing, harvesting, auto-discovering and reusing values from a common repository and extending the data to other future install/config. processes ("SHARE" model).
  - "Job Generator - Product Selection" updates - the product table has been updated with the latest list of supported products.
  - "Job Generator - Parameters" updates - jobcard information is now available for user customization. Same jobcard is reused in PARMGEN.
  - "Job Generator - CALLLIBS Selections" updates - default entries for the CALLLIBS system libraries are now provided as models. Same CALLLIBS system libraries are reused in PARMGEN if certain products also require the same system libraries.
A. PARMLIB Configuration Framework Enhancements: (cont’d)

- Provide a new KCIJPCCF standalone job to clone user-customized members (Kpp$/Kpp@** imbeds, $GBL$USR profile, $JOBCARD) from an existing RTE’s WCONFIG library to a new RTE (reuse the same customized members from version to version, RTE to RTE).

- Provide new $GBL$IBM and $GBL$USR global PARMLIB CONFIG profile members in WCONFIG (ideal for copying to other WCONFIG RTEs if these global system libraries are typically the same across LPARs. These values are being made available for configuration use to harvest the same information that may have already been customized from JOBGEN.

- Provide %GBL_SYSDA_UNIT% support in the KCIJPCFG set-up job and KCIJPUP1 IEBUPDTE job.
A. PARMLIB Configuration Framework Enhancements: (cont’d)

- Enhance WCONFIG(KCIJPUP1) populate TK*-->IK* IEBUPDTE job to back up WCONFIG library in each run based on "SET CLONE" setting in the job.
- Enhance WKANSAMU(KCIJPSUB) master auto-SUBMIT job to add a timer so jobs are submitted in priority sequence.
- Enhance WKANSAMU(KCIJV*) PARMLIB jobs to support user-defined system variables in addition to static symbols and KCIPARSE-extracted symbols.
- Enhance WCONFIG($PARSE*)-related jobs to add a //SYSVROUT DDNAME to report on all PARMLIB CONFIG parameters and variables, and their corresponding values used in the RTE configuration.
Using PARMLIB Application Configuration Enhancements:

- Provide PARMLIB application configuration support for:
  - OMEGAMON XE for Mainframe Networks (N3420) Interim Feature.
  - OMEGAMON XE for IMS V4.2.0 (I5420) Interim Features.
  - OMEGAMON XE for DB2 PE/PM V5.1.0 (D5510) additional upgrade configuration support.
  - OMEGAMON XE for Messaging: WebSphere Message Broker Monitoring V7.0.1 (QI701) Fix Pack 1(7.0.1.1-TIV-XEforMsg-FP0001) configuration support.
  - IBM Tivoli Advanced Catalog Management for z/OS V2.3.0 (RN230) and V2.4.0 (RN240) upgrade configuration support.

- Provide full High-Availability (HA) Hub TEMS configuration support for products that require additional HA Hub support installed @ the HA Hub (beyond the normal product catalog (KppCAT) and attribute (KppATR) files). These are products like OMEGAMON XE for Messaging and OMEGAMON XE for CICS on z/OS.
Summary of 1H11 Deliverables (cont’d)

- **B. PARMLIB Application Configuration Enhancements: (cont’d)**
  - Rearchitect PARMLIB System Variables support for the Persistent Datastore (PDS) facility of products that configure short-term historical data collection @ TEMS or Agent.
PARMLIB Workflow – Summary of Steps
Configuring products with the PARMLIB method: Steps

- **Step 1.** Apply the latest PARMLIB PTF.
- **Step 2.** Set up the PARMLIB work libraries for a runtime environment (RTE).
  - **Method 1:** Supply values for global parameters in ISPF panels (“PARMGEN”). -or-
  - **Method 2:** Edit the KCIJPCFG job directly.
- **Step 3.** Review the PARMLIB WCONFIG($JOBINDEX) job index planning purposes.
- **Step 4.** Submit the WCONFIG(KCIJPUP1) IEBUPDTE job to populate the IK* interim staging libraries.
- **Step 5.** Set up your PARMLIB configuration profiles (global and LPAR-specific).
- **Step 6.** Submit WCONFIG($PARSE) or WCONFIG($PARSESERV) job to create runtime members and WKANSAMU jobs.
- **Step 7.** Submit WKANSAMU batch jobs to complete the PARMLIB setup.
  - Submit the composite KCIJcSUB master PARMLIB auto-SUBMIT job instead of submitting the following jobs individually:
    - KCIJcALO composite runtime library allocation job
    - KCIJcLOD composite TK*→RK* runtime library load job
    - KCIJcSEC composite product security job
    - KCIJcUSP composite USS preparation job
    - KCIJcLNK composite ASM/LINK job
    - KCIJcUPV composite System Variables IEBUPDTE job
    - **KCIJcSYS composite system set-up and copy job**
    - **KCIJcUSS composite USS create HFS system set-up job**
    - **KCIJcCPY backup PARMLIB work libs. (IK*/WK*) or runtime (RK*) user libs. job**
    - **KCIJcW2R WK*→RK* deployment job**
    - KCIJcIVP configuration verification job
  - **c = P or V**
    - KCIJcSUB: non-SYSV mode
    - KCIJcVSUB: SYSV mode
- **Step 8.** Complete the configuration and start the products.
PARMLIB User Interface (UI) Roadmap
PARMLIB UI Roadmap – Current (pre-1H11)

Option #1: Set-up the PARMLIB work environment *manually*

- **Command:** `C 'RTE_VSAM_HILEV% 'TDITNT.ONESAPM' ALL`  
  - **Columns:** 00001 00072

  000039 //** 5 REQUIRED parameters to change:
  000040 //** - TDITNT.ITM62242 = SMP/E Target
  000041 //** High-Level Qualifier (HLQ) of the TKANPAR library.
  000042 //** - TDITNT.PARMLIB.JCL = User’s JCL library
  000043 //** for PARMLIB use. Typical PARMLIB members stored
  000044 //** in this library are applicable to all runtime
  000045 //** environments (RTEs). An example of a member that
  000046 //** gets created in this library is TESTSYSG
  000047 //** System Variable member for user-defined symbols
  000048 //** if System Variable mode is enabled in this RTE.
  000049 //** It is also where KCIJPCFG job gets customized
  000050 //** initially.
  000051 //** Tip: It is recommended that you specify a new
  000052 //** global library for PARMLIB common RTE usage.
  000053 //** - TDITNT.ONESAPM = Non-VSAM HLQ
  000054 //** of the PARMLIB WCONFIG control library and
  000055 //** the PARMLIB interim staging (IK*) and work output
  000056 //** (WK*) libraries representing the production
  000057 //** runtime (PR*) libraries.
  000058 //** - RTE_VSAM_HILEV% = VSAM HLQ
PARMLIB Workflow – Low-level details

Step 1. Apply the latest PARMLIB PTF.

HKCI310 PTF delivers the PARMLIB samples and runtime files per product, KCIJP* batch JCL, and other elements ($PARSE* KCIPARSE jobs, $SYSIN SYSIN control card, RTE CONFIG profile).


Step 1 Results:

✔ After the PTF is applied, the TKANCMD, TKANPAR and TKANSAM SMP/E target libraries contain the new PARMLIB elements.

✔ The TKANCUS SMP/E target library also contains new PARMLIB configuration code to support the PARMLIB functions such as the on-line parameter help facility, validation, conversion, and others.

✔ The TKANMOD SMP/E target library also contains the KCIPARSE PARMLIB module to support the file-tailoring functions.

جمال ممارسات: تنسخ وتخزين المكتبات المرجعية الحالية من SMP/E قبل تطبيق PTF.
PARMLIB Workflow – Low-level details (cont’d)

Step 2. Set-up the PARMLIB work environment by customizing the KCIJPCFG job.

- Copy the &thilev.TKANSAM(KCIJPCFG) job to a USER JCL library for PARMLIB use.
- Customize the JCL accordingly:

```
EDIT TDITNT.PARMLIB.JCL(KCIJPCFG) - 01.99
Command ==> Scroll ==> CSR
000793 * ******************************************************
000794 * ----------- BEGIN - USER SECTION: CONFIG ----------- *
000795 * ******************************************************
000796 * CONFIGURE FLAGS: Set to "Y" or "N". *
000797 * Note: &pppVER flag are for reference only. *
000798 * ******************************************************
000799 * Tivoli Enterprise Monitoring Server: KDS flag *
000800 SET CONFIGURE_TEMS_KDS = "Y"
000801 * IBM Tivoli OMEGAMON XE for CICS on z/OS: KC5 flag *
000802 SET CONFIGURE_CICS_KC5 = "Y"
000803 * IBM Tivoli OMEGAMON XE for CICS TG on z/OS: KGW flag *
000804 SET CONFIGURE_CICS_TG_KGW = "Y"
000805 * IBM Tivoli OMEGAMON XE for DB2 PE/PM: KD2 and KD5 flags *
000806 SET CONFIGURE_DB2_PEP_PM_KD2 = "Y"
000807 SET CONFIGURE_DB2_AGENT_KD5 = "Y"
000808 * IBM Tivoli OMEGAMON XE for IMS on z/OS: KI5 flag *
000809 SET CONFIGURE_IMS_KI5 = "Y"
000810 * IBM Tivoli OMEGAMON XE on z/OS: KM5 flag *
000811 SET CONFIGURE_ZOS_KM5 = "Y"
000812 * IBM Tivoli System Automation for z/OS: KAH flag *
```

Hands-on updates to KCIJPCFG set-up job

SMP/E and non-SMP/E runtime high-level qualifiers (HLQs)

Optional SMS-related values

Runtime environment (RTE) name

"CONFIGURE_PRODUCTS" product selection list

Submit the job to:

- allocate the &rhilev.JKANWCONFIG PARMLIB work control library.
- allocate the PARMLIB interim staging libraries (IKAN*, IKD2*) and work output libraries (WKAN*, WKD2*) representing the equivalentof RKANCMDU/RKANPARU/RKANSAMU and RKD2PAR/RKD2PRF/RKD2SAM production runtime user libraries.

[where &thilev = SMP/E target high-level qualifier; &rhilev = RTE HLQ; &rte = RTE name]
Option #2: Set-up the PARMLIB work environment via Job Generator

Welcome to the Job Generator. This routine will generate batch jobs to create and update an SMP/E environment.

Processing will be done in the following steps requiring user input.
(1) Select the products for installation into an SMP/E environment.
(2) Enter values for data set allocation and SMP/E processing.
(3) Enter values for HFS/zFS or CALLLIBS processing when required.

All information will be saved into an output PDS. Restarting the processor with this PDS will cause the values to be reinstalled on each panel. For this reason, a positive response will be required to accept values and selections.

Enter the fully qualified PDS name and output location for the generated jobs.

<table>
<thead>
<tr>
<th>Output PDS Name</th>
<th>Volser</th>
<th>STORCLAS</th>
<th>MGMTCLAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDITNT.PARMLIB.JCL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter=Next   F1=Help   F3=Back

or ➔ TSO EX ‘&thilev.TKCIINST (KCIJG00)’
Option #2: Set-up the PARMLIB work environment via Job Generator

Select the products to be included from the install media. The list below is the list of all supported products and might have entries for products that are not available. Be sure to select only products that exist for this install.

You must select at least one product.
To add additional products, type UPDATE on the command line and hit ENTER.
Verify your selections and change this field to accept ==> Y (Y, N)

Clear all product selections (X)

<table>
<thead>
<tr>
<th>Sel</th>
<th>Product Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>IBM Tivoli Advanced Reporting and Management for DFSMSHsm</td>
<td>V2.4.0</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Automated Tape Allocation Manager for z/OS</td>
<td>V3.2.0</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Composite Application Manager for SOA</td>
<td>V7.1.1</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Decision Support for z/OS</td>
<td>V1.6.1</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Management Services on z/OS</td>
<td>V6.2.1</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli Management Services on z/OS</td>
<td>V6.2.2</td>
</tr>
<tr>
<td>S</td>
<td>IBM Tivoli Management Services on z/OS</td>
<td>V6.2.3</td>
</tr>
<tr>
<td></td>
<td>IBM Tivoli OMEGAMON DE on z/OS</td>
<td>V4.2.0</td>
</tr>
</tbody>
</table>
Option #2: Set-up the PARMLIB work environment via Job Generator

Provide a more integrated "SHARE" model:

- **S** - ave values
- **H** - arvest values from a common repository where user settings were previously configured once
- **A** - utodiscover values from a common repository where user settings were previously configured once
- **R** - euse values by any subsequent install/config. process
- **E** - xtend common JobGen./PARMLIB repository to other installers/configurators to render similar "SHARE" model (i.e. future z/OS MF Config. Workflow UI, 1SAPM Set-up Center)
Option #2: Set-up the PARMLIB work environment via Job Generator

**KCIPJG03**

Enter the CALLLIBS data set names for SMP/E target zone definitions. Entries beginning with an asterisk are models. Remove the asterisk and use or modify the entry to enable the data set name.

Verify your entries and change this field to accept => N (Y, N)

<table>
<thead>
<tr>
<th>DDname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSSLIB</td>
<td>MVS callable services</td>
</tr>
<tr>
<td>Data set name =&gt; SYS1.CSSLIB</td>
<td></td>
</tr>
<tr>
<td>SCCNOBJ</td>
<td>XL C Compiler object library</td>
</tr>
<tr>
<td>Data set name =&gt; CBC.SCCNOBJ</td>
<td></td>
</tr>
<tr>
<td>SCEEBIND</td>
<td>LE/370 C spt - XPLINK resident routines</td>
</tr>
<tr>
<td>Data set name =&gt; CEE.SCEEBIND</td>
<td></td>
</tr>
<tr>
<td>SCEEBND2</td>
<td>LE/370 C spt - XPLINK LP64 library</td>
</tr>
<tr>
<td>Data set name =&gt; CEE.SCEEBND2</td>
<td></td>
</tr>
<tr>
<td>SCEELIB</td>
<td>LE/370 C spt - side-deck library</td>
</tr>
<tr>
<td>Data set name =&gt; CEE.SCEELIB</td>
<td></td>
</tr>
<tr>
<td>SCEELKD</td>
<td>LE/370 C spt - non-XPLINK, short names</td>
</tr>
<tr>
<td>Data set name =&gt; CEE.SCEELKD</td>
<td></td>
</tr>
</tbody>
</table>
PARMLIB UI Roadmap – Parameter Generator UI (“PARMGEN”) – Phase 1

Option #3: Set-up the PARMLIB work environment via “PARMGEN”

Welcome to the PARMLIB configuration mode’s Parameter Generator User Interface (PARMGEN).

