Configuring the OMEGAMON Product Family: The PARMLIB Alternative Approach

Cecile C. Day (dayce@us.ibm.com)
z/OS Installation & Configuration Assistance Tool (ICAT/PARMLIB) Developer
IBM Corporation

Wednesday, August 4, 2010: 9:30 AM-10:30 AM ET
Room 107 (Hynes Convention Center)
Session 7627

Agenda

- Summary – What is PARMLIB?
- Installation of OMEGAMON via “ICAT”
- Installation via “PARMLIB”
- List of PARMLIB-enabled Products by Phase
- PARMLIB Phase 2 Details
  - *Details of PARMLIB Procedure*
  - *KCIJP* Batch Jobs*
  - *$PARSE Batch Job*
  - *$CFG* CONFIG User Profile*
  - *Parameter On-line Help*
  - *Parameter Validation*
  - *PARMLIB IVP*
- PARMLIB Documentation

Summary – What is PARMLIB?

- **PARMLIB:**
  - Alternative configuration method to the ICAT (a.k.a. z/OS Configuration Tool) for OMEGAMON/ITM-based products.
  - Rivals the ease and speed of configuring OMEGAMON applications through ICAT's "Batch Mode" configuration method.
  - Updates OMEGAMON runtime members according to a PARMLIB CONFIG profile pre-defined by the customer as that JCL is copied from SMP/e targets to work output libraries equivalent to the production runtime environment (RTE) libraries, so customers are able to stage changes based on their schedules.
  - As of Phase 2, it is as easy to use as submitting ***10-14 "RTE-centric" composite batch jobs***. It is an intuitive process that is simpler than learning to use a the current ICAT tool.
  - Customers can use PARMLIB, or until they are ready to use this new process, they can continue to use ICAT. ICAT is ***still the primary mode*** of configuration in the current GA releases of z/OS ITM V6.2.2 and OMEGAMON XE V420s, OMEGAMON XE for Messaging V701, etc.
  - Is being delivered in phases throughout ***2010 – 2011**** to maximize the opportunity for customers to provide their input and influence the design.

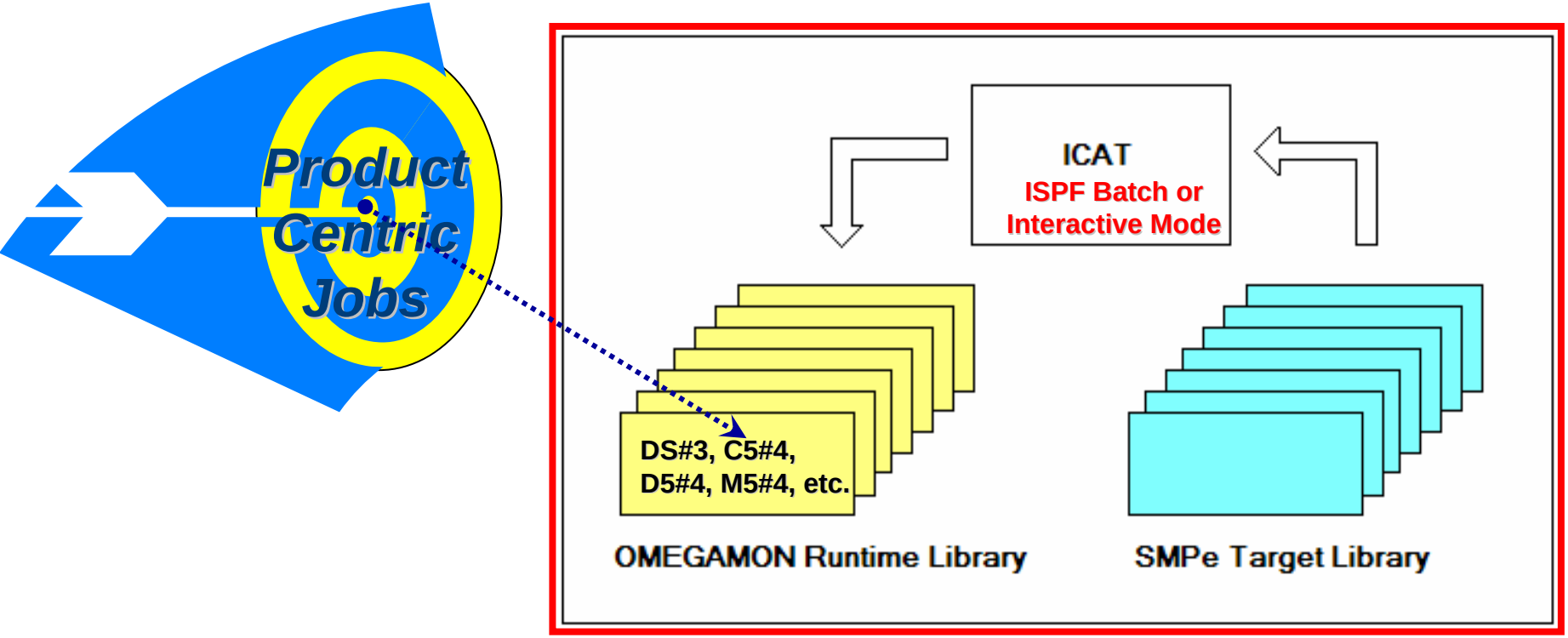
** Any statements regarding IBM future direction and intent are subject to change or withdrawal, and represents goals and objectives only.*

Summary – What is PARMLIB?, continued

- **CAVEATS*:**
  - PARMLIB mode of maintaining RTEs in a mixed environment is not supported if you have products in the RTE that are enabled for PARMLIB and products that are not enabled for PARMLIB support yet*.
  - For current phases, 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) as there are major enhancements implemented in PARMLIB mode (e.g., RTE centric) that are not supported in ICAT mode.
  - Current PARMLIB phases do not contain system variable support yet*.

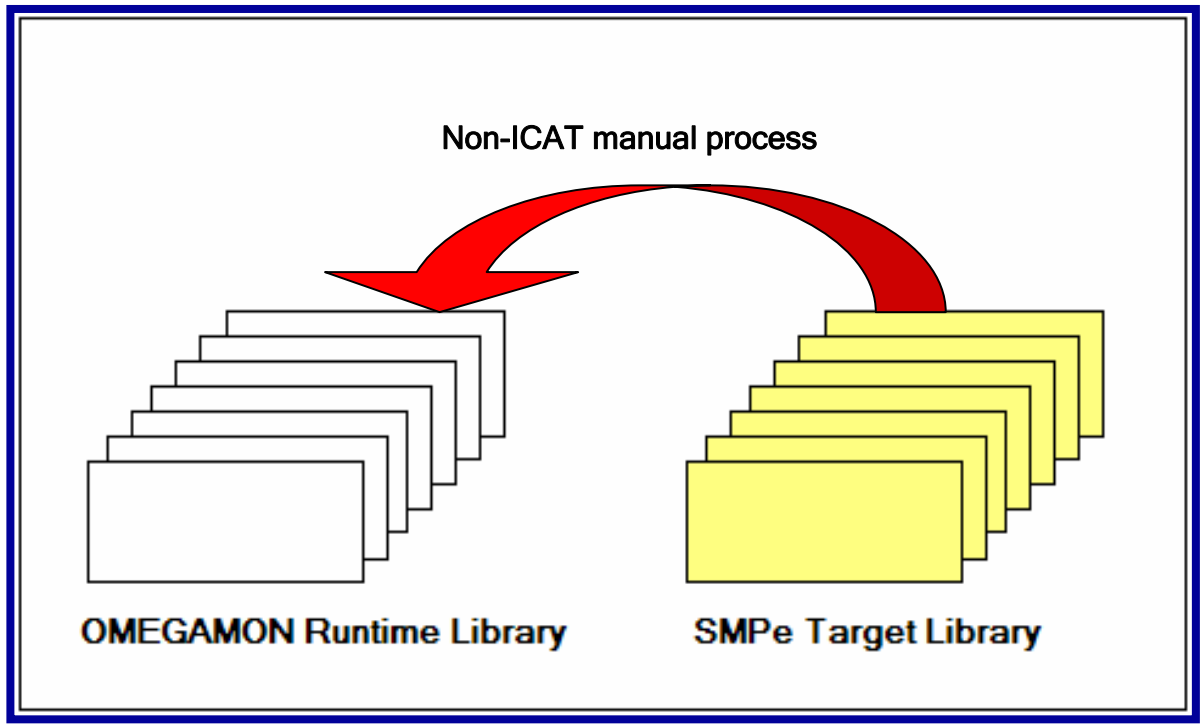
Installation of OMEGAMON via "ICAT" (Product-centric)

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.