Specify the location of the PARMLIB global user JCL library.

**GBL_USER_JCL: TDITNT.PARMLIB.JCL**

Specify the PARMLIB CONFIG profile library and member. If this is an ICAT-to-PARMLIB conversion, specify the ICAT RTE Batch member location.

**TDQMP.TM623.INSTJOBS(TESTSYSG)**

If PARMLIB CSI parameters are to be obtained from a JOBGEN work file, then enter its location.

**TDITNT.PARMLIB.JCL**

Enter Jobcard data:

```plaintext
==>/CCAPIPLB JOB (ACCT), 'CECILE CAPINPIN-DAY', CLASS=A,
==>/ MSGCLASS=X, MSGLEVEL=(1,1), NOTIFY=&SYSUID.,
==>/ REGION=0M
==>/ *** SYSJOBNAME=&SYSJOBNAME% SYSMEMBER=&SYSMEMPER%
```

Enter=Next  F1=Help  F3=End/Cancel
PARMLIB UI Roadmap – Parameter Generator UI ("PARMGEN") – Phase 1 (cont’d)

Option #3: Set-up the PARMLIB work environment via "PARMGEN"

KCIPLB1 ---- SET-UP PARMLIB WORK ENVIRONMENT PARAMETERS (1 OF 2) ---------------

Enter parameter values appropriate for your environment:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBL_INST_HILEV</td>
<td>TDOMPT.ITM623</td>
</tr>
<tr>
<td></td>
<td>High-Level Qualifier (HLQ) of INSTLIB/INSTJOBS datasets</td>
</tr>
<tr>
<td>GBL_Target_HILEV</td>
<td>TDITNT.ITM62351</td>
</tr>
<tr>
<td></td>
<td>HLQ of SMP/E target (TK*) libraries</td>
</tr>
<tr>
<td>GBL_SYSDA_UNIT</td>
<td>SYSDA</td>
</tr>
<tr>
<td></td>
<td>Non-VSAM disk UNIT (global work datasets)</td>
</tr>
<tr>
<td>RTE_HILEV</td>
<td>TDITN.ONESAPM</td>
</tr>
<tr>
<td></td>
<td>Non-VSAM HLQ of PARMLIB work and runtime libraries</td>
</tr>
<tr>
<td>RTE_VSAM_HILEV</td>
<td>TDITN.ONESAPM</td>
</tr>
<tr>
<td></td>
<td>VSAM HLQ of the runtime (RK*) libraries</td>
</tr>
<tr>
<td>RTE_NAME</td>
<td>TESTSYSG</td>
</tr>
<tr>
<td></td>
<td>Runtime environment (RTE) name for this LPAR</td>
</tr>
<tr>
<td>CSI_DSN</td>
<td>TDITN.ITM62351.CSI</td>
</tr>
<tr>
<td></td>
<td>DSNNAME of the SMP/E CSI for this RTE</td>
</tr>
<tr>
<td>TARGET_ZONE</td>
<td>CANTZL</td>
</tr>
<tr>
<td></td>
<td>Name of the SMP/E target zone for this RTE</td>
</tr>
</tbody>
</table>

Enter=Next  F1=Help  F3=End/Cancel
PARMLIB UI Roadmap – Parameter Generator UI ("PARMGEN") – Phase 1 (cont’d)

Option #3: Set-up the PARMLIB work environment via “PARMGEN”

**KCIPPLB2 ---- SET-UP PARMLIB WORK ENVIRONMENT PARAMETERS (2 OF 2) ---------**

**COMMAND ==>**

Enter parameter values appropriate for your environment:

*Note: If using NONSMS-managed RTE_HILEV and RTE_VSAM_HILEV HLQs, then the RTE_SMS_VOLUME, RTE_SMS_VSAM_VOLUME and RTE_SMS_UNIT values are required.*

- **RTE_SMS_PDSE_FLAG:** Y (PDSE flag [Y, N])
- **RTE_SMS_UNIT:** RTEU____ (Non-VSAM disk UNIT type)
- **RTE_SMS_VOLUME:** RTEV__ (Non-VSAM disk VOLSER)
- **RTE_SMS_MGMTCLAS:** RTESMGT_ (Non-VSAM disk MGMTCLAS)
- **RTE_SMS_STORCLAS:** RTESTOR_ (Non-VSAM disk STORCLAS)
- **RTE_SMS_VSAM_VOLUME:** RTEVV_ (VSAM disk VOLSER)
- **RTE_SMS_VSAM_MGMTCLAS:** RTESVMGT (VSAM disk MGMTCLAS)
- **RTE_SMS_VSAM_STORCLAS:** RTEVSTOR (VSAM disk STORCLAS)

Enter=Next  F1=Help  F3=End/Cancel

“SHARE” & pre-populate config. values automatically – “Ask me once”
PARMLIB UI Roadmap – Parameter Generator UI ("PARMGEN") – Phase 1 (cont’d)

Option #3: Set-up the PARMLIB work environment via “PARMGEN”

```
KCIPPLB3          --- DISPLAY PARMLIB ENVIRONMENT ANALYSIS - Row 1 to 17 of 17
COMMAND ==>> _

Review message traffic before proceeding.
KCIRJG02 - I Starting 27 Jul 2011 00:01:29
KCIRJG02 - I Extracting information from:
KCIRJG02 - I  CSI - TDITNT.ITM62351.CSI
KCIRJG02 - I  TZONE - CANTZ1
KCIRJG02 - I  End of EXEC, RC = 0

Active FMIDs installed in target zone CANTZ1: 42
HABR310 HABO320 HAES220 HAKD230 HARH230 HCKM230 HKCF701 HKC1310 HKC5420 HKDB51X
HKDB510 HKDO181 HKDS623 HKD711 HKET620 HKGW420 HKHL410 HKL4520 HKLV623 HKM701
HKM0701 HKMV310 HKM5420 HKN3420 HKO8260 HKQ1701 HKRM230 HKRJ310 HKS320
HKRN230 HKRS110 HKRV220 HKRW220 HKS3420 HKT1710 HKW0310 HKY7110 HPMZ410
HTAP220 JKM0420

Active, installed components configured in ICAT RTE batch deck TESTSYSG: 28
KC5 KSB KDB KDS KD4 KGW KHL KIS KMC KMQ
KMV KM5 KN3 KQI KRC KRI KRR KRN KRY
KRW KS3 KWD KYN KD2 KD5 KAR KNA

Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down
```
PARMLIB UI Roadmap – Parameter Generator UI ("PARMGEN") – Phase 1 (cont’d)

Option #3: Set-up the PARMLIB work environment via "PARMGEN"

KCIPLIB4 EXCLUDE PRODUCTS FROM PARMLIB CUSTOMIZATION Row 1 to 15 of 28
COMMAND ===>

Select (X) products to EXCLUDE from PARMLIB customization.

When finished, change "N" to "Y" to confirm selections. Confirm ==> Y (Y, N)

ALL Exclude all not configured in ICAT RTE Batch deck
_ KAH* IBM Tivoli System Automation for z/OS V330
_ KC5* IBM Tivoli OMEGAMON XE for CICS on z/OS V420
_ KDD* IBM Tivoli Decision Support for z/OS V181
_ KDS* Tivoli Enterprise Monitoring Server V623
_ KD4* IBM Tivoli Composite Application Manager for SOA V711
_ KD5* IBM Tivoli OMEGAMON XE for DB2 PE/PM V510
_ KGW* IBM Tivoli OMEGAMON XE for CICS TG on z/OS V420
_ KHL* IBM OMEGAMON z/OS Management Console V410
_ KI5* IBM Tivoli OMEGAMON XE for IMS on z/OS V420
_ KMC* IBM Tivoli OMEGAMON XE for Messaging - WebSphere MQ Configuration V701
_ KMO* IBM Tivoli OMEGAMON XE for Messaging - WebSphere MQ Monitoring V701
_ KM5* IBM Tivoli OMEGAMON XE on z/OS V420
_ KNA* IBM Tivoli NetView for z/OS Agent V610
_ KN3* IBM Tivoli OMEGAMON XE for Mainframe Networks V420
PARMLIB UI Roadmap – Parameter Generator UI ("PARMGEN") – Phase 1 (cont’d)

Option #3: Set-up the PARMLIB work environment via “PARMGEN”

4 PARMLIB Workflow steps saved!

49 optional/required parms. automatically file-tailored!

File-tailored KCIJPCFG via “PARMGEN”
High-level Details of PARMLIB Workflow and Data Flow – Diagrams
PARMLIB Workflow - Diagram

The following Diagram 1 through Diagram 6 show a high-level overview of the steps involved in configuring the product in a new runtime environment (RTE) using the PARMLIB mode. The same RTE-centric procedure applies whether you are configuring one component or the whole suite of z/OS products.

Diagram 1

DONE! via PARMGEN already

Apply the latest PARMLIB HKCI310 PTF

Customize job #1: KCIJPCFG set-up job copied from TKANSAM to PARMLIB USER JCL

(Optional) Review the PARMLIB $JOBINDX job index in WCONFIG

DONE! via PARMGEN already
PARMLIB Workflow – Diagram (cont’d)

Diagram 2

A

DONE! via PARMGEN already

B

DONE! via PARMGEN already

SUBMIT

KCIJPUP1 IEBUPDTE job in WCONFIG

Set up the &rte_name PARMLIB CONFIG profile (IBM default or converted ICAT RTE Batch deck) in WCONFIG
(Optional)
Edit Kpp%C*, Kpp%P* & Kpp%S*
parameter override imbeds in
WCONFIG for inserting
into PARMLIB runtime
members

(Optional)
Edit $SYSIN
SYSIN Controls
in WCONFIG
(to activate
preferred CONFIG
profile if other than
the default)

(Optional)
KCIJPCNV
conversion
job in WCONFIG
(if you have existing
ICAT RTE Batch
decks in INSTJOBS)
(Optional) Customize &rte_name member in PARMLIB USER JCL to define any user-defined symbolics if System Variables is enabled in the RTE.

$PARSE
file-tailoring job in WCONFIG (if System Variables is not enabled in the RTE)

or

$PARSES
V
file-tailoring job in WCONFIG (if System Variables is enabled in the RTE)
PARMLIB Workflow – Diagram (cont’d)

Includes load for new TK* to RK* libraries

Includes alloc. for new products’ libraries

Includes copy of new products STC & node to SYS1 libs.

(Optional) KCIJPSEC composite security job in WKANSAMU

KCIJPUSP composite USS preparation job in WKANSAMU (required for some products only)

KCIJP* PARMLIB function-centric jobs

Diag 5A
PARMLIB Workflow – Diagram (cont’d)

D2

SUBMIT

KCIJVALO composite allocation job in WKANSAMU

SUBMIT

KCIJVLOD composite load job in WKANSAMU

SUBMIT

KCIJVUSEC composite security job in WKANSAMU

(Optional)

KCIJVUSP composite USS preparation job in WKANSAMU (required for some products only)

SUBMIT

KCIJVUSP composite USS creation HFS job in WKANSAMU (required for some products only)

SUBMIT

KCIJVUPV composite System Variables IEBUPDTE job in WKANSAMU (if SYSV is enabled only)

SUBMIT

KCIJVUSV composite USS configuration verification job in WKANSAMU

SUBMIT

KCIJVSYN composite system set-up job in WKANSAMU

SUBMIT

KCIJVVLNK composite ASM/LINK job in WKANSAMU

SUBMIT

KCIJVIVP configuration verification job in WKANSAMU
Review and submit KCIJcSUB composite master SUBMIT job in WKANSAMU

$c = P \text{ or } V$

KCIJPSSUB: non-SYSV mode
KCIJVSSUB: SYSV mode
Review $IVPRPT report in WCONFIG and deltas in WSUPERC library

Deploy WK* runtime members to production RK* user libraries using site-approved CHG controls. (Optional) Use KCIJcCPY & KCIJcW2R clone/copy jobs in WKANSAMU

Perform applicable “Complete the configuration” steps

Start the product started tasks. (Optional) xxxxSTRT composite STC /S JCL & xxxxAPF APF listing available

DONE!

Same procedure whether you are configuring 1 component or 37 components!
Low-level Details of PARMLIB Workflow
PARMLIB Workflow – Low-level details

Step 1. Apply the latest PARMLIB PTF. HKCI310 PTF delivers the PARMLIB samples and runtime files per product, KCIJP* batch JCL, and other elements ($PARSE* KCIPARSE jobs, $SYSIN SYSIN control card, RTE CONFIG profile).


Step 1 Results:

- After the PTF is applied, the TKANCMD, TKANPAR and TKANSAM SMP/E target libraries contain the new PARMLIB elements.
- The TKANCUS SMP/E target library also contains new PARMLIB configuration code to support the PARMLIB functions such as the on-line parameter help facility, validation, conversion, and others.
- The TKANMOD SMP/E target library also contains the KCIPARSE PARMLIB module to support the file-tailoring functions.

Best Practices: Back-up the current SMP/E target libraries prior to applying the PTF.
PARMLIB Workflow – Low-level details (cont’d)

Step 2. Set-up the PARMLIB work environment by customizing the KCIJPCFG job.