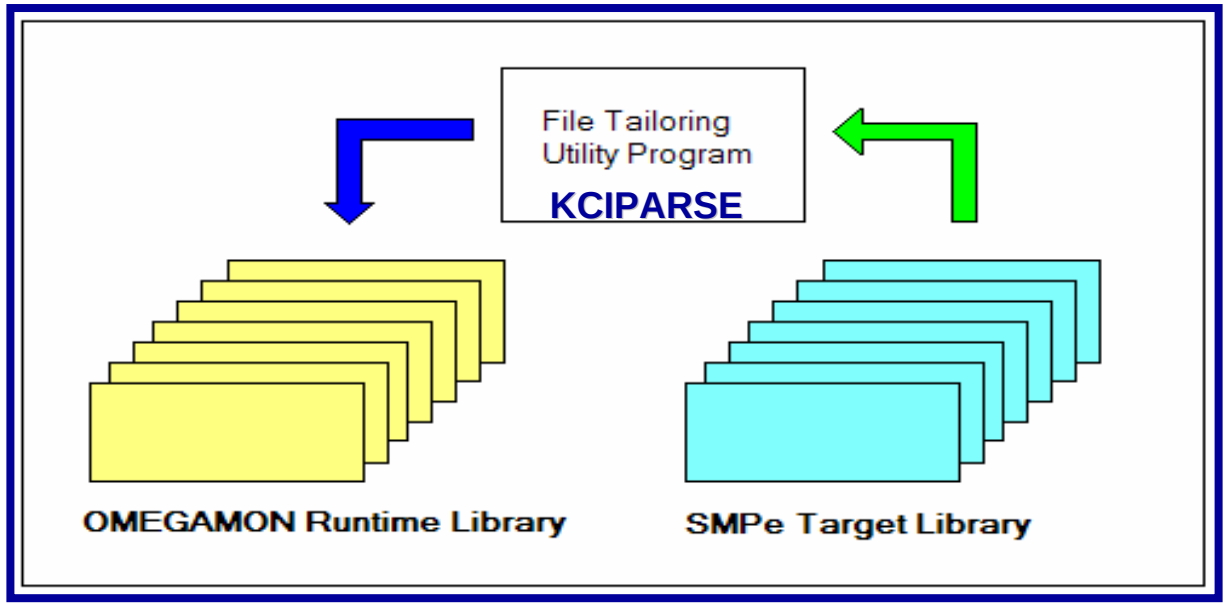
Installation via “PARMLIB” (RTE-centric)

1. With other IBM products, customers are used to manipulating parameter values directly in SYS1.PARMLIB or another PDS.
2. In the OMEGAMON version, the “PARMLIB” approach delivers an SMP/e-controlled equivalent of ICAT output (already built). The individual PDS members would require manual file tailoring, and manual propagation into a runtime PDS library.

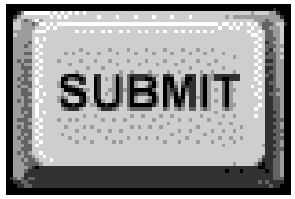


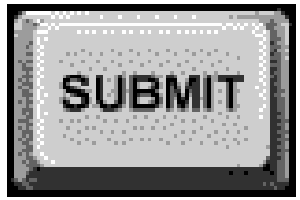
Installation via “PARMLIB” (RTE-centric), continued

1. There is no way we could expect customers to duplicate manually all of the customized values ICAT provides for the OMEGAMON/ITM suite of products.
2. Therefore, a new configuration utility, called KCIPARSE, was created to assist with this hands-on (manual) task.
3. It exhibits IEBCOPY-like behavior to reduce learning curve.
4. It includes various parsing features such as Search, Replace, Imbed, Conditional IF/THEN logic, and much more.
5. SMP/E-controlled files are fully customized based on a PARMLIB CONFIG user profile as they are copied to the equivalent of the production runtime libraries.



Product-centric (ICAT) vs. RTE-centric jobs (PARMLIB)





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

ICAT vs. PARMLIB

10-14 PARMLIB RTE-centric jobs to configure components for 1 LPAR RTE regardless how many products!

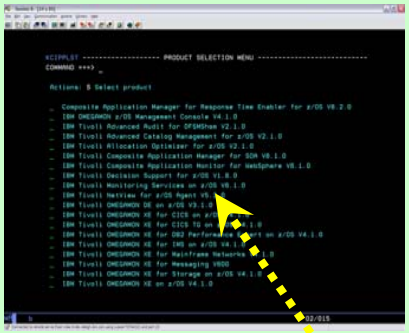
List of PARMLIB-enabled Products by Phase

z/OS Product Families that Use the ICAT Today

SHARE
Technology • Connections • Results

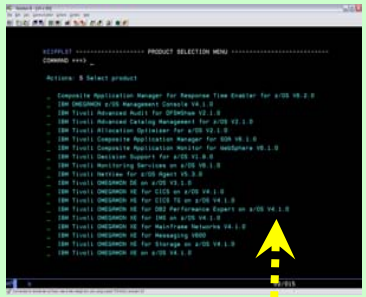
z/OS TMS family

TEMS



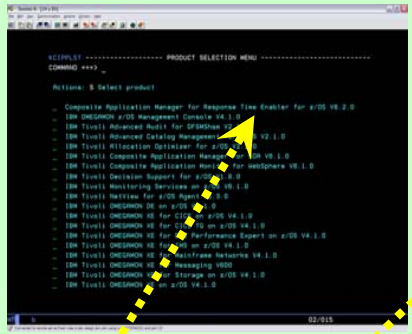
OMEGAMON family

z/OS, CICS, DB2, IMS, Storage, OMEGAVIEW, Management Console, Mainframe Networks, Messaging



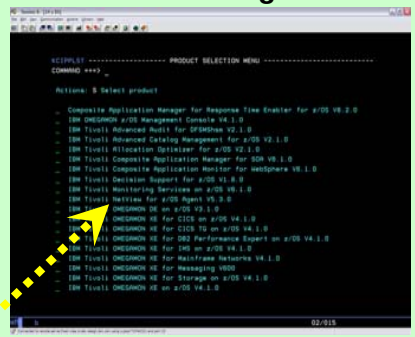
z/OS ITCAM family

SOA, WebSphere (Appl. Diagnostics), File Transfer Enabler



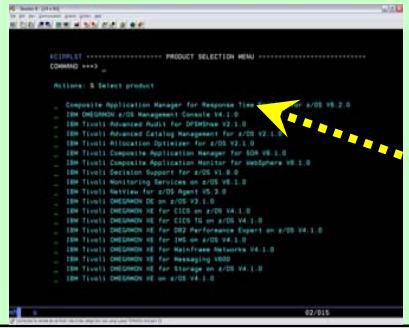
NetView family

NetView for z/OS Agent

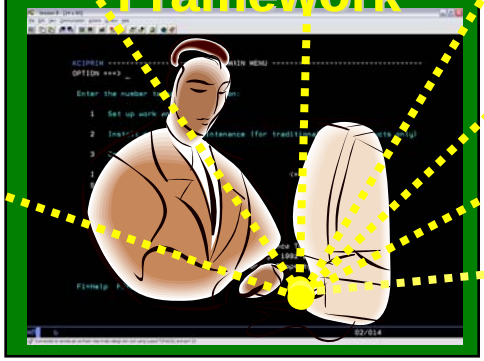


Rocket family

Advanced Audit for DFSMShsm, Advanced Catalog Management, Allocation Optimizer, Advanced Reporting, Advanced Backup & Recovery, Automated Tape Allocation Manager, Tape Optimizer

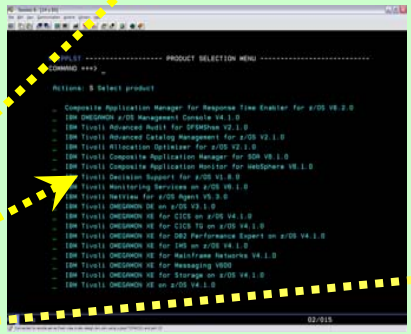


ICAT Configuration Framework



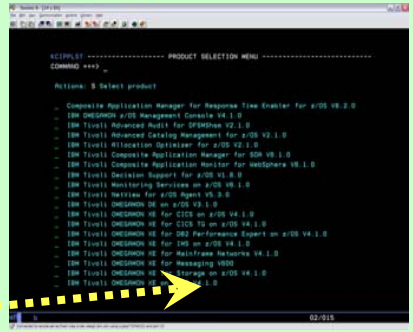
TDS family

Tivoli Decision Support Agent



System Automation family

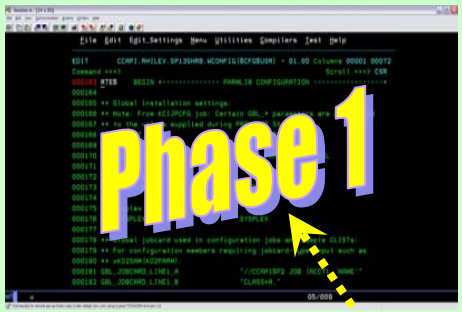
System Automation for z/OS Agent



Planned Products for PARMLIB Enablement

z/OS TMS family

TEMS



OMEGAMON family

z/OS, CICS, DB2, IMS, Storage, OMEGAVIEW, OMEGAVIEW II Management Console, Mainframe Networks, Messaging

z/OS ITCAM family

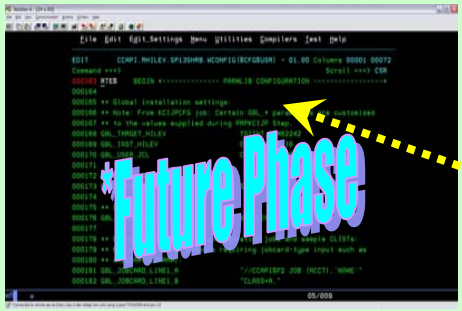
SOA, WebSphere (Appl. Diagnostics), File Transfer Enabler

NetView family

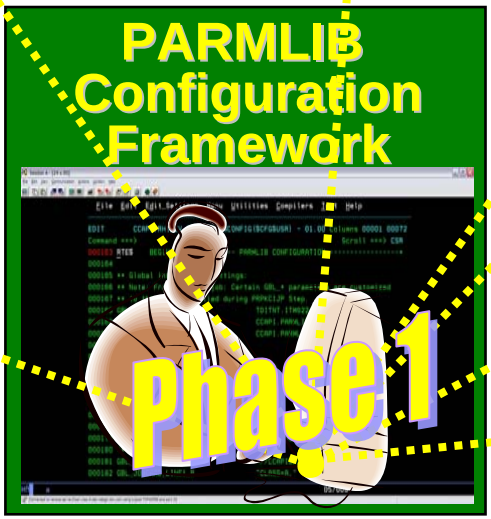
NetView for z/OS Agent

Rocket family

Advanced Audit for DFSMShsm, Advanced Catalog Management, Allocation Optimizer, Advanced Reporting, Advanced Backup & Recovery, Automated Tape Allocation Manager, Tape Optimizer

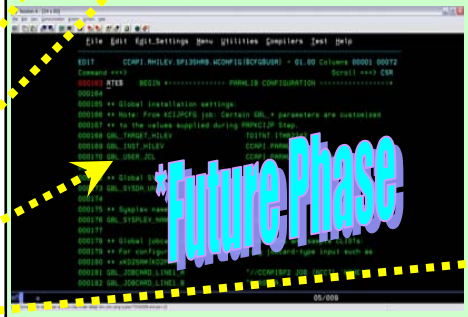


PARMLIB Configuration Framework



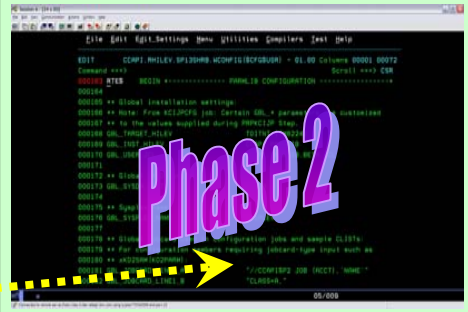
TDS family

Tivoli Decision Support Agent



System Automation family

System Automation for z/OS Agent



Pilot OMEGAMON Products for PARMLIB Enablement (Phase 1)

Pilot OMEGAMON Products for PARMLIB Enablement

The following products are enabled for **PARMLIB - Phase 1 support via HKCI310 PTF UA52371**.