- Copy the &thilev.TKANSAM(KCIJPCFG) job to a USER JCL library for PARMLIB use.
- Customize the JCL accordingly:

```
EDIT TDITNT.PARMLIB.JCL(KCIJPCFG) - 01.99
```

```
Command ===>
Scroll ===> CSR
000793 * **********************************************************************
000794 * ------------------------ BEGIN - USER SECTION: CONFIG --------------------- *
000795 * **********************************************************************
000796 * CONFIGURE FLAGS: Set to "Y" or "N".                             *
000797 * Note: &pppVER flag are for reference only.                      *
000798 * Tivoli Enterprise Monitoring Server: KDS flag                *
000800 SET CONFIGURE_TEMS_KDS = "Y"
000801 * IBM Tivoli OMEGAMON XE for CICS on z/OS: KC5 flag            *
000802 SET CONFIGURE_CICS_KC5 = "Y"
000803 * IBM Tivoli OMEGAMON XE for CICS TG on z/OS: KGW flag          *
000804 SET CONFIGURE_CICS_TG_KGW = "Y"
000805 * IBM Tivoli OMEGAMON XE for DB2 PE/PM: KD2 and KD5 flags       *
000806 SET CONFIGURE_DB2_PEP_KD2 = "Y"
000807 SET CONFIGURE_DB2_AGENT_KD5 = "Y"
000808 * IBM Tivoli OMEGAMON XE for IMS on z/OS: K15 flag              *
000809 SET CONFIGURE_IMS_K15 = "Y"
000810 * IBM Tivoli OMEGAMON XE on z/OS: KM5 flag                     *
000811 SET CONFIGURE_ZOS_KM5 = "Y"
000812 * IBM Tivoli System Automation for z/OS: KAH flag              *
```

Hands-on updates to KCIJPCFG set-up job

(HLQs)

- Copy the &thilev.TKANSAM(KCIJPCFG) job to a USER JCL library for PARMLIB use.
- Customize the JCL accordingly:

```
EDIT TDITNT.PARMLIB.JCL(KCIJPCFG) - 01.99
```

```
Command ===>
Scroll ===> CSR
000793 * **********************************************************************
000794 * ------------------------ BEGIN - USER SECTION: CONFIG --------------------- *
000795 * **********************************************************************
000796 * CONFIGURE FLAGS: Set to "Y" or "N".                             *
000797 * Note: &pppVER flag are for reference only.                      *
000798 * Tivoli Enterprise Monitoring Server: KDS flag                *
000800 SET CONFIGURE_TEMS_KDS = "Y"
000801 * IBM Tivoli OMEGAMON XE for CICS on z/OS: KC5 flag            *
000802 SET CONFIGURE_CICS_KC5 = "Y"
000803 * IBM Tivoli OMEGAMON XE for CICS TG on z/OS: KGW flag          *
000804 SET CONFIGURE_CICS_TG_KGW = "Y"
000805 * IBM Tivoli OMEGAMON XE for DB2 PE/PM: KD2 and KD5 flags       *
000806 SET CONFIGURE_DB2_PEP_KD2 = "Y"
000807 SET CONFIGURE_DB2_AGENT_KD5 = "Y"
000808 * IBM Tivoli OMEGAMON XE for IMS on z/OS: K15 flag              *
000809 SET CONFIGURE_IMS_K15 = "Y"
000810 * IBM Tivoli OMEGAMON XE on z/OS: KM5 flag                     *
000811 SET CONFIGURE_ZOS_KM5 = "Y"
000812 * IBM Tivoli System Automation for z/OS: KAH flag              *
```

Hands-on updates to KCIJPCFG set-up job

(HLQs)
PARMLIB Workflow – Low-level details (cont’d)

Step 2. Set-up the PARMLIB work environment by customizing the KCIJPCFG job. (cont’d)

- Copy the &thilev.TKANSAM(CIUS.CFG) job to a USER JCL library
- Customize the JCL accordingly:
  - SMP/E and non-SMP/E runtime high-level qualifiers (HLQs)
  - SMS-related values
  - runtime environment (RTE) name
  - "CONFIGURE_PRODUCTS" product selection list
- Submit the job to:
  - allocate the &rhilev.&rte.WCONFIG PARMLIB work control library.
  - allocate the PARMLIB interim staging libraries (IKAN*, IKD2*) and work output libraries (WKAN*, WKD2*) representing the equivalent of RKANCMDU/RKANPARU/RKANSAMU and RKD2PAR/RKD2PRF/RKD2SAM production runtime user libraries.

[where &thilev = SMP/E target high-level qualifier; &rhilev = RTE HLQ; &rte = RTE name]


Step 2 Results:

- In WCONFIG, the following members are created by KCIJPCFG job and file-tailored based on the values you customized in the job:

Table B1. WCONFIG Contents After KCIJPCFG Jobrun.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCCF</td>
<td>WCONFIG override imbeds (Kpp$C*/Kpp@$C*, Kpp$P*/Kpp@$P*, and/or Kpp$S*/Kpp@$S*), $JOBCARD, etc., and would like to save time and reuse the same customized members for creating the next RTE, use this job to clone the already-customized WCONFIG members after you run the KCIJPCFG set-up job for the next RTE.</td>
</tr>
</tbody>
</table>

In 1Q11, new KCIJPCCF WCONFIG cloner job saves you time From having to copy any WCONFIG user overrides.
PARMLIB Workflow – Low-level details (cont’d)

Step 3. Review the PARMLIB $JOBINDX job index in WCONFIG for planning purposes.

TDITNT.ONESAPM.TESTSYSG.WCONFIG(%JOBINDX)

================================================================================================================================
* * * PARMLIB Batch Jobs Index * * *
================================================================================================================================

A PARMLIB job index [WCONFIG($JOBINDX) or IKANSAMU/WKANSAMU(KCIJ$NDX)] is also supplied for reference. It lists the required and optional jobs that should be submitted and executed in the order presented below. "User Copy" job location is also included:

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function</th>
<th>Required</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCFG</td>
<td>Set up the PARMLIB work libraries and Lists all KCIJP* PARMLIB jobs</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>KCIJPUP1</td>
<td>1. Populate the interim staging libraries</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

User Copy: Copy from TKANSAM to a user JCL library. A copy is created in WCONFIG.
PARMLIB Workflow – Low-level details (cont’d)

- Step 4. Set-up PARMLIB jobcard-related elements and online help macro.
  - Customize the optional JOBCARD macro if desired, then copy the JOBCARD and the KCIRPLBS parameter online help set-up macro from the PARMLIB WCONFIG work control library, to your SYSPROC concatenation.
  - Customize the sample $JOBCARD$ member in PARMLIB WCONFIG.

Tip: Via PARMGEN, these steps are already done.
A **JOBCARD** macro is supplied and user has the option to copy the macro to his/her SYSPROC concatenation. Sample **JOBCARD** macro is shown below:

```plaintext
TDITNT.ONE SAPM.TEST SYSG.WCONFIG(JOBCARD)
/** *******************************************************************
/** MEMBER: KCIRJCRD
/** PRIMARY SOURCE: TKANCUS(KCIRJCRD) USER COPY: WCONFIG(JOBCARD)
/** INSTRUCTIONS:
/** Copy the JOBCARD macro supplied in the
/** %RTE_HILEV%.%RTE_NAME%.WCONFIG
/** PARMLIB CONFIG control library to your SYSPROC concatenation.
/** Tip: Customize the JOBCARD macro based on user's site
/** requirements prior to copying it to the SYSPROC library.
/** To query what libraries are concatenated in your SYSPROC
/** concatenation, invoke the TSO ISRDDN command. To invoke
/** ISRDDN, use option 6 or invoke it from any ISPF panel at any
/** TSO command prompt, such as: COMMAND ==> TSO ISRDDN
/** *******************************************************************
ISREDIT MACRO
.
SET JC1 = &STR('//&SYSUID.A JOB (00192,B300,&SYSUID),&SYSUID,' )
SET JC2 = &STR('// CLASS=&C,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&&SYSUID,' )
SET JC3 = &STR('// REGION=0M ')
SET JC4 = &STR('// *ROUTE PRINT N1R1 ')
SET JC5 = &STR('// ** SYSJOBNAME=%SYSJOBNAME% SYSMEMBER=%SYSMEMBER% ')
.
```

---

**Notice:** This text is extracted from a document provided by SHARE in Orlando in 2011. The content is a sample **JOBCARD** macro for a system configuration purpose, illustrating the steps and considerations for copying and customizing such configuration files in a system environment.
A $JOBCARD sample is also supplied and user has the option to execute the JOBCARD macro or hand-update the sample below:

```
TDITNT.ONESAPM.TESTSYSG.WCONFIG($JOBCARD)
*******************************************************************************
=> JOBCARD
/*&ZUSER.A  JOB (ACCT),'NAME',CLASS=A,
 /*  MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
 /*  REGION=0M
```

Sample JOBCARD macro
---

```
EDIT  TDITNT.ONESAPM.TESTSYSG.WCONFIG(KCIJPCNV) - 01. Columns 000
Command ==>  Scroll =

*******************************************************************************
000001 //CCAPISYG JOB (00192,B300,CCAPI),'CECILE CAPINPIN-DAY',
000002 // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=CCAPI,
000003 // REGION=0M
000004 //ROUTE PRINT N1R1
000005 //** SYSJOBNAME=CCAPISYG SYSMEMBER=KCIJPCNV
000006 //** System-Defined Jobcard: 01. Columns 000
000007 //** Member: KCIJPCNV
000008 //** Master Source: TDITNT.ITM62242.TKANSAM(KCIPRMLB)
000009 //** KCIJPUP1 Batch Job Output:
000010 //** TDITNT.ONESAPM.TESTSYSG.IKANSAMU(KCIJPCNV) - IBM Default
000011 //** TDITNT.ONESAPM.TESTSYSG.WCONFIG(KCIJPCNV) - Customer Copy
000012 //** $PARSE or $PARSESYY Batch Job Output:
000013 //** TDITNT.ONESAPM.TESTSYSG.WKANSAMU(KCIJPCNV)
000014 //** System-Defined Jobcard: 01. Columns 000
```

Example of PARMLIB-processed SYSJOBNAME and SYSMEMBER
PARMLIB Parameter On-line Help

The **KCIRPLBS** on-line parameter help set-up macro copied from your SYSPROC concatenation is issued on the command line:

```
EDIT TDITNT,ONESQPM,TESTSYSG,WCONFIG(TESTSYSG) - 01. Columns 00001 00120.
Command == \%TSO KCIRPLBS
```

**PARMLIB on-line parameter help set-up macro**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000090</td>
<td><strong>PARMLIB CONFIG Parameter</strong> PARMLIB CONFIG Value</td>
</tr>
<tr>
<td>000091</td>
<td><strong>SECTION: CONFIGURE_PRODUCTS:</strong></td>
</tr>
<tr>
<td>000095</td>
<td>* Note: Specify &quot;Y&quot; or &quot;N&quot; to the product-specific CONFIGURE_x_&amp;ppp product flag if the &amp;ppp product is to be configured in the TESTSYSG RTE:</td>
</tr>
<tr>
<td>000096</td>
<td><strong>Tivoli Enterprise Monitoring Server: KDS flag</strong></td>
</tr>
<tr>
<td>000097</td>
<td><strong>CONFIGURE_CICS_KDS</strong> &quot;Y&quot;</td>
</tr>
<tr>
<td>000098</td>
<td><strong>IBM Tivoli OMEGAMON XE for CICS on z/OS: KC5 flag</strong></td>
</tr>
<tr>
<td>000099</td>
<td><strong>CONFIGURE_CICS_KC5</strong> &quot;Y&quot;</td>
</tr>
<tr>
<td>0000100</td>
<td><strong>IBM Tivoli OMEGAMON XE for CICS TG on z/OS: KGW flag</strong></td>
</tr>
<tr>
<td>0000101</td>
<td><strong>CONFIGURE_CICS_TG_KGW</strong> &quot;Y&quot;</td>
</tr>
<tr>
<td>0000102</td>
<td><strong>IBM Tivoli OMEGAMON XE for DB2 PE/PM: KD5 flag</strong></td>
</tr>
</tbody>
</table>

or ➔ **TSO EX ‘&thilev.TKANCUS (KCIRPLBS)’**
The **KCIRPLBS** on-line parameter help set-up macro copied from your SYSPROC concatenation is issued on the command line. A pop-up window is invoked:

A pop-up dialog is invoked to ask for the SMP/E target high-level qualifier of the TKANCUS library where the help members are read.

```
**KCIPPLBS** ------ **KCIRPLBS SETUP ROUTINE** ------
COMMAND ==> _

Enter the GBL_TARGET_HILEV SMP/E target HLQ:
Target HLQ ===> **TDITNT.1TM62351**

F3=Back (Cancel)
```
PARMLIB Parameter On-line Help (cont’d)

Once KCIRPLBS help macro is set-up, type **PFSHOW ON** to display the PF Keys. **Place the cursor** anywhere on the line containing the parameter for which help is to be displayed and hit PF14 (**F14=ParmHelp**).

```
IT TDINIT OnesAPM.TESTSYS.GWCONFIG(TESTSYS) - 01 Columns 0000 0072
Command ===> PFSHOW_ ===> CSR

00492 ** If the TEMS requires network interface list support: 000492 KDS_TEMS_TCP_KDEB_INTERFACELIST "|"*

000494 ** If the TEMS requires address translation support: 000495 KDS_TEMS_COMM_ADDRESS_XLAT N

000497 **KDS_TEMS_PARTITION_NAME INSIDE
000498 **KDS_PA BEGIN * Table begin *

000499 **KDS_PA01_ROW 01
000500 **KDS_PA01_PARTITION_NAME INSIDE
000501 **KDS_PA01_PARTITION_ADDRESS 9.42.46.21
000502 **KDS_PA END * Table end *

000503 ** TEMS VTAM information:
000505 KDS_TEMS_VTAM_LU62_DLOGMOD CANCTDCS
000506 KDS_TEMS_VTAM_LU62_MDEMTAB KDSMTAB1
000507 KDS_TEMS_VTAM_NETID USCAC001

000508 ** TEMS - local VTAM and logon information:
F13=PFK Help F14=ParmHelp F15=PFK Back F16=RETURN F17=RFIND F18=RCHANGE
F19=UP F20=DOWN F21=PFKShow F22=LEFT F23=RIGHT F24=RETRIEVE
```
Help is displayed in a pop-up dialogue. The utility isolates the parameter, perform a look-up, and displays a pop-up dialogue with the detailed help information.

**Tip**

- **Specify a list of network interfaces used by the server to use.** This parameter is required for sites that are running multiple TCP/IP interfaces or network adapters on the same z/OS image. Setting this parameter allows you to direct the Server to connect to a specific TCP/IP interface. Specify one or more network adapters by hostname (fully-qualified hostname, or first only part of the fully-qualified hostname), or by TCP addresses to be used for input and output. If your site supports DNS, you must enter the fully-qualified hostname. If your site does not support DNS, you must enter the short hostname or an IP address. If your site does not support DNS, you must enter the fully-qualified hostname. This field is only applicable for networks with multiple interface cards for which a specific output network interface is required.
PARMLIB Parameter On-line Help (cont’d)

From within this pop-up, if desired, hit **PF5 to VIEW** the entire KppAHELP/ KppBHELP files.

If an interface address or a list of interface addresses is specified, the Configuration tool generates the KDEB_INTERFACELIST parameter in the KDSENV member of the RKANPARU library. Note: Separate the entries using a null space between interface addresses. For example:

```
  ==> {129.0.131.214 SYS1 SYS.IBM.COM}
```

- **Required:** No
- **Maximum Length:** 44
- **Type of Data:** Character
- **Default value:**
- **PMap class:** TCP
- **PMap members:** KDSENV
- **PMap panels:** KDS&DSVPREF.PPC KDS&DSVPREF.PPD
- **PMap parm:** KDEB_INTERFACELIST=&DSKDEB
- **PMap skeletons:** KDS&DSVPREF.SBB
Step 5. Customize the composite **KCIJPUP1** IEBUPDTE job.

- Add the jobcard to **WCONFIG(KCIJPUP1)** job (either copy the $JOBCARD member you customized in prior step or execute the JOBCARD macro).

- Submit the job to:
  - populate the IK* interim staging libraries with the product-specific PARMLIB samples and elements packaged in the composite KppCMDLB/KppPRMLB master IEBUPDTE members from the SMP/E target libraries.
  - prepare applicable PARMLIB elements *dynamically* (**KCIJP** jobs, RTE CONFIG profile and SYSIN members) based on user-customizations from the KCIJPCFG set-up job.

**Via PARMGEN, this step is already done**
Step 5. Customize the composite KCIJPUP1 IEBUPDTE job.

Add the jobcard to WCONFIG(KCIJPUP1) job (either copy the $JOBCARD member you customized in prior step or execute

Add the jobcard to WCONFIG(KCIJPUP1) job (either copy the $JOBCARD member you customized in prior step or execute

Submit the job to:

- populate the IK* interim staging libraries with the product-specific PARMLIB samples and elements packaged in the composite KppCMDLB/KppPRMLB master IEBUPDTE members from the SMP/E target libraries.
- prepare applicable PARMLIB elements dynamically (KCIJP* jobs, RTE CONFIG profile and SYSIN members) based on user-customizations from the KCIJPCFG set-up job.


IK* libs. populated by KCIJPUP1 IEBUPDTE job

IK* libs. populated by KCIJPUP1 IEBUPDTE job
Step 5. Submit the composite KCIJPUP1 IEBUPDTE job.

Table B2. WCONFIG Contents After KCIJPUP1 Jobrun.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. KCIJP* WCONFIG jobs (See WCONFIG($JOBINDX) for more information)</td>
<td></td>
</tr>
<tr>
<td>KCIJP$R/KPARSEX</td>
<td>additional $PARSE*-related jobs. Same to the $PARSE/PARSESV jobs but these library-specific $PARSE jobs only perform the equivalent of &quot;PART 3 - WK* Steps&quot; without the &quot;PART 1 - VALIDATE Step&quot; &amp; &quot;PART 2 - CPYEMPTY Steps&quot; of the typical $PARSE/PARSESV jobs.</td>
</tr>
<tr>
<td></td>
<td>• KCIJP$RC/PARSEXCM is a subset of the $PARSE job to process the PARMLIB samples from IKANCMU into WKANCMU work output library.</td>
</tr>
<tr>
<td></td>
<td>• KCIJP$RM/PARSEXSM is a subset of the $PARSE job to process the PARMLIB samples from IKANSAMU into WKANSAMU work output library.</td>
</tr>
<tr>
<td></td>
<td>• KCIJP$RP/PARSEXPR is a subset of the $PARSE job to process the PARMLIB samples from IKANPARU into WKANPARU work output library.</td>
</tr>
<tr>
<td></td>
<td>• KCIJP$RPY/PARSEXDV is a standalone job that can be run to get a list of resolved values for KCIPARSE-extracted symbolics. The job provides for TYPE:CE (CHAR extracted) and TYPE:IE (INTEGER extracted) KCIPARSE-extracted symbolics for System Variables use in the PARMLIB CONFIG parameter values. (where y = C,M,P,Y, xx = CM, SM, PR, DV)</td>
</tr>
<tr>
<td>D. WCONFIG Customer Override Imbeds</td>
<td></td>
</tr>
<tr>
<td>Kpp$C/Kpp@C</td>
<td>PART 3 WCONFIGx Steps from KCIJPUP1 copy these PARMLIB override members from the respective IK* interim staging libraries to the WCONFIG PARMLIB control library. Please refer to &quot;Table C. WCONFIG Customer Override...&quot;</td>
</tr>
</tbody>
</table>
Step 6. Set up the PARMLIB CONFIG Profiles. A PARMLIB configuration profile contains parameter values for all the global, LPAR-specific RTE and product-specific parameters. You can set up a profile from any of the following inputs:

- a. New PARMLIB global ($GBL*) and LPAR RTE ($CFG$IBM/RTE_NAME) WCONFIG profiles - You can use the IBM-supplied CONFIG profile members in the WCONFIG library, and use the IBM-supplied default values as initial PARMLIB parameter values.

- b. Converted PARMLIB RTE Batch member - If you have an RTE that is already configured by the Configuration Tool (ICAT) method and you want to use the batch parameter values of that RTE, you can run a conversion tool (KCIJPCNV job) and use the existing parameter values as initial PARMLIB parameter values. Note: After you convert the batch parameter member and then use the PARMLIB method to configure a new RTE, you cannot use the Configuration Tool to edit or maintain the configuration.
**Time-saving tips about PARMLIB CONFIG Profiles:**

✓ Submit WCONFIG(KCIJPMCF) job to merge parameter values from an old CONFIG profile member into a new one. This job can be used to merge a backup WCONFIG profile to a new one (old $GBL$USR-->new $GBL$USR, old &rte_name LPAR profile-->new &rte_name). This job is handy when reconfiguring an RTE – i.e., when applying maintenance to an existing RTE and you want to override the IBM-supplied configuration defaults in global $GBL$IBM or LPAR RTE $CFG$IBM for new configuration options you want to exploit.

✓ Submit WCONFIG(KCIJPCNV) job to convert the existing parameter values as initial PARMLIB parameter values if you have an RTE that is already configured by the Configuration Tool (ICAT) method.
The PARMLIB CONFIG profile member for candidate global system-related values that can be reused for all LPARs. IBM-supplied default is shown below:

```
EDIT     TDITNT.ONEASAPM.TESTSYS.GWCONFIG($GBL$IBM) - 01. Columns 00001 00072
Command ==> _ Scroll ==> CSR
000031 ** ============================== ================================
000032 ** PARMLIB CONFIG Parameter   PARMLIB CONFIG Value
000034 ** ================
000035 * SECTION: GLOBAL_SMP_RTE_ENVIRONMENT: SMP/E and CALLLIBS values:
000036 **
000037 ***** SMP/E Target High-Level Qualifier (HLQ):
000038  GBL_TARGET_HILEV       "TDITNT.ITM62351"
000039 000040 ** PARMLIB User JCL:
000041  GBL_USER_JCL      "TDITNT.PARMLIB.JCL"
000042 000043 ** Configuration Tool (ICAT) work library HLQ of the INSTJOBS library:
000044  GBL_INST_HILEV     "TDOMPT.ITM623"
000045 000046 ** Global SYSDA unit:
000047  GBL_SYSDA_UNIT    "SYSDA"
000048 000049 ** Sysplex name:
000050  GBL_SYSPLEX_NAME  LPARPLEX
```

Values you customized in KCIJPCFG job

Autodiscovered value
A number of product-specific system library names in ICAT were converted to common, global parameters so the same value can be shared by other products that need the same configuration. These are created in the new WCONFIG($GBL*) profiles:

** Common system libraries (if applicable):**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBL_DSN_WMQ_SCSQANLE</td>
<td>&quot;CSQ.V7R0M1.SCSQANLE&quot;</td>
</tr>
<tr>
<td>GBL_DSN_WMQ_SCSQAUTH</td>
<td>&quot;CSQ.V7R0M1.SCSQAUTH&quot;</td>
</tr>
<tr>
<td>GBL_DSN_WMQ_SCSQLOAD</td>
<td>&quot;CSQ.V7R0M1.SCSQLOAD&quot;</td>
</tr>
</tbody>
</table>

** GBL_DSN_CEE_* system libraries:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBL_DSN_CEE_SCEELKED</td>
<td>&quot;CEE.SCEELKED&quot;</td>
</tr>
<tr>
<td>GBL_DSN_CEE_SCEERUN</td>
<td>&quot;CEE.SCEERUN&quot;</td>
</tr>
<tr>
<td>GBL_DSN_CEE_SCEEBIND</td>
<td>&quot;CEE.SCEEBIND&quot;</td>
</tr>
</tbody>
</table>

** GBL_DSN_SYS1_* system libraries:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBL_DSN_SYS1_PARMLIB</td>
<td>&quot;SYS1.PARMLIB&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_PROCLIB</td>
<td>&quot;SYS1.PROCLIB&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_SAXREXEC</td>
<td>&quot;SYS1.SAXREXEC&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_VTAMLIB</td>
<td>&quot;SYS1.VTAMLIB&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_VTAMSLT</td>
<td>&quot;SYS1.VTAMSLT&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_BRODCAST</td>
<td>&quot;SYS1.BRODCAST&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_CSSLIB</td>
<td>&quot;SYS1.CSSLIB&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_HSMLOGY</td>
<td>&quot;SYS1.HSMLOGY&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_LINKLIB</td>
<td>&quot;SYS1.LINKLIB&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_MODGEN</td>
<td>&quot;SYS1.MODGEN&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_SBLSSCLI0</td>
<td>&quot;SYS1.SBLSSCLI0&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_SBPXEXEC</td>
<td>&quot;SYS1.SBPXEXEC&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_SISTMAC1</td>
<td>&quot;SYS1.SISTMAC1&quot;</td>
</tr>
<tr>
<td>GBL_DSN_SYS1_MACLIB</td>
<td>&quot;SYS1.MACLIB&quot;</td>
</tr>
</tbody>
</table>

System libraries extracted from JobGen CALLLIBS repository you customized – example of “SHARE” model
** GBL DSN TCP * TCP system libraries:
- GBL_DSN_TCP_SYSTCPD_TCPDATA_MFN "TCPIP.SEZAINST(TCPDATA)"
- GBL_DSN_TCP_SYSTCPD_TCPDATA "TCPIP.SEZAINST"
- GBL_DSN_TCP_ETC_SERVICES "TCPIP.ETC.SERVICES"
- GBL_DSN_TCP_SEZACMTX "EZA.SEZACMTX"
- GBL_DSN_TCP_SEZARNT1 "EZA.SEZARNT1"
- GBL_DSN_TCP_SEZATCP "EZA.SEZATCP"

** GBL DSN CICS * CICS system libraries:
- GBL_DSN_CICS_CTG_DLL "SYS1.SCTGDLL"
- GBL_DSN_CICS_SCTGSID "CTG.V8R0M0.SCTGSID"
- GBL_DSN_CICS_SDFHC370 "DFH.V4R2M5P.SDFHC370"
- GBL_DSN_CICS_SDFHLOAD "DFH.V4R2M5P.SDFHLOAD"

** GBL DSN IMS * IMS system libraries:
- GBL_DSN_IMS_RESLIB "DFS.V12R0M0.SDFSRESL"
- GBL_DSN_IMS_SCEXLINK "IMS.SCEXLINK"
- GBL_DSN_IMS_SFUNLINK "IMS.SFUNLINK"

** GBL DSN DB2 * DB2 system libraries:
- GBL_DSN_DB2_SD5NLOAD "DSN.V9R1M0.SDSNLOAD"
- GBL_DSN_DB2_LOADLIB_V8 "DSN.V8R1M0.SDSNLOAD"
- GBL_DSN_DB2_LOADLIB_V9 "DSN.V9R1M0.SDSNLOAD"
- GBL_DSN_DB2_LOADLIB_V10 "DSN.VAR1M0.SDSNLOAD"
- GBL_DSN_DB2_RUNLIB_V8 "DSN.V8R1M0.RUNLIB"
- GBL_DSN_DB2_RUNLIB_V9 "DSN.V9R1M0.RUNLIB"
- GBL_DSN_DB2_RUNLIB_V10 "DSN.VAR1M0.RUNLIB"
- GBL_DSN_DB2_DSNEXIT "DSN.V9R1M0.DSNEXIT"

** GBL DSN NETVIEW * NetView system libraries:
- GBL_DSN_NETVIEW_CNMLINK "NETVIEW.VNRMN.CNMLINK"

** GBL DSN CSF * ICSF system libraries:
- GBL_DSN_CSF_SCSFMOD0 "CSF.SCSFMOD0"

* $GBL$IBM END

After a KCIJPCNV ICAT→PARMLIB conversion, any converted GBL_DSN_* are also generated In the LPAR-specific WCONFIG(RTE_NAME) which takes precedence over the $GBL* profile. This provides customers the flexibility to use the LPAR profile as the ultimate overriding CONFIG profile [as generated in WCONFIG($SYSIN)].
The RTE CONFIG User Profile allows the customer to override defaults as necessary. Sample \&rte\_name (modeled after \$CFG\$IBM IBM default) CONFIG member is shown below:

```plaintext
EDIT TDIPTNT.ONESAPM.TESTSYS.G.WCONFIG(TEESTSYS) - 01. Columns 00001 00072
Command ==> Scroll ==> CSR

000326 ** VTAM SNA values:
000327 RTE_VTAM_NETID SYGNETID
000328 RTE_VTAM_LU62_DLOGMOD CANCTDCS
000329 RTE_VTAM_LU62_MOTETAB KDSMTAB1
000330 RTE_VTAM_GBL_MAJOR_NODE KCANDLE1
000331 RTE_VTAM_APPLID_MODEL Y
000332
000333 ** TCP/IP communications values:
000334 RTE_TCP_HOST "SYSGHOST"
000335 RTE_TCP_STC "x"
000336 RTE_TCP_PORT_NUM 1918
000337
000338 ** (Optional) If any products to be configured in this RTE require
000339 ** Unix System Services (USS) directories created, specify the main RTE
000340 ** HFS/zFS directory (#rtedir):
000341 RTE_USS_RTEDIR "/tdiptnt"
000342
000343 ** Persistent Datstore options:
000344 RTE_PDS_KPDPROC_PREFIX KPDPROC
000345 RTE_PDS_FILE_COUNT 3
```

Autodiscovered value

Autodiscovered value
PARMLIB CONFIG User Profile – LPAR-specific (RTE)

If customer has an existing INST* environment, a conversion utility (KCIJPCNV job) is provided to convert the RTE Batch Parameter Member. Sample converted PARMLIB CONFIG member is shown below (TESTSYSG is the name of the RTE Batch Member):

```
***************************** Top of Data ***************************************
*                                                               *
* File created on 31 July 2011 at 23:38:40 by KCIRPLBC          *
* Input file was 'TDOMPT.ITM623.INSTJOBS(TESTSYSG)'             *
* RTE$ BEGIN *------------------------ CONFIGURATION TOOL V310 ------------------------*
RTE_NAME                          TESTSYSG

** Tivoli Enterprise Monitoring Server (TEMS) flag and CMS_NODEID name:
RTE_TEMS_CONFIGURED_FLAG          Y
RTE_TEMS_NAME_NODEID              "TESTSYSG:CMS"

** Security options:
** Specify the security system to be used for this RTE. Options are:
** RACF, SAF, ACF2, TSS, NAM, or NONE.
RTE_SECURITY_USER_LOGON           RACF
RTE_SECURITY_FOLD_PASSWORD_FLAG   Y

** System procedure libraries:
GBL_DSN_SYS1_PROCLIB              SYS1.SYSG.PROCLIB
GBL_DSN_SYS1_VTAMLST              SYS1.SYSG.VTAMLST

** Persistent Datastore options:
RTE_PDS_HILEV                     TDITNT.ONESAPM.TESTSYSG
RTE_PDS_KPDPROC_PREFIX            KPDPROC
RTE_PDS_FILE_COUNT                3
```