A. Product-specific components and base versions supported:

- **IBM Tivoli OMEGAMON XE for CICS on z/OS V4.2.0**
  - Component: OMEGAMON II for CICS (KOC/KC2)
  - Component: OMEGAMON XE for CICS on z/OS (KC5)
  - Component: OMEGAMON XE for CICS TG on z/OS (KGW)
- **IBM Tivoli OMEGAMON XE for DB2 Performance Expert/Monitor V4.2.0**
  - Component: OMEGAMON XE for DB2 PE/PM (KO2/KD2)
  - Component: OMEGAMON XE for DB2 PE/PM Agent (KD5)
- **IBM Tivoli OMEGAMON XE for IMS on z/OS V4.2.0**
  - Component: OMEGAMON II for IMS (KOI/KI2)
  - Component: OMEGAMON XE for IMS on z/OS (KI5)
- **IBM Tivoli OMEGAMON XE on z/OS V4.2.0**
  - Component: OMEGAMON II for MVS (KOM/KM2)
  - Component: OMEGAMON XE on z/OS (KM5)

Pilot OMEGAMON Products for PARMLIB Enablement

B. Common infrastructure components and base versions supported:

- **IBM Tivoli Management Services on z/OS V6.2.2**
  - Component: Tivoli Enterprise Monitoring Server (KDS)
  - Component: Common Agent configuration framework (KAG)
  - Component: Common Persistent Datastore (PDS) configuration framework (KPD)
- **OMNIMON Base V6.2.0**
  - Component: OMEGAMON Subsystem (KCN/KOB)
- **End-to-End V6.2.0**
  - Component: End-to-End (KET)

Available Products for PARMLIB Enablement (Phase 2)

Available Products for PARMLIB Enablement (Phase 2)

The remaining OMEGAMON products are also enabled for **PARMLIB - Phase 2 support via HKCI310 PTF UA53118**. Phase 2 enablement also includes the System Automation Monitoring Agent (KAH) & the NetView for z/OS Agent (KNA).

- **IBM Tivoli OMEGAMON DE on z/OS V4.2.0**
  - Component: OMEGAVIEW (KMV)
  - Component: OMEGAVIEW II for the Enterprise (KWO)
- **IBM Tivoli OMEGAMON XE for Storage on z/OS V4.2.0**
  - Component: OMEGAMON II for SMS (KDF)
  - Component: OMEGAMON XE for Storage on z/OS (KS3)
- **IBM OMEGAMON z/OS Management Console V4.1.0**
  - Component: OMEGAMON z/OS Management Console Agent (KHL)
- **IBM Tivoli OMEGAMON XE for Mainframe Networks V4.2.0**
  - Component: OMEGAMON II for Mainframe Networks (KON)
  - Component: OMEGAMON XE for Mainframe Networks (KN3)
- **IBM Tivoli NetView for z/OS Agent V5.4.0**
  - Component: NetView for z/OS Agent (KNA)
- **IBM Tivoli System Automation for z/OS V3.1.0**
  - Component: System Automation Monitoring Agent (KAH)

Available Products for PARMLIB Enablement (Phase 2)

Phase 2 enablement also includes the ITCAM File Transfer Enabler component which became part of ITM622 z/OS TEMS.

- **IBM Tivoli OMEGAMON XE for Messaging on z/OS V7.0.1**
  - Component: OMEGAMON XE for WebSphere MQ Configuration (KMC)
  - Component: OMEGAMON XE for WebSphere MQ Monitoring (KMQ)
  - Component: OMEGAMON XE for WebSphere Message Broker Monitoring (KQI)
- **ITCAM for Transactions, File Transfer Enabler for z/OS V7.1.0**
  - Component: File Transfer Enabler for z/OS Agent (KT1)

PARMLIB Phase 1 and Phase 2 Scope and Schedule

PARMLIB Phase 1 and Phase 2 Scope and Schedule

Base Contents:

- Provide PARMLIB samples, KCIJP* batch jobs and KCIPARSE 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.
- 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**: APAR#: OA30575 (CI) for PTF **HKCI310 UA52371** (GA End of Feb. 2010)
- **Phase 2**: APAR#: OA32122 (CI) for PTF **HKCI310 UA53118** (GA End of May 2010)

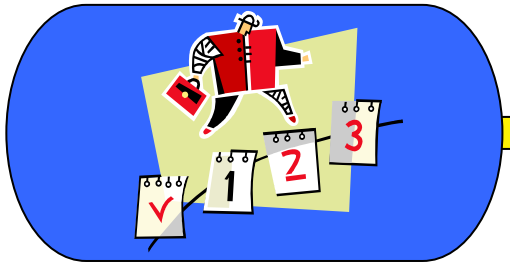
Documentation:

- 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***" Phase 2
- New "***Chapter 15. Using the PARMLIB method to set parameter values***" Phase 2 in the Tivoli® OMEGAMON XE and Tivoli Management Services on z/OS V6.2.2 Common Planning and Configuration Guide

High-level Details of PARMLIB Procedure and Data Flow – Diagrams

PARMLIB Procedure - Diagram

The following **Diagram 1** through **Diagram 4** show a high-level overview of the steps involved in configuring the products 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.



Apply PARMLIB Phase 2 HKCI310 PTF UA53118 (GA)

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

KCIJPCFG set-up job in USER JCL

(Optional) Customize JOBCARD macro & copy **JOBCARD / KCIRPLBS** on-line help set-up macro from WCONFIG to SYSPROC concatenation

Customize sample $JOBCARD in WCONFIG

A

Diagram 1

PARMLIB Procedure – Diagram (cont'd)

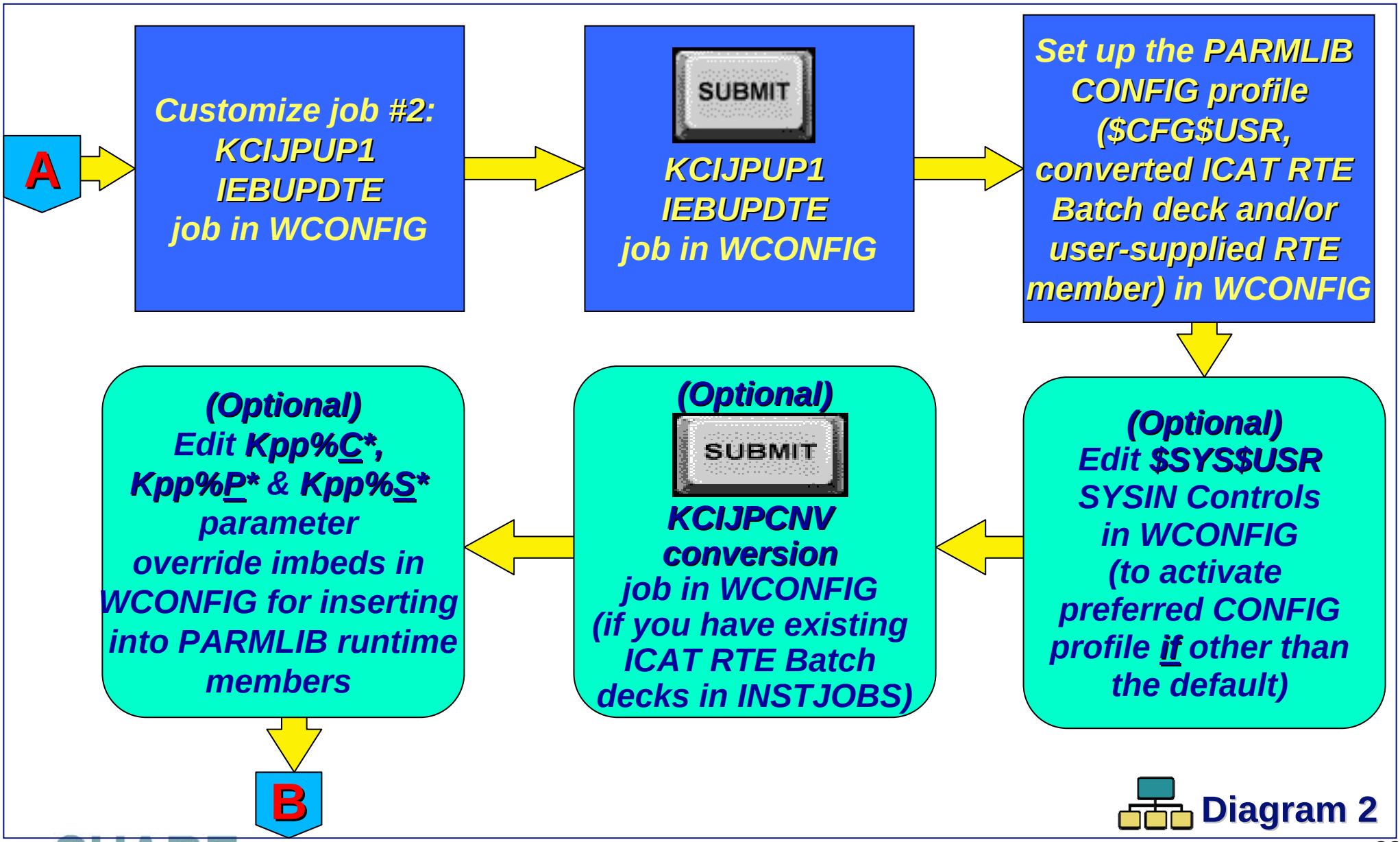
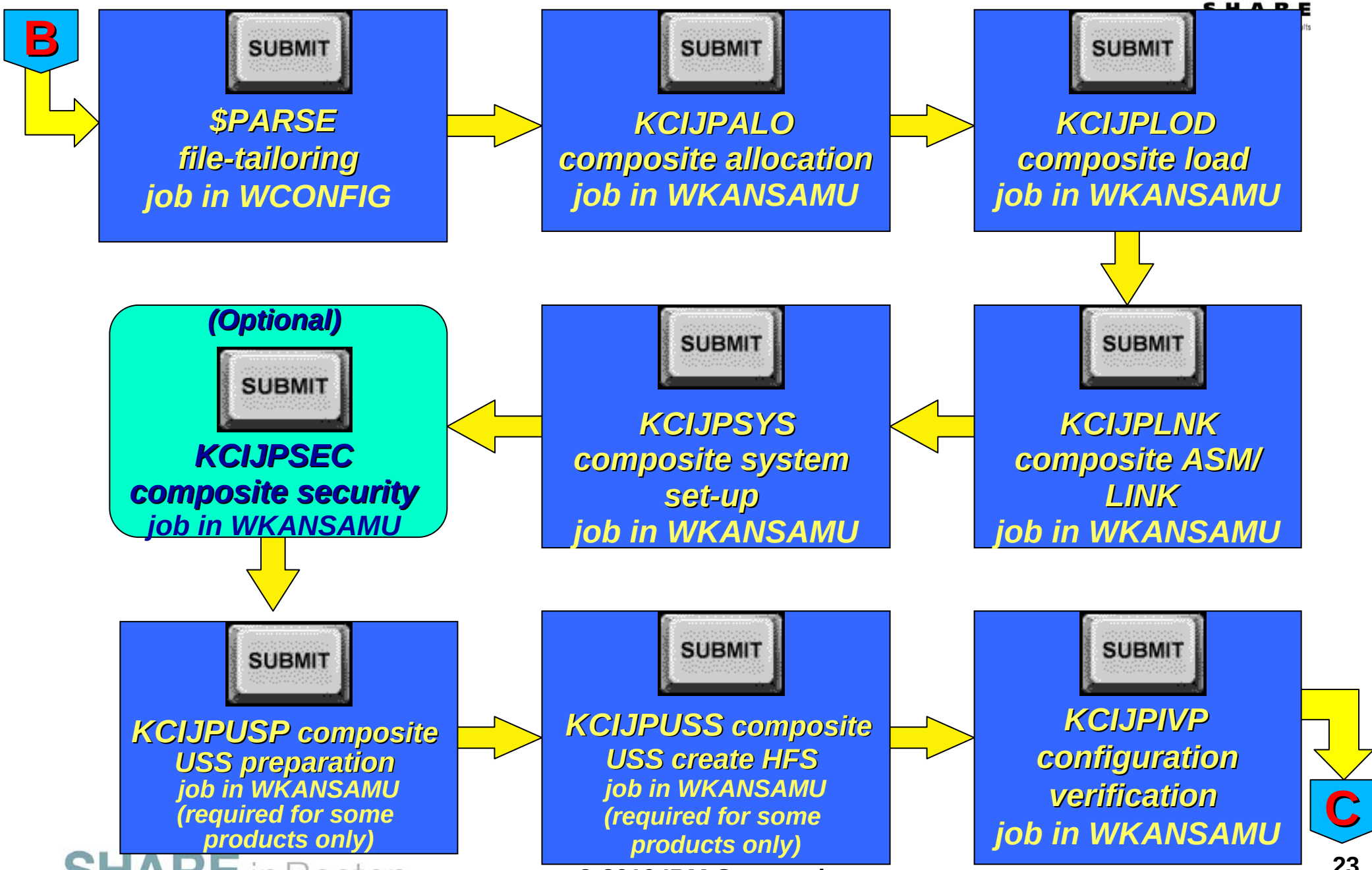


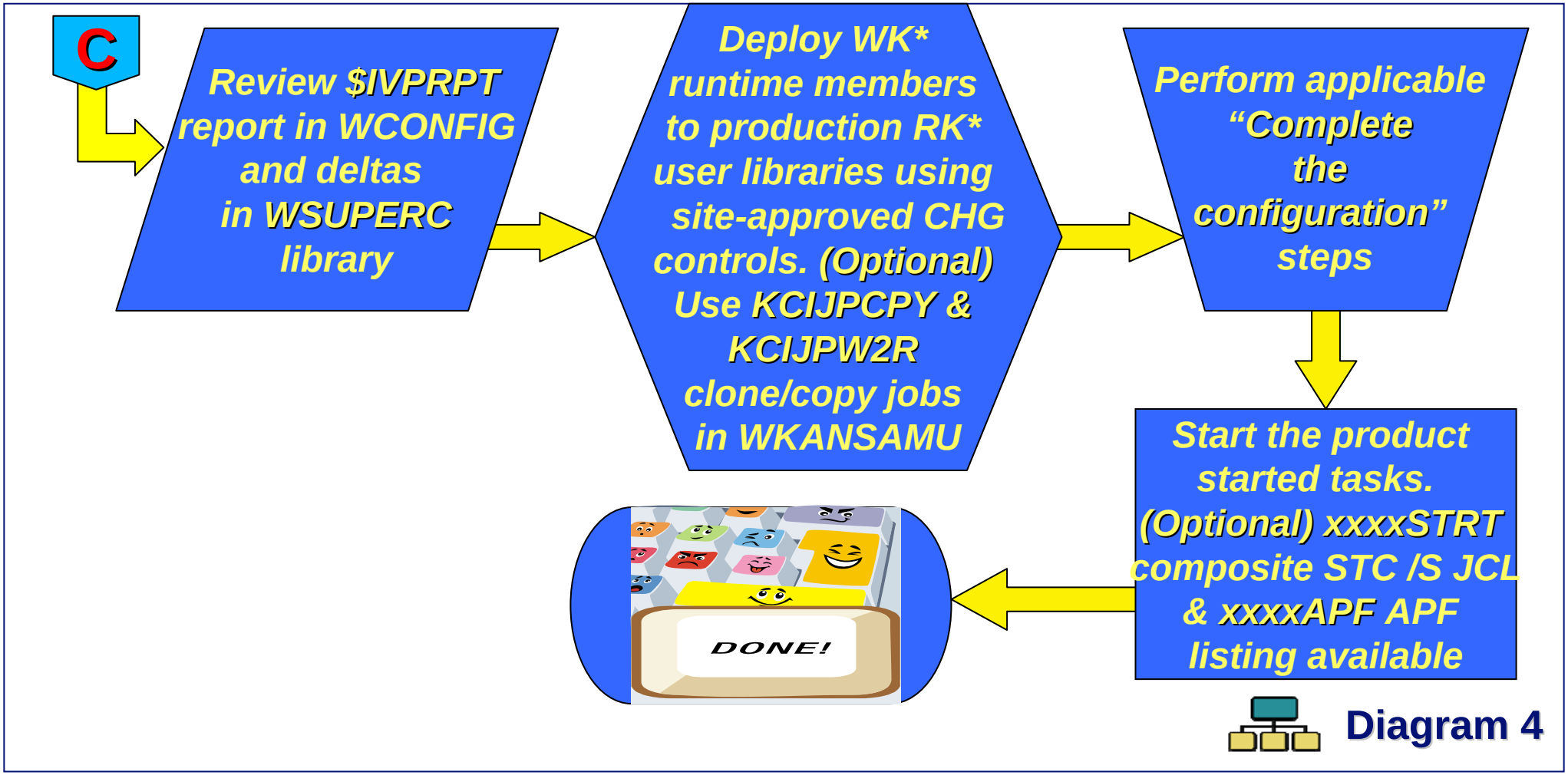
Diagram 2

PARMLIB Procedure – Diagram (cont'd)

Diagram 3



PARMLIB Procedure – Diagram (cont'd)



Same procedure whether you are configuring 1 component or 37 components!