Clearer parameter names
PARMLIB CONFIG User Profile – LPAR-specific (RTE)

The RTE CONFIG User Profile has an optional USER PROLOG SECTION if you wish to log your changes:

```
TDITNT.ONESAPM.TESTSYSG.WCONFIG(TESTSYSG)
******************************************************************************* Top of Data **********

<table>
<thead>
<tr>
<th>NO.</th>
<th>CHANGE DESCRIPTION</th>
<th>DATE</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>@03</td>
<td>Override KC5_X_AGT_STORAGE_* LIMIT()/RESERVE()</td>
<td>07/31/11</td>
<td>CD</td>
</tr>
<tr>
<td>@02</td>
<td>Set GBL_DSN_TCP_SYSTCPD_TCPDATA to new library</td>
<td>07/31/11</td>
<td>CD</td>
</tr>
<tr>
<td>@01</td>
<td>Override K%%_X_STC_SYSTCPD_INCLUDE_FLAG=Y</td>
<td>07/31/11</td>
<td>CD</td>
</tr>
</tbody>
</table>

*******************************************************************************

** Additional OMEGAMON XE for CICS Agent settings: **
KC5_X_AGT_STORAGE_LIMIT_EXTEND    23
KC5_X_AGT_STORAGE_LIMIT_PRIMARY   20
KC5_X_AGT_STORAGE_RESERVE_PRI     4096
KC5_X_AGT_STORAGE_RESERVE_EXT     4096
KAG_X_STC_SYSTCPD_INCLUDE_FLAG    Y
KAG_X_KDE_TRANSPORT_HTTP_OPTIONS  "HTTPS:0 HTTPS_CONSOLE:Y"
KAG_X_KDE_TRANSPORT_POOL_OPTIONS  "POOL:1000-1023 POOL:3000-4023"
KAG_X_KDE_TRANSPORT_OPTIONS       "EPHEMERAL:Y"
```

*In ICAT, these parameters equate to hardcoded settings (not externalized on ICAT panels)*
PARMLIB Workflow – Low-level details (cont’d)

Step 7. Finish setting up your PARMLIB configuration profile:

- a.) (Optional) In WCONFIG library, edit the $SYSIN SYSIN control member to activate the preferred configuration profile, if not already activated by default.

- b.) (Optional) In WCONFIG library, edit the applicable Kpp$C* (for WKANCMDOU member overrides), Kpp$P*/Kpp@P* (for WKANPARU member overrides), and Kpp$S* (for WKANSAMU member overrides) parameter override imbeds for the PARMLIB samples.

- c.) (Optional) In WCONFIG, submit the KCIJPVAL job to validate the input to the configuration profile before submitting the $PARSE job.

- d.) (Optional) In PARMLIB user JCL (%GBL_USER_JCL%), customize the &rte_name member which houses user-defined symbolics for an RTE, if System Variables mode is enabled for the configured RTE. These are user-defined symbolics in addition to the typical static system symbols defined in SYS1.IPLPARM and KCIPARSE system variables (for TYPE:CE (CHAR extracted) and TYPE:IE (INTEGER extracted) KCIPARSE-extracted symbolics for System Variables use in the SYSPRINT DDNAME’s GLOBAL VARIABLE TABLE SUMMARY of a KCIPARSE run).
PARMLIB $SYSIN SYSIN Control Card

TDITNT.ONESAPM.TESTSYSG.WCONFIG($SYSIN)

***************************** Top of Data ******************
* Purpose: Customer copy of SYSIN control for overriding CONFIG MEMBER=
* CONFIG profile members and SELECT MEMBER= member selection list.
* *******************************************************************
* USER SECTION: CONFIG/SELECT MEMBER
* *******************************************************************
.
* 3: $GBL$IBM IBM-supplied PARMLIB CONFIG profile (SMP-related and
* other global-specific parameters)
CONFIG MEMBER=(WCONFIG:$GBL$IBM)
* 4. $CFG$IBM IBM-supplied PARMLIB CONFIG profile (RTE-specific)
CONFIG MEMBER=(WCONFIG:$CFG$IBM)
* 5. $GBL$USR Customer-overridable PARMLIB CONFIG profile
* (SMP-related and other global parameters)
* Note: (OPTIONAL) Customize WCONFIG:$GBL$USR accordingly. It is
* ideal for copying to other WCONFIG RTEs if these global system
* libraries are typically the same across LPARs.
CONFIG MEMBER=(WCONFIG:$GBL$USR)
* 6. TESTSYSG Customer-overridable PARMLIB CONFIG profile
* (RTE-specific applicable to this LPAR)
CONFIG MEMBER=(WCONFIG:TESTSYSG)
SELECT MEMBER=(*)

CONFIG member based on RTE_NAME
PARMLIB Parameter Validation Report

TDITNT.ONESAPM.TESTSYSG.WCONFIG($VALRPT)

<table>
<thead>
<tr>
<th>CONFIG Files: File# DSNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 TDITNT.ONESAPM.TESTSYSG.WCONFIG($CFG$IBM)</td>
</tr>
<tr>
<td>2 TDITNT.ONESAPM.TESTSYSG.WCONFIG(TESTSYSG)</td>
</tr>
</tbody>
</table>

This report contains three sections:

1. Parameter Validation Errors
2. Parameter Values Changed from Defaults
3. Components Configured in this RTE

Section 1: Parameter Validation Errors

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>File#/Line#</td>
<td>Error Line1</td>
</tr>
<tr>
<td>KDS_X_TEMS_CONFIRM_SHUTDOWN</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>22/641</td>
</tr>
<tr>
<td></td>
<td>Value must be &lt;= 15.</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>22/641</td>
</tr>
<tr>
<td></td>
<td>Length must be &lt;= 2.</td>
</tr>
<tr>
<td>KDS_X_TEMS_TASKS_ATTACHED_NUM</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>22/643</td>
</tr>
<tr>
<td></td>
<td>Length must be &lt;= 1.</td>
</tr>
<tr>
<td>KDS_X_TEMS_WTO</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>22/640</td>
</tr>
<tr>
<td></td>
<td>Must be in list: Y,N.</td>
</tr>
</tbody>
</table>

Generated automatically by $PARSE* job or via WCONFIG(KCIJPVAL) standalone validation job
Step 8. Submit the $PARSE job in the PARMLIB WCONFIG library to process the PARMLIB samples from the interim (IK*) staging libraries into the corresponding work (WK*) output libraries. The $PARSE job performs the string substitutions and imbeds required by the user overrides in the PARMLIB CONFIG profile member. After completion of the $PARSE job, you have a complete set of customized runtime members in the work output libraries (WKANCMMDU, WKANPARU, WKANSAMU, WKD2PAR, WKD2PRF and WKD2SAM). This job is required if the RTE is not enabled for System Variables ("RTE_SYSV_SYSVAR_FLAG=N" parameter setting in the CONFIG profile).

Notes:
- If RTE is enabled for System Variables, then submit the $PARSES job instead. $PARSES job does not include the "PART 5 - KCIJPUP2 IEBUPDTE Steps" of the $PARSE job. Latter function is split into a standalone KCIJPUPV job as KCIJPUPV job must be submitted in the LPAR for which the system variables referenced in the IEBUPDTE members were configured, for proper resolution at product startup.
- $PARSE job’s first step validates the CONFIG profile parameter values. If you wish to run a standalone validation job prior to submitting $PARSE job, please refer to WCONFIG(KCIJPVAL) job.
The $PARSE KCIPARSE Batch JCL file-tailors the PARMLIB samples from the interim staging libraries (IK*) into the corresponding work output libraries (WK*) equivalent to the production runtime libraries (RK*):

```
TDITNT.ONESAPM.TESTSYSG.WCONFIG($PARSE)
************************************************** Top of Data ****
.
/* WKANPARU Step: IKANPARU-->WKANPARU */
// Process the PARMLIB members from IKANPARU to WKANPARU based on $CFG*
// CONFIG and/or converted PARMLIB RTE Batch deck profile settings.

WKANPARU EXEC PGM=KCIPARSE,COND=(4,LT,VALIDATE),
// PARM='MV=32000,MAXL=32000,ML=500,MI=255,MS=20000,LV=Y, LG=10'
//STEPLIB  DD DISP=SHR,
//         DSN=%GBL_TARGET_HILEV%.TKANMOD,
//INPUT1   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.IKANPARU,
//INPUT2   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU,
//WCONFIG  DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WCONFIG,
//SYSUT2   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU,
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSINLST DD SYSOUT=*
//SYSIN DD DISP=SHR,
//         DSN= %RTE_HILEV%.%RTE_NAME%.WCONFIG($SYSIN)
//SYSPRINT DD SYSOUT=*
//SYSGOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSGOUT DD SYSOUT=*
//SYSGOUT DD SYSOUT=*
```

**Input DD** – PARMLIB templates in IK* libraries, CONFIG profiles & Override imbeds in WCONFIG

**Output DD** - tailored runtime members written in WK* lib. based on CONFIG profile values

---

**Input DD** – PARMLIB templates in IK* libraries, CONFIG profiles & Override imbeds in WCONFIG

**Output DD** - tailored runtime members written in WK* lib. based on CONFIG profile values
PARMLIB $PARSE Batch JCL (cont’d)

TDITNT.ONESAPM.TESTSWSYS.WCONFIG($PARSE)
********************************************************************** Top of Data ********************************************************

.******************************************************************************

/** WKANPARU Step: IKANPARU-->$WKANPARU 
/** Process the PARMLIB members from IKANPARU to WKANPARU based on $CFG*
/** CONFIG and/or converted PARMLIB RTE Batch deck profile settings.
******************************************************************************

/**WKANPARU EXEC PGM=KCIPARSE,COND=(4,LT,VALIDATE), 
PARM='MV=32000,MAXL=32000,ML=500,MI=255,MS=20000,LV=Y, LG=10' 
STEPLIB DD DISP=SHR, 
  DSN=%GBL_TARGET_HILEV%.TKANMOD 
  INPUT1 DD DISP=SHR, 
    DSN=%RTE_HILEV%.%RTE_NAME%.IKANPARU 
  INPUT2 DD DISP=SHR, 
    DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU 
WCONFIG DD DISP=SHR, 
  DSN=%RTE_HILEV%.%RTE_NAME%.WCONFIG 
SYSUT2 DD DISP=SHR, 
  DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU 
SYSPRINT DD SYSOUT=* 
SYSPRINT DD SYSOUT=* 
SYSSINLST DD SYSOUT=* 
SYSSIN DD DISP=SHR, 
  DSN= =%RTE_HILEV%.%RTE_NAME%.WCONFIG($SYSIN) 
  DSN= =%RTE_HILEV%.%RTE_NAME%.WCONFIG($SYS$IBM)

SYSIN DD – list of CONFIG profiles ($GBL*, &rte_name) in priority sequence + list of members to be processed from input libraries

SYSVROUT DD – List of all variables and values in the $PARSE* RTE process (diagnostics)
The $PARSE KCIPARSE Batch JCL file-tailors the PARMLIB samples from the interim staging libraries (IK*) into the corresponding work output libraries (WK*) based on your PARMLIB CONFIG profile settings:

** Values that describe the TEMS to which the agent will connect:

- **KC5_TEMS_LOCAL_CONNECT_FLAG**       Y
- **KC5_TEMS_NAME_NODEID**                "TEST&SYSNAME.:CMS"

** Agent's Primary TEMS TCP/IP information:

- **Note:** KC5_TEMS_TCP_HOST and KC5_AGT_TCP_HOST must be the same value
- **if** KC5_TEMS_LOCAL_CONNECT_FLAG=Y (Agent connects to local TEMS)

** KC5_TEMS_TCP_HOST &SYSIPHOSTNAME.

** Agent's local TCP/IP information:

- **KC5_AGT_TCP_HOST**                  "&SYSIPHOSTNAME."
- **KC5_AGT_TCP_STC**                   "**"

** Agent's Primary TEMS VTAM information:

- **KC5_TEMS_VTAM_LU62_DLOGMOD**        CANCTDCS
- **KC5_TEMS_VTAM_LU62_MODETAB**        KDSMTAB1
- **KC5_TEMS_VTAM_NETID**               &SYSVTAMNETID.
PARMLIB $PARSE Batch JCL (cont’d)

The $PARSE KCIPARSE Batch JCL file-tailors the PARMLIB samples from the interim staging libraries (IK*) into the corresponding work output libraries (WK*) based on your PARMLIB CONFIG profile settings:

```
SDSF OUTPUT DISPLAY CCAPI$SG JOB27670 DSID LINE 16,996
COMMAND INPUT ===>                          SCROLL ===> CSR
00034 BEFORE IP.PIPE:%KC5_TEMSTCP_HOST%;
00034 AFTER  IP.PIPE:&SYSIPHOSTNAME.;\
00035 BEFORE IP.PIPE:%KC5_TEMSBKUP1_TCP_HOST%;
00035 AFTER  IP.PIPE:&AGT_BKUP1_TEMSTCP_HOST.;\n00039 BEFORE %KC5_TEMSVTAM_NETID%.\n00039 AFTER  &SYSVTAMNETID..\n00040 BEFORE %KC5_TEMSVTAM_APPLLLBBROKER%\n00040 AFTER  K&SYSCLONE.DSLB.\n00041 BEFORE %KC5_TEMSVTAMLU62DLOGMOD%.SNASOCKETS;
00041 AFTER  CANCTDCS.SNASOCKETS;\n. WRITE MEMBER KC5ENV RECORDS: 00102
```
Step 9A. Submit the composite \texttt{KCIJcSUB} master WKANSAMU PARMLIB auto-SUBMIT job instead of submitting the following jobs individually:

- \texttt{KCIJcALO} composite runtime library allocation job
- \texttt{KCIJcLOD} composite TK*\textrightarrow{}RK* runtime library load job
- \texttt{KCIJcSEC} composite product security job
- \texttt{KCIJcUSP} composite USS preparation job
- \texttt{KCIJcLNK} composite ASM/LINK job
- **\texttt{KCIJcUPV}** composite System Variables IEBUPDTE job
- **\texttt{KCIJcSYS}** composite system set-up job
- **\texttt{KCIJcUSS}** composite USS create HFS system set-up job
- **\texttt{KCIJcCPY}** backup runtime libraries job
- **\texttt{KCIJcW2R}** WK*\textrightarrow{}RK* deployment job
- \texttt{KCIJcIVP} configuration verification job

**Note:** Review the NOTES section of \texttt{KCIJcSUB} to see if certain jobs should be auto-submitted or not auto-submitted by \texttt{KCIJcSUB}.