PARMLIB KCIJ$NDX Batch Job Index

PARMLIB KCIJ$NDX* Batch Job Index

```
================================================================
 * * *  P A R M L I B   B a t c h  J o b s  I n d e x  * * *
================================================================
In the IKANSAMU/WKANSAMU library, a PARMLIB job index (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:
========  ========================================= ========
Member                                               Required
Name                        Function                   Y/N?
========  ========================================= ========
KCIJPCFG  Set up the PARMLIB work libraries and configuration    Y
          elements for the runtime environment (RTE):
          - allocate the &rhilev.&rte.WCONFIG PARMLIB work
            control library (for the PARMLIB control members).
          - 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.
          - copy/rename applicable PARMLIB control members
            from the SMP/E targets to PARMLIB WCONFIG.
          - prepare KCIJPUP1 IEBUPDTE job.
          User Copy: Copy from TKANSAM to a user JCL library.
                     A copy is created in WCONFIG.
```

Lists all KCIJP* PARMLIB jobs

PARMLIB KCIJP* Batch Jobs – ICAT Cross-reference

KCIJPCFG job – ICAT Cross-reference

```
CCAPI.PARMLIB.GBL.CONFIG(KCIJPCFG)
.
.
* ************************************************************
* USER SECTION: CONFIGURE_PRODUCTS                           *
* ---------------------- BEGIN - USER SECTION: CONFIG ------- *
* ************************************************************
* CONFIGURE FLAGS: Set to "Y" or "N".
* ************************************
* Tivoli Enterprise Monitoring Server: KDS fla
  SET CONFIGURE_TEMS_KDS       = "Y"
* IBM Tivoli OMEGAMON XE for CICS on z/OS: KC5
  SET CONFIGURE_CICS_KC5       = "Y"
* IBM Tivoli OMEGAMON XE for CICS TG on z/OS:
  SET CONFIGURE_CICS_TG_KGW    = "Y"
* IBM Tivoli OMEGAMON XE for DB2 PE/PM: KD2 an
  SET CONFIGURE_DB2_PEPM_KD2   = "Y"
  SET CONFIGURE_DB2_AGENT_KD5 = "Y"
* IBM Tivoli OMEGAMON XE for IMS on z/OS: KI5
  SET CONFIGURE_IMS_KI5        = "Y"
* IBM Tivoli OMEGAMON XE on z/OS: KM5 flag
  SET CONFIGURE_ZOS_KM5        = "Y"
* IBM Tivoli OMEGAMON XE for Messaging - WebSp
  SET CONFIGURE_MESSAGING_KMC = "Y"
* IBM Tivoli OMEGAMON XE for Messaging - WebSp
  SET CONFIGURE_MESSAGING_KMQ = "Y"
.
.
```

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.

KCIJPPRS/$PARSE job – ICAT Cross-reference

```
CCAPI.ZCAC.CIDSSYSG.WCONFIG($PARSE)

CCAPI.PARMLIB.INSTJOBS(CB#RCIDSSYSG)
********************************************************************************
*     SECTION 3: JOBS SORTED BY GENERATION SEQUENCE                            *
********************************************************************************
GEN   MEMBER   JOB                                    JOB
SEQ   NAME     DESCRIPTION                            NOTES
---   -------- ------------------------------- ------------------------
3     DS#3CIDSSYSG CREATE RUNTIME MBRS
10    C2#3CIDSSYSG CREATE RUNTIME MBRS
16    C5#3CIDSSYSG CREATE RUNTIME MBRS, AGT ADRSP
19    DF#3CIDSSYSG CREATE RUNTIME MBRS
.
69    MV#3CIDSSYSG CREATE RUNTIME MBRS
71    M2#3CIDSSYSG CREATE RUNTIME MBRS
.
7     AH#4CIDSSYSG REGISTER PRODUCT TO THE TEMS
.
35    D5#4CIDSSYSG REGISTER PRODUCT TO THE TEMS
42    GW#4CIDSSYSG REGISTER PRODUCT TO THE TEMS
46    HL#4CIDSSYSG REGISTER PRODUCT TO THE TEMS
54    I5#4CIDSSYSG REGISTER PRODUCT TO THE TEMS
75    M5#ICIDSSYSG CREATE RUNTIME MBRS, AGT TEMS
126   S3#ICIDSSYSG CREATE RUNTIME MBRS, AGT TEMS
26    D2#XCIDSSYSG CREATE PROFILE MEMBERS
27    D2#6CIDSSYSG INSTALL DB2 SSID RELATED MBRS
4     PD#PCIDSSYSG CREATE PDS MBRS
.
67    MQ#PCIDSSYSG CREATE PDS MBRS
127   S3#PCIDSSYSG CREATE PDS MBRS
```

Sample ICAT Batch CICATB job report

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

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPALO)
=============================================================
%IMBED% DDNAME:MEMBER       *ICAT CROSS-REFERENCE/COMMENTS
=============================================================
%IMBED%=INPUT1:???JPAL1          *pp#1 RTE Build job
%$IMBED_KDS_KDSJPAL3_INPUT1%     *DS#1 RTE Build job
%$IMBED_KCI_KCIJPAL2_INPUT1%     *pp#1 RTE Build job
%$IMBED_KCI_KCIJPAL3_INPUT1%     *pp#1 RTE Build job
%$IMBED_KC5_KC5JPAL2_INPUT1%     *C5#1 RTE Build job
%$IMBED_KD5_KD5JPAL2_INPUT1%     *D5#1 RTE Build job
%$IMBED_KI5_KI5JPAL2_INPUT1%     *I5#1 RTE Build job
%$IMBED_KM5_KM5JPAL2_INPUT1%     *M5#1 RTE Build job
%$IMBED_KOB_KOBJPAL2_INPUT1%     *pp#1 RTE Build job
%IMBED%=INPUT1:???JPAL4          *pp#4 TEMS registra
                                  for RKCP*/RKCF* VS
%IMBED%=INPUT1:???JPAL5          *pp#5 Allocate add'
                                  Epilog VSAM
%IMBED%=INPUT1:???JPALX          *D2#X ALLOCDS
%IMBED%=INPUT2:KC2JPA*           *C2#5 Allocate add'
                                  KC2##JPA template
%IMBED%=INPUT2:KC2JPH*           *C2#H RKC2HIST hist
                                  using KC2##JPH
%IMBED%=INPUT2:KI2JPA*           *I2#5 Allocate add'l using
                                  KI2##JPA template
%IMBED%=INPUT1:???JPALQ          *pp#Q Persistent Datastore
                                  KppAL* jobs
```

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

```
CCAPI.ZCAC.CIDSSYSG.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
 .
 .
%IMBED%=INPUT1:???JPLDX          *Special exceptions steps
```

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

KCIJPSEC job – ICAT Cross-reference

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPSEC)
=============================================================
%IMBED% DDNAME:MEMBER          *ICAT CROSS-REFERENCE/COMMENTS
=============================================================
%IMBED%=INPUT1:???JPSCO        *pp#O xKANSAMU(KppSUPD) -
                                Classic command table
%$IMBED_KDS_KDSJPSC3_INPUT1%  *xKANSAMU(KDSDKAES)-
                                xKANPARU(KAES256) key
%$IMBED_KDS_KLVJPSC3_INPUT1%  *xKANSAMU(KLV@ASM) – KLVxxNEV
%$IMBED_KC5_KOCJPSC3_INPUT1%  *xKANSAMU(KOCJxxxx) - KOCAxxxx
%$IMBED_KD5_KO2JPSC3_INPUT1%  *xKANSAMU(KO2xxxxA) - KO2xxxxX
%$IMBED_KI5_KOIJPSC3_INPUT1%  *xKANSAMU(KOIxxxxA) - KOIUxxHK
%$IMBED_KM5_KOMJPSC3_INPUT1%  *xKANSAMU(KOMxxxxA) - KOMxxxxX
```

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

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPSYS)
=============================================================
%IMBED% DDNAME:MEMBER       *ICAT CROSS-REFERENCE/COMMENTS
=============================================================
%IMBED%=INPUT1:KCIJPSYN     *CB#N Copy nodes to VTAMLST;
                             xKANSAMU(KCISYNJB)
%IMBED%=INPUT1:KCIJPSYP     *CB#P Copy STCs to PROCLIB;
                             xKANSAMU(KCISYPJB)
%IMBED%=INPUT1:KCIJPSYH     *CB#K APF/STC Health Checks;
                             xKANSAMU(KCIHCKJB)
%IMBED%=INPUT1:KDSJPSYL     *DS#L KDSMTAB1 to VTAMLIB;
                             xKANSAMU(KDSLNKJB)
```

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.

KCIJPLNK job – ICAT Cross-reference

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPLNK)
=============================================================
%IMBED% DDNAME:MEMBER       *ICAT CROSS-REFERENCE/COMMENTS
=============================================================
%IMBED%=INPUT1:KOBJPLK3     *xKANSAMU(KOBVTPLX) job for
                             KOBVTPL module
%IMBED%=INPUT1:KONJPLK3     *xKANSAMU(KONLINK) job for
                             KONACTCS module
%IMBED%=INPUT1:KN3JPLK3     *xKANSAMU(KN3LINK) job for
                             KN3ACTCS/KN3ANMON
```

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

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPUSP)
=======================================
%IMBED% DDNAME:MEMBER      *ICAT CROSS-REFERENCE/COMMENTS
=======================================

%IMBED%=INPUT2:???JPUS6    *pp#6 job to create RKANDATV
                            members for USS
```

New in Phase 2

```
CCAPI.ZCAC.CIDSSYSG.IKANSAMU(KCIJPUSS)
=======================================
%IMBED% DDNAME:MEMBER      *ICAT CROSS-REFERENCE/COMMENTS
=======================================

%IMBED%=INPUT2:???JPUSU    *pp#U SBPXEXEC job for USS
```

New in Phase 2

Think of KCIJPUSP/KCIJPUSS as 1 job that performs the equivalent of running ICAT's pp#6/pp#U jobs for products that have USS requirements

PARMLIB $PARSE Batch Job

PARMLIB $PARSE Batch JCL

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*):

CCAPI.ZCAC.CIDSSYSG.WCONFIG($PARSE)