\texttt{c} = P or V

- \texttt{KCIJ$_P$SUB}: non-SYSV mode
- \texttt{KCIJ$_V$SUB}: SYSV mode

In 4Q10, RTE-SUBMIT\_KCIJPSUB\_FLAG in KCIJPCFG allows you to auto-submit as part of the previous step’s $\text{PARSE}^*$ job.
Step 9B1. Submit the composite **KCIJcALO** allocation job in the WKANSAMU library to allocate the product execution (runtime) libraries.

- **Note:** *Required if KCIJcSUB job is not submitted.*

Step 9B2. Submit the composite **KCIJcLOD** load job in the WKANSAMU library to copy the SMP/E target elements to the runtime libraries.

- **Note:** *Required if KCIJcSUB job is not submitted.*

Step 9B3. (Optional) Submit the composite **KCIJcSEC** security job in the WKANSAMU library to create security-related members (load modules, encryption key, and other elements) based on the product security requirements.

- **Notes:**
  - *Required if the product-specific IBM-supplied security exit or input needs to be customized.*
  - *Required if KCIJcSUB job is not submitted.*

\[c = P \text{ or } V\]
Step 9B4. (Optional) Submit the composite KCIJcUSP USS preparation job in the WKANSAMU library to create the USS-related members in the RKANDATV RTE library for use in the composite KCIJcUSS job. See companion KCIJcUSS job.

**Notes:**
- Required if configuring products with USS req. (WebSphere Message Broker, ITCAM for SOA, and ITCAM for Application Diagnostics).
- Required if KCIJcSUB job is not submitted.

Step 9B5. (Optional) Submit the composite KCIJcLNK assembly/link job in the WKANSAMU library to assemble/link elements into the SYSLMOD RKANMODU user load library.

**Notes:**
- Required if configuring certain products only (OMEGAMON XE for Mainframe Networks)
- Required if the common KOBVTPL OBVTM1 exit for OMNIMON Base needs to be customized.
- Required if KCIJcSUB job is not submitted.
Step 9B6. (Optional) Submit the composite **KCIJcUPV** System Variables IEBUPDTE job. This job populates variable-named members contained in the application-specific KppJPUPB composite IEBUPDTE members in the WK* work output libraries.

**Notes:**
- **Required if RTE is enabled for System Variables support.** ("RTE_SYSV_SYSVAR_FLAG=Y" parameter setting in the CONFIG profile).
- **KCIJPSUV** (job submitted by **KCIJVSUB** auto-SUBMIT job) SUBMITs the **KCIJcUPV** job by default. Edit the **KCIJPSUV** job accordingly.
- **Required for submission in the target LPAR where the symbolic is resolved.**

Step 9B7. Submit the composite **KCIJcSYS** system-related set-up job in the WKANSAMU library to copy the product started tasks, VTAM major node members, and health check elements for the products and components into system libraries, and to assemble and link product module(s) into system libraries.

**Notes:**
- **Requires write access to system libraries.**
- **KCIJPSUB/KCIJPSUV** auto-SUBMIT jobs comment out the SUBMIT command for **KCIJcSYS** by default. Edit the **KCIJPSUB/KCIJPSUV** jobs accordingly.
Step 9B8. (Optional) Submit **KCIJcUSS** job to create the HFS directories and sub-directories and to copy files to HFS.

**Notes:**
- Required if configuring products with USS config. requirements (WebSphere Message Broker, ITCAM for SOA, and ITCAM for Application Diagnostics).
- **KCIJPSUB/KCIJPSUV** auto-SUBMIT jobs comment out the SUBMIT command for KCIJcUSS by default. Edit the KCIJPSUB/KCIJPSUV jobs accordingly.

Step 9B9. Submit the **KCIJcIVP** configuration verification job in the WKANSAMU library to verify that all the required runtime datasets members, and configuration jobs for this RTE were created, and that the jobs were executed successfully. Review the resulting output in WCONFIG($IVPRPT) report and WSUPERC output.

**Note:** Required if KCIJcSUB job is not submitted.
**TDITNT.ONESAPM.TESTSYSG.WCONFIG($IVPRPT)**

* THE REPORT CONTAINS THE FOLLOWING SECTIONS:
  * 1. REQUIRED CONFIGURATION BATCH JOBS
  * 2. REQUIRED SEQUENTIAL DATASETS
  * 3. REQUIRED PARTITIONED DATASETS AND MEMBERS.
  * 4. REQUIRED VSAM DATASETS

*****************************************************************************

**SECTION 1: REQUIRED CONFIGURATION BATCH JOBS**

*****************************************************************************

<table>
<thead>
<tr>
<th>JOB</th>
<th>STATUS</th>
<th>JOBNAME</th>
<th>JOB#</th>
<th>DATE</th>
<th>TIME</th>
<th>HI-CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCFG</td>
<td>OK</td>
<td>CCAPICFG</td>
<td>J04711</td>
<td>10.147</td>
<td>11:58:58</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPUP1</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J08747</td>
<td>10.148</td>
<td>15:26:58</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPCNV</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J04746</td>
<td>10.147</td>
<td>12:08:01</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPMCF</td>
<td>OPTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJPVAL</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J12895</td>
<td>10.147</td>
<td>12:10:39</td>
<td>00000</td>
</tr>
<tr>
<td>$PARSE</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J19382</td>
<td>10.148</td>
<td>15:47:07</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPALO</td>
<td>WARNING</td>
<td>CCAPI$JP</td>
<td>J06230</td>
<td>10.061</td>
<td>09:29:21</td>
<td>00002</td>
</tr>
<tr>
<td>KCIJPLOD</td>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJPSYS</td>
<td>WARNING</td>
<td>CCAPI$SY</td>
<td>J06331</td>
<td>10.061</td>
<td>09:38:36</td>
<td>00004</td>
</tr>
<tr>
<td>KCIJPUSP</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J19410</td>
<td>10.148</td>
<td>16:47:36</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPUSS</td>
<td>OK</td>
<td>CCAPI$SA</td>
<td>J19412</td>
<td>10.148</td>
<td>16:47:44</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPSEC</td>
<td>OK</td>
<td>CCAPI$SC</td>
<td>J06376</td>
<td>10.061</td>
<td>09:42:59</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPLNK</td>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJPCPY</td>
<td>OK</td>
<td>CCAPI$JP</td>
<td>J15302</td>
<td>10.063</td>
<td>12:00:05</td>
<td>00000</td>
</tr>
<tr>
<td>KCIJPW2R</td>
<td>OK</td>
<td>CCAPIW2R</td>
<td>J02160</td>
<td>10.139</td>
<td>09:25:07</td>
<td>00000</td>
</tr>
</tbody>
</table>

Generated by KCIJPIVP job

Review any STATUS=ERROR in the IVP report
**PARMLIB IVP (cont’d)**

```
TDITNT.ONESAPM.TESTSYSG.WCONFIG($IVPRPT)

******************************************************************************
* SECTION 2: REQUIRED SEQUENTIAL DATASETS
******************************************************************************
DSNAME                                        STATUS  JOBNAME
----------------------------------------------------------------------------------------
TDITNT.ONESAPM.TESTSYSG.RKM5LPR3                   OK      KCIJP
TDITNT.ONESAPM.TESTSYSG.RKM5LPR2                   OK      KCIJP
TDITNT.ONESAPM.TESTSYSG.RKM5LPR1                   OK      KCIJP
TDITNT.ONESAPM.LPARPLEX.RKM5PLX3                   OK      KCIJP
TDITNT.ONESAPM.LPARPLEX.RKM5PLX2                   OK      KCIJP
TDITNT.ONESAPM.LPARPLEX.RKM5PLX1                   OK      KCIJP
TDITNT.ONESAPM.TESTSYSG.RNASGRP3                   TDITNT.ONESAPM.TESTSYSG.RNASGRP3                   MISSING
TDITNT.ONESAPM.TESTSYSG.RNASGRP2                   TDITNT.ONESAPM.TESTSYSG.RNASGRP2                   MISSING
TDITNT.ONESAPM.TESTSYSG.RNASGRP1                   TDITNT.ONESAPM.TESTSYSG.RNASGRP1                   MISSING
TDITNT.ONESAPM.TESTSYSG.RKNAHIS3                   TDITNT.ONESAPM.TESTSYSG.RKNAHIS3                   MISSING
TDITNT.ONESAPM.TESTSYSG.RKNAHIS2                   TDITNT.ONESAPM.TESTSYSG.RKNAHIS2                   MISSING
TDITNT.ONESAPM.TESTSYSG.RKNAHIS1                   TDITNT.ONESAPM.TESTSYSG.RKNAHIS1                   MISSING
```

Check the offending job (KCIJPALO in this example) as to why the datasets are missing.
Step 10. Deploy the runtime members created in the WK* libraries into the production RK* runtime libraries. An optional KCIJcCPY backup job in the WKANSAMU library is provided to help backup the production execution (RK*) runtime user libraries and WK* work output libraries. Copy the WK* work output libraries to the respective production RK* runtime user libraries using your site-approved change process to update the RK* production libraries. An optional KCIJcW2R copy job may be used.

Step 11. Perform the applicable “Complete the configuration” steps as outlined in the product configuration guides then start the product started tasks. In addition, the following WKANSAMU jobs have been provided for certain requirements:

- Submit the composite xxxxSTRT STC startup member that has the /START &stc_name commands for all configured products as well as the composite APF authorization list of libraries (xxxxxAPF) (where xxxx = RTE_STC_PREFIX; “CANS” by default)
- If you have configured the OMEGAMON XE on z/OS product, review the WKANPARU(KM5PARM) parameter insert to SYS1.PARMLIB(CSFPRMxx) member if you intend to collect Integrated Cryptographic Service Facility (ICSF) data.
Sample PARMLIB User Stories
1. ICAT solicits the end-user for configuration information.
2. Customer-supplied values are verified for correctness.
3. ICAT reads in SMP/e-controlled PDS members.
4. ICAT substitutes the values supplied by the user while applying local knowledge, such as knitting together various application dependencies (i.e., runtime parameters for Hub TEMS).
5. ICAT generates customized runtime PDS members.
Sample PARMLIB User Stories

2. Clone a second RTE.
3. Add a product into an existing RTE.
4. Delete a product into an existing RTE.
5. Upgrade a product in an existing RTE.
6. Apply maintenance to an existing RTE but no new configuration changes.
7. Apply maintenance to an existing RTE but with new configuration changes (use configuration defaults).
8. Apply maintenance to an existing RTE but with new configuration changes (customer wants to override).
9. Use COPY jobs that will clone $JOBCARD, Kpp*$ into cloned RTE's WCONFIG.
Story#1: As a brand new OMEGAMON/ITM customer, I want to create a brand new RTE. This RTE needs to:

- Configure a High-Availability Hub TEMS (Shell Hub) that can be started on any LPARs.
- Support all z/OS-based products, including products with special Hub-type requirements beyond the normal KppCAT catalog and KppATR attribute requirements.
Sample PARMLIB User Stories (cont’d)

✓ Showcase TKANSAM(KCIJPCFG)->USER_JCL(KCIJPCFG) set-up job customization with no auto-submit.
✓ Showcase WCONFIG($JOBCARD/JOBCARD).
✓ Showcase WCONFIG(KCIJPUP1) IEBUPDTE job.
✓ Showcase WCONFIG($HAHUB) PARMLIB profile customization.
✓ Showcase WCONFIG($PARSE) job output.
✓ Showcase WKANSAMU(KCIJPSUB) auto-submit job with uncommented KCIJPSYS job.
✓ Showcase WKANSAMU(KCIJPALO) RTE Build job.
✓ Showcase WKANSAMU(KCIJPLOD) RTE Load job.
✓ Showcase WKANSAMU(KCIJPSYS) SYS1 processing job.
✓ Showcase WCONFIG($IVPRPT) error.
✓ Showcase WCONFIG($HAHUB) GBL_DSN_HZSPROC_LOADLIB correction and $PARSESEM job.
Story#2: As an existing OMEGAMON/ITM customer, I want to convert my ICAT-created Remote TEMS RTE to PARMLIB, without re-customizing all my previous ICAT settings. This ICAT RTE has configured all 37 components running on SYSG LPAR. PARMLIB Requirements: This converted PARMLIB RTE:

- Needs to preserve my ICAT manual overrides for RKANPARU(KC2SYS00) OMXE CICS CUA storage settings for LIMIT() and MINIMUM(). In ICAT, I have manually updated RKANPARU outside ICAT.
- Needs to preserve my ICAT manual overrides for RKANPARU(KOCVTM00) OMXE CICS Classic LROWS= and USER= (updated outside ICAT control).
- Needs to preserve my ICAT manual overrides for RKANPARU(KOSDEVIN) OMXE on z/OS DASD Data Collection settings (updated outside ICAT control).
- Exploit many types of system variables.
Sample PARMLIB User Stories (cont’d)

Q: How do I add delete a PARMLIB-created RTE?
A: See WKANSAMU(KCIIJPDEL) job.

Q: How do I apply maintenance to an existing RTE but no new configuration changes.
A: See WKANSAMU(KCIIJPLOD) job.

Q: How do I apply maintenance to an existing RTE but with new configuration changes (use IBM-supplied configuration defaults).
A: See WKANSAMU(KCIIJPMNT) job.

Q: How do I apply maintenance to an existing RTE but with new configuration changes (I want to override the IBM-supplied configuration defaults).
A: See WCONFIG(KCIIJPUP1) job and WCONFIG(KCIIJPMCF) job.

Q: How do I clone my WCONFIG RTE overrides and $JOBCARD to a 2nd RTE?
A: See WCONFIG(KCIIJPCCF) job.
Sample PARMLIB User Stories (cont’d)

Q: How do I add my override KDS_NCSLISTEN=512 parameter in KDSENV?
A: See WCONFIG(KDS$PENV) to add KDS_NCSLISTEN=512

Q: How do I turn off Autonomous Agent mode for all Agents?
A: See WCONFIG(KAG$PENV) to add IRA_AUTONOMOUS_MODE=N parameter

Q: How do I turn off Autonomous Agent mode for CICS Agent only?
A: See WCONFIG(KC5$PENV) to add IRA_AUTONOMOUS_MODE=N parameter

Q: How do I override KC2SYS* CICS CUA storage settings to MINIMUM(131072,X)?
A: See WCONFIG(&rte_name) PARMLIB CONFIG profile’s new KC2_X_CICS_STORAGE_MIN_EXTEND parameter

Q: How do I override KOCVTM* CICS Classic to LROWS=999?
A: See WCONFIG(&rte_name) PARMLIB CONFIG profile’s new KC2_X_CLASSIC_LROWS parameter

Q: How do I override KOSDEVIN DASD Data Collection settings for OMXE on z/OS Agent?
A: See WCONFIG(&rte_name) PARMLIB CONFIG profile’s new KM5_X_KOSDEVIN_* parameters
Q: How do I override KC5SYSIN's OMXE for CICS Agent storage settings to higher LIMIT() and RESERVE() and preserve those changes?
A. See WCONFIG(&rte_name) PARMLIB CONFIG profile’s new KC5_X_AGT_STORAGE_* parameters

Q: How do I preserve my SYSTCPD DD overrides in the TEMS and Agent started tasks?
A. See WCONFIG(&rte_name) PARMLIB CONFIG profile’s new Kpp_X_STC_SYSTCPD_INCLUDE_FLAG parameters

Q: How do I enable “Forward Take Action to NetView” for all the TEMS and Agent started tasks including CNMLINK RKANMODL DD support?
A. See WCONFIG(&rte_name) PARMLIB CONFIG profile’s global GBL_DSN_NETVIEW_CNMLINK and *_PPI_RECEIVER parameters.

Q: How do I override the Agent failover ITM default of switching back to the original primary TEMS?
A: See WCONFIG(KAG$PENV) to add CTIRA_PRIMARY_FALLBACK_INTERVAL=0

Q: If you refresh $CFG$IBM IBM Default CONFIG profile via maintenance, how do I sync-up my copy?
A: See WCONFIG(KCIJPMCF) merge CONFIG profile job
PARMLIB KCIJP* Batch Jobs – ICAT Cross-Reference
Think of KCIJPCFG’s “CONFIGURE PRODUCTS" section as the alternative to ICAT’s product-centric approach. In ICAT, you select one product at a time on the “Product Selection Menu", then configure that product, then select another product, etc.

In PARMLIB, select all products upfront then configure.
* USER SECTION: CONFIGURE_PRODUCTS

* ---------------------------------------- BEGIN - USER SECTION: CONFIG ------------------------------  

```
* IBM Tivoli Decision Support for z/OS: KDO flag
SET CONFIGURE_TDS_KDO       = "Y"
* IBM Tivoli Composite Application Manager for SOA: KD4 flag
SET CONFIGURE_SOA_KD4       = "Y"
* IBM Tivoli Advanced Audit for DFSMSShsm: KRG flag
SET CONFIGURE_AAD_KRG       = "Y"
* IBM Tivoli Advanced Reporting for DFSMSShsm: KRH flag
SET CONFIGURE_ARD_KRH       = "Y"
* IBM Tivoli Advanced Allocation Management for z/OS: KRJ flag
SET CONFIGURE_AAM_KRJ       = "Y"
* IBM Tivoli Automated Tape Allocation Manager for z/OS: KRK
SET CONFIGURE_ATAM_KRK      = "Y"
* IBM Tivoli Advanced Catalog Management for z/OS: KRN flag
SET CONFIGURE_ACM_KRN       = "Y"
* IBM Tivoli Advanced Backup and Recovery for z/OS: KRV flag
SET CONFIGURE_ABR_KRV       = "Y"
* IBM Tivoli Tape Optimizer for z/OS: KRW flag
SET CONFIGURE_TOZ_KRW       = "Y"
* ITCAM for Application Diagnostics on z/OS: KYN flag
SET CONFIGURE_ITCAMAD_KYN   = "Y"
```

25 product configuration flags total (1 for each suite)
KCIJPPR*/$PARSE* job – ICAT Cross-reference

**Sample ICAT Batch CICATB job report**

<table>
<thead>
<tr>
<th>GEN</th>
<th>MEMBER</th>
<th>JOB DESCRIPTION</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>DS#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>C2#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>C5#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS, AGT ADRSP</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DF#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>MV#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>M2#3TESTSYSG</td>
<td>CREATE RUNTIME MBRS</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>AH#4TESTSYSG</td>
<td>REGISTER PRODUCT TO THE TEMS</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>D5#4TESTSYSG</td>
<td>REGISTER PRODUCT TO THE TEMS</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>GW#4TESTSYSG</td>
<td>REGISTER PRODUCT TO THE TEMS</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>HL#4TESTSYSG</td>
<td>REGISTER PRODUCT TO THE TEMS</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>I5#4TESTSYSG</td>
<td>REGISTER PRODUCT TO THE TEMS</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>M5#ITESTSYSG</td>
<td>CREATE RUNTIME MBRS, AGT TEMS</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>S3#ITESTSYSG</td>
<td>CREATE RUNTIME MBRS, AGT TEMS</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>D2#XTESTSYSG</td>
<td>CREATE PROFILE MEMBERS</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>D2#6TESTSYSG</td>
<td>INSTALL DB2 SSID RELATED MBRS</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PD#PTESTSYSG</td>
<td>CREATE PDS MBRS</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>MQ#PTESTSYSG</td>
<td>CREATE PDS MBRS</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>S3#PTESTSYSG</td>
<td>CREATE PDS MBRS</td>
<td></td>
</tr>
</tbody>
</table>

Think of $PARSE as 1 job that performs the equivalent of running ICAT’s pp#3 jobs, pp#4 jobs, pp#I jobs, pp#X jobs, pp#P jobs, pp#G jobs, etc.
KCIJPALO job – ICAT Cross-reference

<table>
<thead>
<tr>
<th>DDNAME: MEMBER</th>
<th>*ICAT CROSS-REFERENCE/COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>%IMBED% = INPUT1: ???JPAL1</td>
<td>*pp#1 RTE Build job</td>
</tr>
<tr>
<td>%IMBED% = INPUT1: ???JPAL5</td>
<td><em>pp#4 TEMS registration for RKCP</em>/RKCF* VSAM Epilog VSAM</td>
</tr>
<tr>
<td>%IMBED% = INPUT1: ???JPALX</td>
<td>*pp#5 Allocate add'l for Epilog VSAM</td>
</tr>
<tr>
<td>%IMBED% = INPUT2: KC2JPA*</td>
<td>*C2#5 Allocate add'l using KC2##JPA template</td>
</tr>
<tr>
<td>%IMBED% = INPUT2: KC2JPH*</td>
<td>*C2#H RK2HIST historical using KC2##JPH</td>
</tr>
<tr>
<td>%IMBED% = INPUT2: KI2JPA*</td>
<td>*I2#5 Allocate add'l using KI2##JPA template</td>
</tr>
<tr>
<td>%IMBED% = INPUT1: ???JPALQ</td>
<td><em>pp#Q Persistent Datastore KppAL</em> jobs</td>
</tr>
</tbody>
</table>

Think of KCIJPALO as 1 job that performs the equivalent of running ICAT’s pp#1 jobs, pp#5 job, pp#4 job’s REPRO steps, pp#H jobs, pp#Q jobs, etc.
KCIJPLOD job – ICAT Cross-reference

TDITNT.ONESAPM.TESTSYSG.IKANSAMU(KCIJPLOD)

%IMBED% DDNAME:MEMBER *ICAT CROSS-REFERENCE/COMMENTS

%IMBED%=INPUT1:???JPLDA *pp#2 RTE Load job
%$IMBED_KDS_KDSJPLDB_INPUT1% *DS#2 RTE Load job
%$IMBED_KCI_KCIJPLD2_INPUT1% *pp#2 RTE Load job
%$IMBED_KCI_KCIJPLD3_INPUT1% *pp#2 RTE Load job
%$IMBED_KC5_KC5JPLD2_INPUT1% *C5#2 RTE Load job
%$IMBED_KGW_KGWJPLD2_INPUT1% *GW#2 RTE Load job
%$IMBED_KD5_KD5JPLD2_INPUT1% *D5#2 RTE Load job
%$IMBED_KI5_KI5JPLD2_INPUT1% *I5#2 RTE Load job
%$IMBED_KM5_KM5JPLD2_INPUT1% *M5#2 RTE Load job
%$IMBED_KOB_KOBJPLD2_INPUT1% *pp#2 RTE Load job
%$IMBED_KET_KETJPLD2_INPUT1% *pp#2 RTE Load job
%$IMBED_KN3_KN3JPLD2_INPUT1% *pp#2 RTE Load job
%$IMBED_KS3_KS3JPLD2_INPUT1% *S3#2 RTE Load job
%$IMBED_KWO_KWOJPLD2_INPUT1% *WO#2 RTE Load job
%$IMBED_KMQ_KMQJPLD2_INPUT1% *QI#2 RTE Load job

Think of KCIJPLOD as 1 job that performs the equivalent of running ICAT’s pp#2 jobs, etc.

%IMBED%=INPUT1:???JPLDX *Special exceptions steps
KCIJPSEC job – ICAT Cross-reference

Think of KCIJPSEC as 1 job that performs the equivalent of running ICAT’s DS#3 job’s KAES256 step, pp#3 job’s KLV@ASM step, pp#O jobs, etc.
KCIJPSYS job – ICAT Cross-reference

Think of KCIJPSYS as 1 job that performs the equivalent of running ICAT’s CB#N job, CB#P job, CB#K job, DS#L job, then running the sample jobs generated by these jobs, etc.
Think of **KCIJPLNK** as 1 job that performs the equivalent of running ICAT’s **pp#3** jobs that generate sample **ASM/LINK** jobs, etc.
**KCIJPUS% job – ICAT Cross-reference**

**TDITNT.ONESAPM.TESTSYSG.IKANSAMU (KCIJPUSP)**

```verbatim
%IMBED% DDNAME:MEMBER *ICAT CROSS-REFERENCE/COMMENTS
%IMBED%=INPUT2:???JPUS6 *pp#6 job to create RKANDATV members for USS
```

**TDITNT.ONESAPM.TESTSYSG.IKANSAMU (KCIJPUSSS)**

```verbatim
%IMBED% DDNAME:MEMBER *ICAT CROSS-REFERENCE/COMMENTS
%IMBED%=INPUT2:???JPUSU *pp#U SBPXEXEC job for USS
```

*Think of KCIJPUSP/KCIJPUSSS as 1 job that performs the equivalent of running ICAT’s pp#6/pp#U jobs for products that have USS requirements*
PARMLIB KCIJP* Batch Jobs
– Detailed Description
### PARMLIB KCIJP* Batch Jobs

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCFG</td>
<td>Set up the PARMLIB work libraries and configuration elements for the RTE. Its function is to:</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>- allocate the PARMLIB work control library (for the PARMLIB control members).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- allocate the PARMLIB interim staging libraries (IKAN*, IKD2*) and work output libraries (WKAN*, WKD2*) representing the equivalent of RKANCMDU, RKANPARU, RKANSAMU, RKD2PAR, RKD2PRF and RKD2SAM production runtime user libraries.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- copy/rename applicable PARMLIB control members from the SMP/E targets to PARMLIB WCONFIG.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- prepare the following members and jobs for subsequent PARMLIB processing:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$JOBINDX: PARMLIB Job Index README</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$HELPPLB: PARMLIB Parameter Help README</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$JOBCARD: Sample jobcard for user customization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$JOBCARD: Sample jobcard macro for user customization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$VERSION: Sample version file for reference</td>
<td></td>
</tr>
</tbody>
</table>

*New in 4Q10*

*Updated*
PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCFG</td>
<td>- prepare the following members and jobs for</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>subsequent PARMLIB processing:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KCIJPCFG: RTE-specific KCIJPCFG set-up job (copied</td>
<td></td>
</tr>
<tr>
<td></td>
<td>from CCAPI.PARMLIB.BETA.JCL)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KCIJPLOG: Job logger imbed for PARMLIB internal use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KCIJPUP1: IEBUPDTE job for user customization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(See $JOBINDX for more information).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KCIJPUP2: PRPKCIJP step refresh job</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(See $JOBINDX for more information).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KCIRPLBS: PARMLIB Parameter on-line help macro</td>
<td></td>
</tr>
</tbody>
</table>

New in 4Q10:
- KCIJPCFG

New in 1Q11:
- KCIJPUP2: PRPKCIJP step refresh job
- KCIRPLBS: PARMLIB Parameter on-line help macro

Updated
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function</th>
<th>Required Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCFG</td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>

- set-up KCIJPCCF WCONFIG cloner job. If you are rerunning KCIJPCFG set-up job for additional RTEs to set-up, and you want to clone any customized WCONFIG members from the first fully-deployed RTE, then use the WCONFIG(KCIJPCCF) WCONFIG cloner job after you run the KCIJPCFG set-up job for the next RTE.

- set-up KCIJPUP1 IEBUPDTE job and other PARMLIB elements dynamically created based on products or components selected for configuration on the "CONFIGURE_PRODUCTS" product selection list section of the job.

**User Copy:** Copy from TKANSAM to a user JCL library. A copy is created in WCONFIG.
### KCIJPCCF

Clone the WCONFIG customized members. If you are rerunning KCIJPCFG set-up job for additional RTEs to set-up, and you want to clone any customized WCONFIG members from the first fully-deployed RTE, then use the WCONFIG(KCIJPCCF) WCONFIG cloner job after you run the KCIJPCFG set-up job for the next RTE.