```
****************************** Top of Data ***************************
.
//*************************************************************
//* WKANPARU Step: IKANPARU-->WKANPARU
//* Process the PARMLIB members from IKANPARU to WKANPARU based on $CFG*
//* CONFIG and/or converted PARMLIB RTE Batch deck profile settings.
//*************************************************************
//WKANPARU EXEC PGM=KCIPARSE,COND=(4,LT,VALIDATE),
// PARM='MV=32000,MAXL=32000,ML=500,MI=255,MS=20000,LV=Y'
//STEPLIB  DD DISP=SHR,
//         DSN=%GBL_TARGET_HILEV%.TKANMOD
//INPUT1   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.IKANPARU
//INPUT2   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU
//WCONFIG  DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WCONFIG
//SYSUT2   DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WKANPARU
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DISP=SHR,
//         DSN=%RTE_HILEV%.%RTE_NAME%.WCONFIG($SYS$USR)
.
```

INDDx input DDNAMEs can be up to 5 libraries if needed.

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_TEMS_TCP_HOST%;
00034 AFTER  IP.PIPE:&SYSIPHOSTNAME.;\

00035 BEFORE IP.PIPE:%KC5_TEMS_BKUP1_TCP_HOST%;
00035 AFTER  IP.PIPE:&AGT_BKUP1_TEMS_TCP_HOST.;\

00039 BEFORE %KC5_TEMS_VTAM_NETID%.\
00039 AFTER  &SYSVTAMNETID..\

00040 BEFORE %KC5_TEMS_VTAM_APPL_LLB_BROKER%.\
00040 AFTER  K&SYSCLONE.DSLB.\

00041 BEFORE %KC5_TEMS_VTAM_LU62_DLOGMOD%.SNASOCKETS;
00041 AFTER  CANCTDCS.SNASOCKETS;\

WRITE MEMBER KC5ENV    RECORDS: 00102
```

PARMLIB CONFIG Profile parameter

PARMLIB CONFIG Profile value

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**:

```
EDIT       CCAPI.ZCAC.CIDSSYSG.WCONFIG(CIDSSYSG)
Command ===>
** Values that describe the TEMS to which the agen
KC5_TEMS_LOCAL_CONNECT_FLAG          Y
KC5_TEMS_NAME_NODEID                 "CIDS&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 CONFIG Profile parameter

PARMLIB CONFIG Profile value

PARMLIB CONFIG User Profile

PARMLIB CONFIG User Profile

The **$CFG* CONFIG User Profile** allows the customer to override defaults as necessary. Sample CFGUSR composite CONFIG member is shown below:

CCAPI.ZCAC.CIDSSYSG.WCONFIG(CFGUSR)

```
****************************** Top of Data **********
.
.
RTE$    BEGIN *-------------- PARMLIB CONFIGURATION -----------------*

** Global installation settings:
** Note: From KCIJPCFG job: Certain GBL_* parameters are customized
** to the values supplied during PRPKCIJP Step.
GBL_TARGET_HILEV                    TDITNT.ITM62242
GBL_INST_HILEV                      CCAPI.PARMLIB
GBL_USER_JCL                        CCAPI.PARMLIB.GBL.CONFIG

** Sysplex name:
GBL_SYSPLEX_NAME                    &SYSPLEX.

** Runtime environment (RTE) settings:
RTE_NAME                            CIDSSYSG
RTE_DESCRIPTION                     "CIDSSYSG LPAR"
RTE_TYPE                            FULL          * FULL, SHARING or BASE *
** RTE global defaults:
RTE_HILEV                           CCAPI.ZCAC
RTE_VSAM_HILEV                      CCAPI.ZCAC
.
```

Values you customized in KCIJPCFG job

XCSF, @HUB, @RTE, etc.

PS0601.OMEGA621

PS0601.OMEGA621

PARMLIB CONFIG User Profile (cont'd)

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 (CIDSSYSG is the name of the RTE Batch Member):

```
CCAPI.ZCAC.CIDSSYSG.WCONFIG(CIDSSYSG)
****************************** Top of Data **********
*
* File created on 28 July 2010 at 23:08:19  by KCIRPLBC
* Input file was 'CCAPI.PARMLIB.INSTJOBS(CIDSSYSG)'
*
RTE$ BEGIN  *------------ CONFIGURATION TOOL
RTE_DESCRIPTION                    "CIDSSYSG PARMLIB Share SYSGBASE RTE"
RTE_TYPE                           SHARING          * FULL, SHARING or BASE

** If RTE_TYPE is SHARING:
RTE_SHARE                          SYSGBASE
**
** RTE global defaults:
RTE_HILEV                          CCAPI.ZCAC
RTE_VSAM_HILEV                     CCAPI.ZCAC
.
.
RTE_TEMS_NAME_NODEID               "CIDSSYSG:CMS"

** Security options:
RTE_SECURITY_USER_LOGON            RACF
RTE_SECURITY_FOLD_PASSWORD_FLAG    Y
** (Optional) If RTE_SECURITY_USER_LOGON is ACF2:
**GBL_DSN_ACF2_MACLIB                SYS1.ACF2.MACLIB
.
** System procedure libraries:
GBL_DSN_SYS1_PROCLIB               SYS1.PROCLIB
GBL_DSN_SYS1_VTAMLST               SYS1.VTAMLST
```

Clearer parameter names

Clearer parameter names

Clearer parameter names

PARMLIB CONFIG User Profile (cont'd)

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:

CCAPI.ZCAC.CIDSSYSG.WCONFIG(CIDSSYSG)

```
.
GBL_DSN_SYS1_VTAMLIB              SYS1.VTAMLIB
GBL_DSN_SYS1_SISTMAC1             SYS1.SISTMAC1
GBL_DSN_SYS1_MODGEN               SYS1.MODGEN
GBL_DSN_SYS1_MACLIB               SYS1.MACLIB
GBL_DSN_SYS1_PARMLIB              SYS1.PARMLIB
GBL_DSN_SYS1_LINKLIB              SYS1.LINKLIB
GBL_DSN_SYS1_SBLSCLI0             SYS1.SBLSCLI0
GBL_DSN_SYS1_SAXREXEC             SYS1.SAXREXEC
GBL_DSN_SYS1_SBPXEXEC             SYS1.SBPXEXEC
GBL_DSN_SYS1_BRODCAST             SYS1.BRODCAST
.
.
GBL_DSN_TCP_SYSTCPD_TCPDATA       TCPIP.SEZAINST
GBL_DSN_CICS_CTG_DLL              SYS1.SCTGDLL
GBL_DSN_NETVIEW_CNMLINK           NETVIEW.V5R4M0.CNMLINK
GBL_DSN_CEE_SCEERUN               SYS1.CEE.SCEERUN
GBL_DSN_CEE_SCEELKED              SYS1.CEE.SCEELKED
GBL_DSN_CSF_SCSFMOD0              CSF.SCSFMOD0
GBL_DSN_IMS_RESLIB                SYS1.IMS.RESLIB
GBL_DSN_IMS_SCEXLINK              SYS1.IMS.SCEXLINK
GBL_DSN_IMS_SFUNLINK              SYS1.IMS.SFUNLINK
GBL_DSN_WMQ_SCSQAUTH              IBM.WMQ.SCSQAUTH
GBL_DSN_WMQ_SCSQANLE              IBM.WMQ.SCSQANLE
GBL_DSN_WMQ_SCSQLOAD              IBM.WMQ.SCSQLOAD
GBL_DSN_DB2_LOADLIB_V8            SYS1.PP.DB2.V8R1.DSNLOAD
GBL_DSN_DB2_LOADLIB_V9            SYS1.PP.DB2.V9R1.DSNLOAD
GBL_DSN_DB2_RUNLIB_V8             SYS1.PP.DB2.V8R1.RUNLIB
GBL_DSN_DB2_RUNLIB_V9             SYS1.PP.DB2.V9R1.RUNLIB
GBL_DSN_DB2_DSNEXIT               SYS1.PP.DB2.DSNEXIT
.
```

In ICAT, these parameters are product-specific so if more than one component needs the same value, the DSNAME is specified more than once.

PARMLIB CONFIG User Profile (cont'd)

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