**User Copy:** WCONFIG(KCIJPCCF)
## PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Required</th>
<th>Function</th>
</tr>
</thead>
</table>
| KCIJPUP1  | Y        | 1. Populate the IK* interim staging libraries with the product-specific PARMLIB samples and elements packaged in the composite KppCMDLB/KppPRMLB master IEBUPDTE members from the SMP/E target libraries.  
2. Prepare applicable KCIJP* PARMLIB sample jobs for KCIPARSE processing. Examples of KCIJP* jobs prepared by KCIJPUP1 job are KCIJPCNV, KCIJPMCF, KCIJPVAL, KCIJPPRS ($PARSE) and KCIJPPRV ($PARSES). |
| KCIJPUP2  | N        | Standalone job to refresh the members created by the PRPKCIJP step of the KCIJPUP1 job. |

**Updated in 4Q10**

In 4Q10, customize new RTE_SUBMIT_KCIJPSUB_FLAG in KCIJPCFG job to build a $PARSE* that automatically submits KCIJPSUB job.
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function Description</th>
<th>Required</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPCNV</td>
<td>Convert an ICAT RTE Batch Parameter Member created via the Configuration Tool Batch Mode Process into PARMLIB configuration profile member to serve as input to the $PARSE (if RTE is not enabled for System Variables) or $PARSESV (if RTE is enable for System Variables) file-tailoring job.</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WCONFIG(KCIJPCNV)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJPMCF</td>
<td>Merge an old version of a customer override CONFIG profile member ($CFG$USR, converted RTE Batch mode member, or a new member named after the RTE name, and cloned from $CFG$USR) into a refreshed copy.</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WCONFIG(KCIJPMCF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJPVAL</td>
<td>Validate parameter value settings in customer override CONFIG profile members. This is a standalone job version. Same function is already performed in the $PARSE or $PARSESV VALIDATE step.</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WCONFIG(KCIJPVAL)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
<th>Required Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPPRS/</td>
<td>Process the PARMLIB samples from the interim (IK*) staging libraries into</td>
<td>Y</td>
</tr>
<tr>
<td>$PARSE</td>
<td>the corresponding work (WK*) output libraries. The $PARSE job performs the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>string substitutions and imbeds required by the user overrides in the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARMLIB configuration profile member. After completion of the $PARSE job,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>you have a complete set of customized runtime members in the work output</td>
<td></td>
</tr>
<tr>
<td></td>
<td>libraries (WKANCMDU, WKANPARU, WKANSAMU, WKD2PAR, WKD2PRF and WKD2SAM).</td>
<td></td>
</tr>
<tr>
<td>Note: Required if RTE is not enabled for System Variables (&quot;RTE_SYSV_SYSVAR_FLAG&quot; parameter in the CONFIG profile is set to &quot;N&quot;). If RTE_SYSV_SYSVAR_FLAG parameter is set to &quot;Y&quot;, then use the WCONFIG($PARSES) job instead.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Copy:</td>
<td>WCONFIG($PARSE)</td>
<td></td>
</tr>
</tbody>
</table>

---

**Note:** Required if RTE is not enabled for System Variables ("RTE_SYSV_SYSVAR_FLAG" parameter in the CONFIG profile is set to "N"). If RTE_SYSV_SYSVAR_FLAG parameter is set to "Y", then use the WCONFIG($PARSES) job instead. **User Copy: WCONFIG($PARSE)**
<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPPRV</td>
<td>Similar to KCIJPPRS/$PARSE job. If the RTE is enabled for System Variables, submit this job instead of $PARSE. $PARSESv does not include the &quot;PART 6 - KCIJPUP2 IEBUPDTE Steps&quot; of the $PARSE job. Latter function is split into a standalone KCIJPUPV job as KCIJPUPV job must be submitted in the LPAR for which the system variables referenced in the IEBUPDTE members were configured, for proper resolution at product startup. User Copy: WCONFIG($PARSESv)</td>
<td>Y</td>
</tr>
<tr>
<td>$PARSESv</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPPRy/PARSEExx</td>
<td>Similar to the $PARSE/$PARSESV jobs but these library-specific $PARSE jobs only perform the equivalent of &quot;PART 3 -- WK* Steps&quot; without the &quot;PART 1 -- VALIDATE Step&quot; &amp; &quot;PART 2 -- CPYEMPTY Steps&quot; of the typical $PARSE/$PARSESV jobs.</td>
</tr>
<tr>
<td></td>
<td>- KCIJPPRC/$PARSEC is a subset of the $PARSE job to process the PARMLIB samples from IKANCMDU into WKANCMDU work output library.</td>
</tr>
<tr>
<td></td>
<td>- KCIJPPRM/$PARSES is a subset of the $PARSE job to process the PARMLIB samples from IKANSAMU into WKANSAMU work output library.</td>
</tr>
<tr>
<td></td>
<td>- KCIJPPRP/$PARSEPR is a subset of the $PARSE job to process the PARMLIB samples from IKANPARU into WKANPARU work output library.</td>
</tr>
</tbody>
</table>
## PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function Description</th>
<th>Required Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPPRy/$PARSExx</td>
<td>Similar to the $PARSE/$PARSESV jobs but these library-specific $PARSE jobs only perform the equivalent of &quot;PART 3 - WK* Steps&quot; without the &quot;PART 1 - VALIDATE Step&quot; &amp; &quot;PART 2 - CPYEMPTY Steps&quot; of the typical $PARSE/$PARSESV jobs.</td>
<td>Y</td>
</tr>
</tbody>
</table>

*New in 4Q10*

`KCIJPPRY/$PARSEDV` is a standalone job that can be run to get a list of resolved values for KCIPARSE-extracted symbolics. The job provides for TYPE:CE (CHAR extracted) and TYPE:IE (INTEGER extracted) KCIPARSE-extracted symbolics for System Variables use in the PARMLIB CONFIG parameter values.

**User Copy: WCONFIG($PARSExx)**

(where y = C,M,P,Y  xx = CM, SM, PR. DV)
## PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJPSUB</td>
<td>Submit the composite KCIJPSUB master WKANSAMU</td>
<td>Y</td>
</tr>
</tbody>
</table>

PARMLIB auto-SUBMIT job instead of submitting the following jobs individually:

1. KCIJPALO composite runtime library allocation job
2. KCIJPLOD composite TK*->RK* runtime library load job
3. KCIJPSEC composite product security job
4. KCIJPUSP composite USS preparation job
5. KCIJPLNK composite ASM/LINK job
6. **KCIJPSYS composite system set-up job
7. **KCIJPUSSS composite USS create HFS system set-up job
8. **KCIJPCPY backup runtime libraries job
9. **KCIJPW2R WK*->RK* deployment job
10. KCIJPIVP configuration verification job

Notes:
- **Review the NOTES section of KCIJPSUB to see if certain jobs should be auto-submitted or not auto-submitted by KCIJPSUB.
- If RTE is enabled for System Variables support ("RTE_SYSV_SYSVAR_FLAG" parameter in the CONFIG profile is set to "Y"), submit the WKANSAMU(KCIJVSUB) instead.

**User Copy: WKANSAMU(KCIJPSUB)**
<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
<th>Required</th>
<th>Member</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJVSUB</td>
<td>Submit the composite KCIJPSUV master WKANSAMU Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARMLIB auto-SUBMIT job instead of submitting the following jobs individually:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. KCIJVALO composite runtime library allocation job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. KCIJVLOD composite TK*-&gt;RK* runtime library load job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. KCIJVSEC composite product security job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. KCIJVUSP composite USS preparation job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. KCIJVLNK composite ASM/LINK job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6. **KCIJVUPV composite System Variables IEBUPDTE job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. **KCIJVSYS composite system set-up job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. **KCIJVUSS composite USS create HFS system set-up job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. **KCIJVCPY backup runtime libraries job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. <em><em>KCIJVW2R WK</em>-&gt;RK</em> deployment job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11. KCIJVIVP configuration verification job</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
- Review the NOTES section of KCIJVSUB to see if certain jobs should be auto-submitted or not auto-submitted by KCIJVSUB.
- If RTE is not enabled for System Variables support ("RTE_SYSV_SYSVAR_FLAG" parameter in the CONFIG profile is set to "N"), submit the WKANSAMU(KCIJPSUB) instead.

**User Copy: WKANSAMU(KCIJVSUB)**
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function</th>
<th>Required Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJcALO</td>
<td>Allocate the RK* execution runtime libraries for all the products and components in the RTE. Notes: - Required if KCIJcSUB job is not submitted. - Where &quot;%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WKANSAMU(KCIJcALO)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJcLOD</td>
<td>Copy members of the products' SMP/E target libraries to the read-only RK* libraries. Notes: - Required if KCIJcSUB job is not submitted. - Where &quot;%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WKANSAMU(KCIJcLOD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJcSEC</td>
<td>Create security-related members (load modules, encryption key, and other elements) based on the product security requirements. Notes: - Required if the product-specific IBM-supplied security exit or input needs to be customized. - Required if KCIJcSUB job is not submitted. - Where &quot;%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td>N</td>
</tr>
<tr>
<td>User Copy: WKANSAMU(KCIJcSEC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member</td>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>KCIJcUSP</td>
<td>Create the USS-related members in the RKANDTVU RTE library for use in the composite KCIJP USS job. See companion KCIJP USS job. Notes: Required if configuring certain products only (WebSphere Message Broker, ITCAM for SOA, and ITCAM for Application Diagnostics). Required if KCIJcSUB job is not submitted. Where &quot;$%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User Copy: WKANSAMU(KCIJcUSP)</td>
<td></td>
</tr>
<tr>
<td>KCIJcLNK</td>
<td>Assemble/link elements into the SYSLMOD RKANMOD* load library. Notes: Required if configuring certain products only (OMEGAMON XE for Mainframe Networks) Required if the common KOBVTPL OBVTM1 exit for OMNIMON Base needs to be customized. Required if KCIJcSUB job is not submitted. Where &quot;$%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User Copy: WKANSAMU(KCIJcLNK)</td>
<td></td>
</tr>
<tr>
<td>Member Name</td>
<td>Function</td>
<td>Required</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>KCIJcUPV</td>
<td>Submit the composite KCIJPUPV System Variables IEBUPDTE job. This job populates variable-named members contained in the application-specific KppJPUPB composite IEBUPDTE members in the WK* work output libraries. This job is equivalent to &quot;PART 5 - KCIJPUP2 IEBUPDTE Steps&quot; of the WCONFIG($PARSE) job. The function of the KCIJPUPV job is split out from &quot;PART 5 - KCIJPUP2 IEBUPDTE Steps&quot; of the $PARSE job as KCIJPUPV job must be submitted in the LPAR for which the system variables referenced in the IEBUPDTE members were configured, for proper resolution at product startup. Notes:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Required if RTE is enabled for System Variables support (&quot;RTE_SYSV_SYSVAR_FLAG&quot; parameter in the CONFIG profile is set to &quot;Y&quot;).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- KCIJPSUV (job submitted by KCIJVSUB auto-SUBMIT job) SUBMITs the KCIJcUPV job by default. Edit the KCIJPSUV job accordingly.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Required for submission in the target LPAR where the symbolics are resolved.</td>
<td></td>
</tr>
</tbody>
</table>

User Copy: WKANSAMU(KCIJcUPV)
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Name</th>
<th>Function</th>
<th>Required</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJcSYS</td>
<td>Complete system-related set-up functions:</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- copy the started tasks, VTAM major node members, and health check elements for the products and components into system libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- assemble/link product modules into system libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Requires write access to system libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- KCIJPSUB/KCIJPSUV auto-SUBMIT jobs comment out the SUBMIT command for KCIJcSYS by default. Edit the KCIJPSUB/KCIJPSUV jobs accordingly.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Where &quot;%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User Copy: WKANSAMU(KCIJcSYS)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJcUSS</td>
<td>Create the HFS directories and sub-directories to copy files to HFS. KCIJPUSP pre-processor job is required prior to submitting KCIJPUSS. KCIJPUSP job is split out from this composite KCIJPUSS job as KCIJPUSS job must be submitted on a machine that has access to the USS directories and the TSO userid that submits it must have write access to the HFS directories.</td>
</tr>
</tbody>
</table>

**Notes:**

- Required if configuring certain products only (WebSphere Message Broker, ITCAM for SOA, and ITCAM for Application Diagnostics).

- KCIJPSUB/KCIJPSUV auto-SUBMIT jobs comment out the SUBMIT command for KCIJcUSS by default. Edit the KCIJPSUB/KCIJPSUV jobs accordingly.

- Where "%" = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)

**User Copy:** WKANSAMU(KCIJcUSS)
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Function</th>
<th>Required</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJcIVP</td>
<td>Verify that all the required runtime datasets members, and configuration jobs for this RTE were created, and that the jobs were executed successfully. Note: Required if KCIJcSUB job is not submitted. User Copy: WKANSAMU(KCIJcIVP)</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Member Name</td>
<td>Function</td>
<td>Required</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>KCIJcCPY</td>
<td>Backup the existing production RK* runtime libraries and the PARMLIB work libraries (IK* interim staging libraries and WK* work output libraries).</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- See companion KCIJcW2R job.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Where &quot;%' = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCIJcW2R</td>
<td>Empty the current RK* production runtime user libraries and copy the runtime members created by $PARSE in the WK* work output libraries, into the RK* production runtime user libraries. Note: Following your normal change control process, copy the WK* work output libraries to the respective production RK* runtime libraries. If you elect to run this job, first run the KCIJPcpy job to backup the RK* libraries. Verify that all RK* libraries were backed up successfully before running this job.</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- See companion KCIJcCPY job.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Where &quot;%' = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PARMLIB KCIJP* Batch Jobs (cont'd)

<table>
<thead>
<tr>
<th>Member</th>
<th>Required</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCIJcDEL</td>
<td>N</td>
<td>Delete the RK* execution runtime libraries for all the products and components in the RTE.</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td>Where &quot;%&quot; = P (if RTE_SYSV_SYSVAR_FLAG=N) or V (if RTE_SYSV_SYSVAR_FLAG=Y)</td>
</tr>
<tr>
<td>User Copy:</td>
<td></td>
<td>WKANSAMU(KCIJcDEL)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Required</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxSTRT</td>
<td>N</td>
<td>Composite list of /S (START) Started Tasks (where xxxx = %RTE_STC_PREFIX%)</td>
</tr>
<tr>
<td>User Copy:</td>
<td></td>
<td>WKANSAMU(xxxxSTRT)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Required</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxSTOP</td>
<td>N</td>
<td>Composite list of /P (STOP) Started Tasks (where xxxx = %RTE_STC_PREFIX%)</td>
</tr>
<tr>
<td>User Copy:</td>
<td></td>
<td>WKANSAMU(xxxxSTOP)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Required</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxAPF</td>
<td>N</td>
<td>Composite list of APF-authorized libraries (where xxxx = %RTE_STC_PREFIX%)</td>
</tr>
<tr>
<td>User Copy:</td>
<td></td>
<td>WKANSAMU(xxxxAPF)</td>
</tr>
</tbody>
</table>

*************** Bottom of Data ***************
PARMLIB Documentation
PARMLIB Documentation

URL: http://www-01.ibm.com/support/docview.wss?uid=swg21417935

Master PARMLIB Technote
THANK YOU

FOR YOUR TIME!

Questions and/or Feedback

Cecile Day
dayce@us.ibm.com

SHARE
Technology, Connections, Results