```
CCAPI.ZCAC.CIDSSYSG.WCONFIG($CFG$USR)
****************************** Top of Data **********
.
000135 * INSTRUCTIONS:
000140 * ************************************************************
000141 * SECTION: USER PROLOG (OPTIONAL)
000142 * ************************************************************
000143 *+---+-----------------------------------------+--------+----+*
000144 *|NO.| CHANGE DESCRIPTION                      |  DATE  | ID |*
000145 *+---+-----------------------------------------+--------+----+*
000146 *|@03| Override KC5_X_AGT_STORAGE_* LIMIT()/RESERVE()|01/20/10| CD |*
000147 *|@02| Set GBL_DSN_TCP_SYSTCPD_TCPDATA to new library|01/20/10| CD |*
000148 *|@01| Override K%%_X_STC_SYSTCPD_INCLUDE_FLAG=Y     |01/19/10| CD |*
000149 * ************************************************************
000150 .
000251 ** Additional OMEGAMON XE for CICS Agent settings:
000252 *KC5_X_AGT_STORAGE_LIMIT_EXTEND    22
000253 *KC5_X_AGT_STORAGE_LIMIT_PRIMARY   16
000254 *KC5_X_AGT_STORAGE_RESERVE_PRI     2048
000255 *KC5_X_AGT_STORAGE_RESERVE_EXT     2048
000256 KC5_X_AGT_STORAGE_LIMIT_EXTEND    23
000257 KC5_X_AGT_STORAGE_LIMIT_PRIMARY   20
000258 KC5_X_AGT_STORAGE_RESERVE_PRI     4096
000259 KC5_X_AGT_STORAGE_RESERVE_EXT     4096
```

In ICAT, these parameters Equate to hardcoded settings (not externalized on ICAT panels)

PARMLIB Parameter On-line Help

PARMLIB Parameter On-line Help

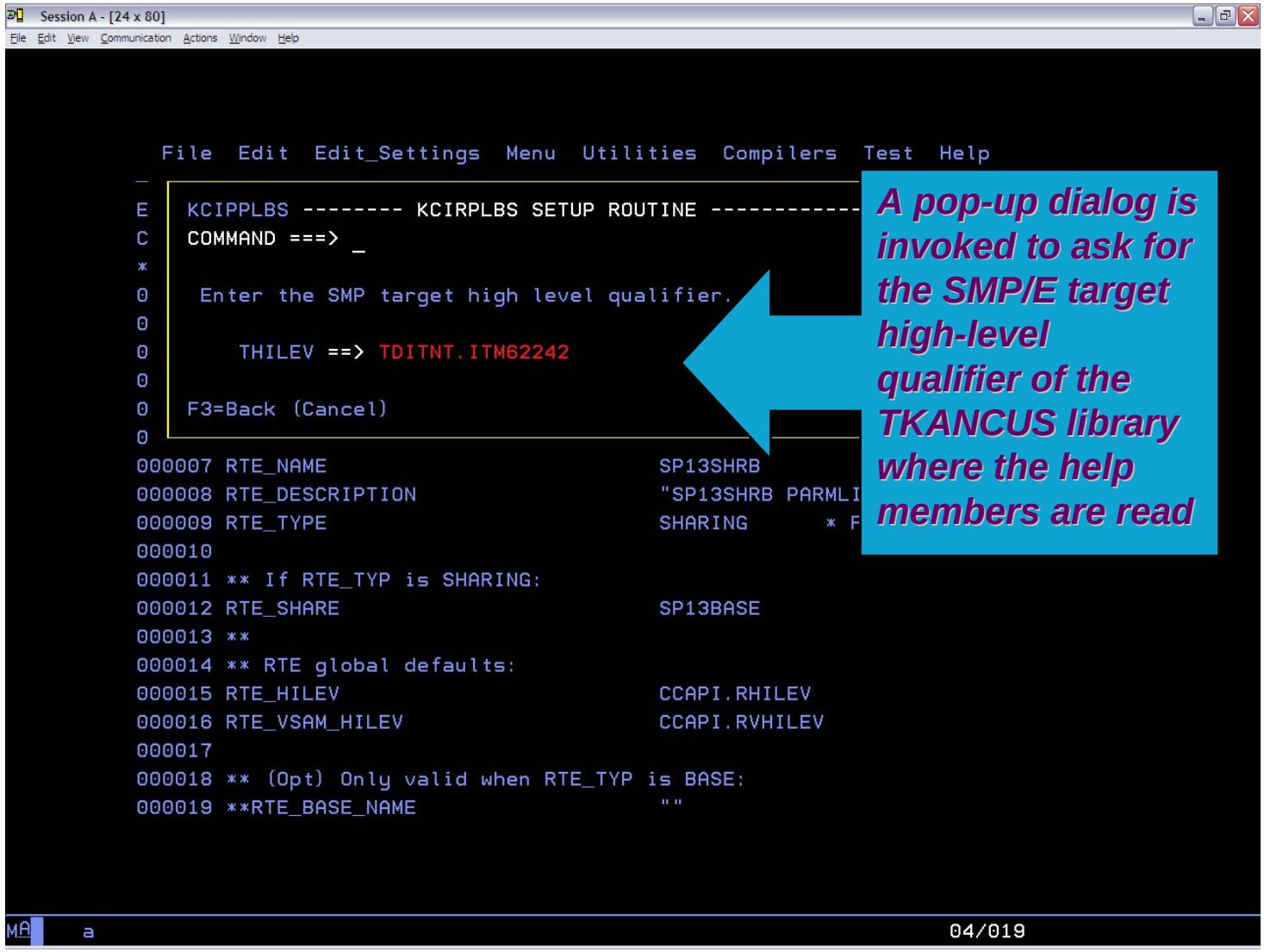
The KCIRPLBS on-line parameter help set-up macro copied from your SYSPROC concatenation is issued on the command line:

```
EDIT       CCAPI.ZCAC.CIDSSYSG.WCONFIG(CIDSSYSG)
Command ===> KCIRPLBS
****** ***************************** Top of Data ******************************
000001 *
000002 * File created on 19 Jan 2010 at 03:30:04 by KCIRPLBC
000003 * Input file was 'CCAPI.PARMLIB.INSTJOBS(CIDSSYSG)'
000004 *
000005
000006 RTE$ BEGIN *------------ CONFIGURATION TOOL V310 ---------------*
000007 RTE_NAME                    CIDSSYSG
000008 RTE_DESCRIPTION             "CIDSSYSG PARMLIB Share SYSGBASE RTE"
000009 RTE_TYPE                    SHARING      * FULL, SHARING or BASE *
000010
000011 ** If RTE_TYP is SHARING:
000012 RTE_SHARE                   SYSGBASE
000013 **
000014 ** RTE global defaults:
000015 RTE_HILEV                   CCAPI.ZCAC
000016 RTE_VSAM_HILEV              CCAPI.ZCAC
000017
000018 ** (Opt) Only valid when RTE_TYP is BASE:
000019 **RTE_BASE_NAME                      ""
.
.
```

PARMLIB on-line parameter help set-up macro

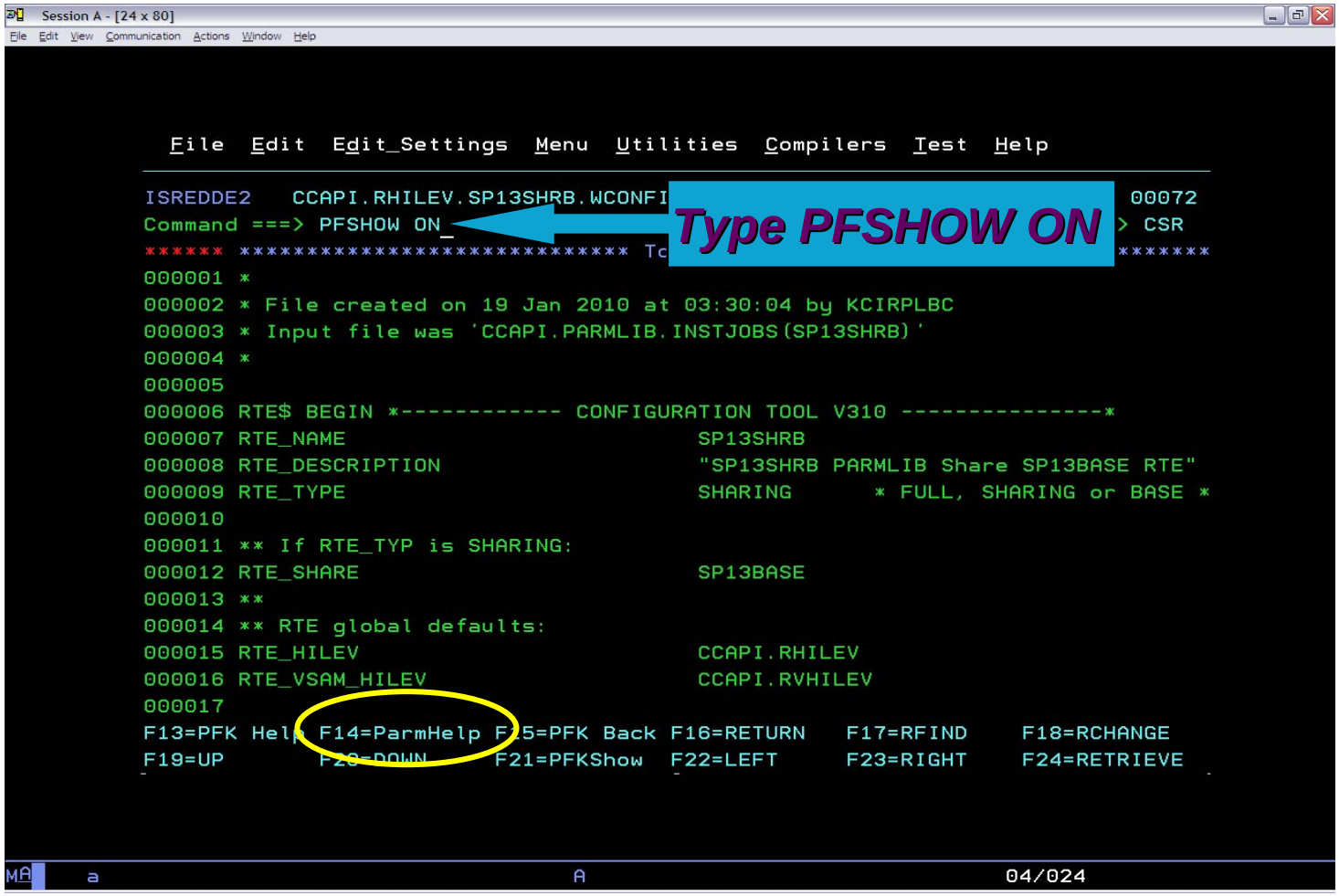
PARMLIB Parameter On-line Help (cont'd)

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:



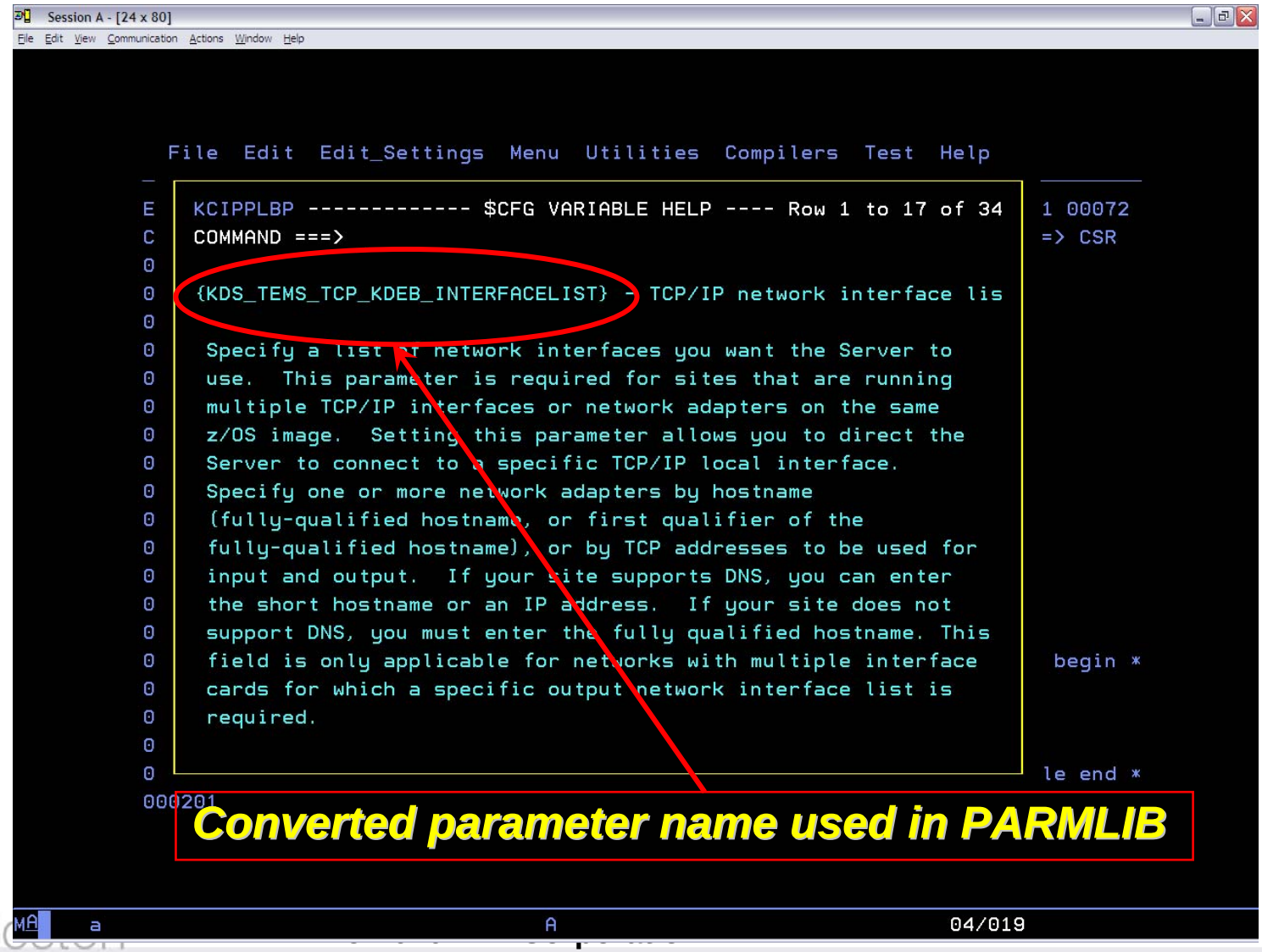
PARMLIB Parameter On-line Help (cont'd)

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.



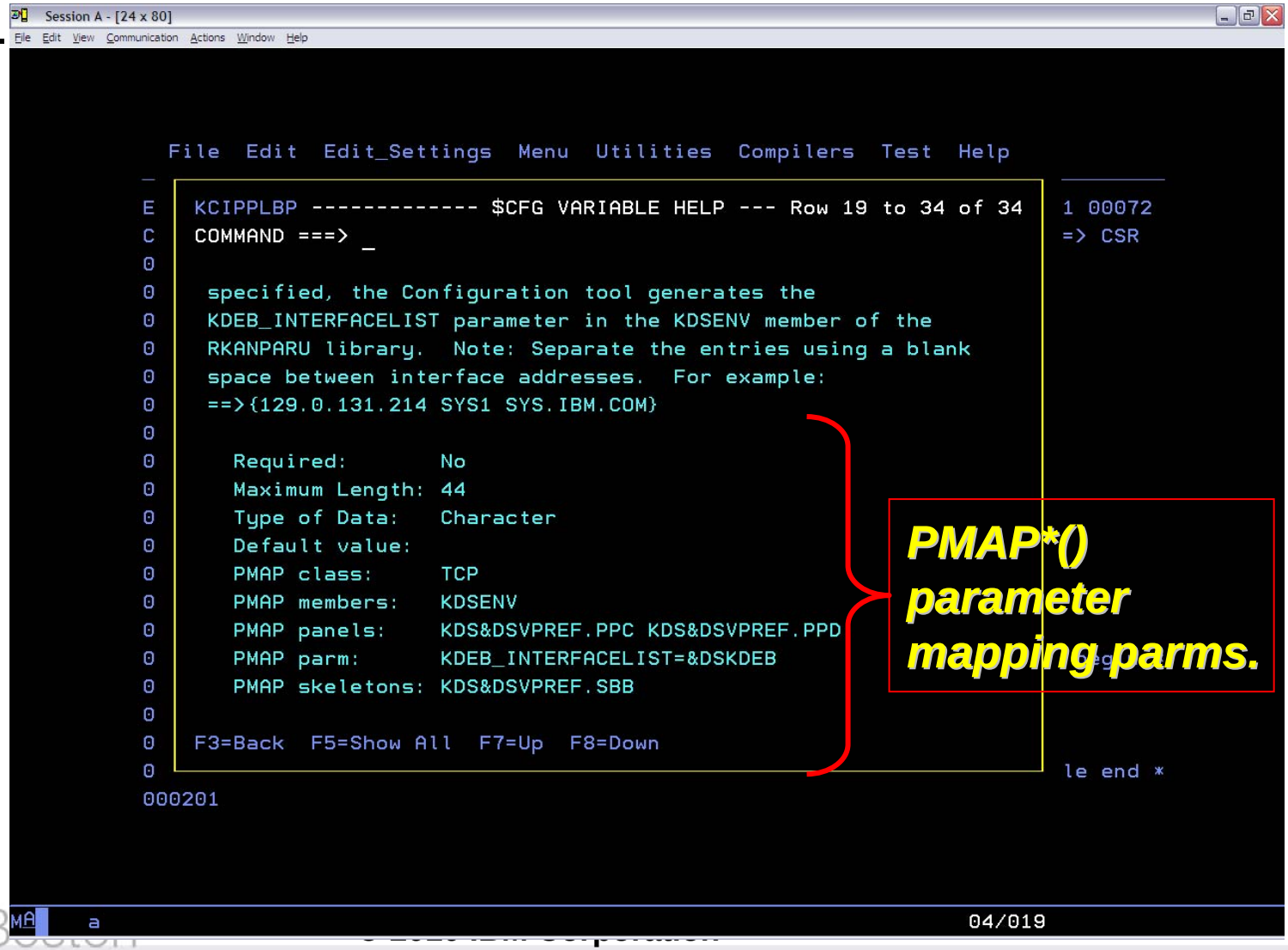
PARMLIB Parameter On-line Help (cont'd)

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



PARMLIB Parameter On-line Help (cont'd)

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



PARMLIB Parameter Validation

PARMLIB Parameter Validation Report

Generated automatically by $PARSE job or via WCONFIG(KCIJPVAL) standalone validation job

```
CCAPI.ZCAC.CIDSSYSG.WCONFIG($VALRPT)
| CONFIG Files: File# DSNAME
|                    1 CCAPI.ZCAC.CIDSSYSG.WCONFIG($CFG$IE
|                    2 CCAPI.ZCAC.CIDSSYSG.WCONFIG($CFG$USR)
|                    3 CCAPI.ZCAC.CIDSSYSG.WCONFIG(CIDSSYSG)
|
| 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                    |
+-------------------------------------------------------------+
| Parameter Name                  Parameter Value             |
| File#/Line#                     Error Line1                 |
|                                 Error Line2                 |
| ------------------------------ ----------------------------- |
| KDS_X_TEMS_CONFIRM_SHUTDOWN     100                         |
| 3/641                           Value must be <= 15.        |
|                                 100                         |
| 3/641                           Length must be <= 2.        |
|                                 ----------------------------- |
| KDS_X_TEMS_TASKS_ATTACHED_NUM   100                         |
| 3/643                           Length must be <= 1.        |
|                                 ----------------------------- |
| KDS_X_TEMS_WTO                  Z                           |
| 3/640                           Must be in list: Y,N.       |
```

PARMLIB
Installation/Configuration Verification (IVP)

PARMLIB IVP

CCAPI.ZCAC.CIDSSYSG.WCONFIG($IVPRPT) ← Generated by KCIJPIVP job

```
.
* 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
*********************************************************************
JOB      STATUS  JOBNAME  JOB#   DATE   TIME     HI-CC
-------- ------- -------- ------ ------ -------- -----
KCIJPCFG OK      CCAPICFG J04711 10.147 11:58:58 00000
KCIJPUP1 OK      CCAPI$SA J08747 10.148 15:26:58 00000
KCIJPCNV OK      CCAPI$SA J04746 10.147 12:08:01 00000
KCIJPMCF OPTION
KCIJPVAL OK      CCAPI$SA J12895 10.147 12:10:39 00000
$PARSE   OK      CCAPI$SA J19382 10.148 15:47:07 00000
KCIJPALO WARNING CCAPI$JP J06230 10.061 09:29:21 00002
KCIJPLOD ERROR   CCAPI$SA J19402 10.148 16:43:27 SE37
KCIJPSYS WARNING CCAPI$SY J06331 10.061 09:38:36 00004
KCIJPUSP OK      CCAPI$SA J19410 10.148 16:47:36 00000
KCIJPUSS OK      CCAPI$SA J19412 10.148 16:47:44 00000
KCIJPSEC OK      CCAPI$SC J06376 10.061 09:42:59 00000
KCIJPLNK ERROR   CCAPILNK J02126 10.139 08:50:03 00012
KCIJPCPY OK      CCAPI$JP J15302 10.063 12:00:05 00000
KCIJPW2R OK      CCAPIW2R J02160 10.139 09:25:07 00000
```

Review any STATUS=ERROR in the IVP report

PARMLIB IVP (cont'd)

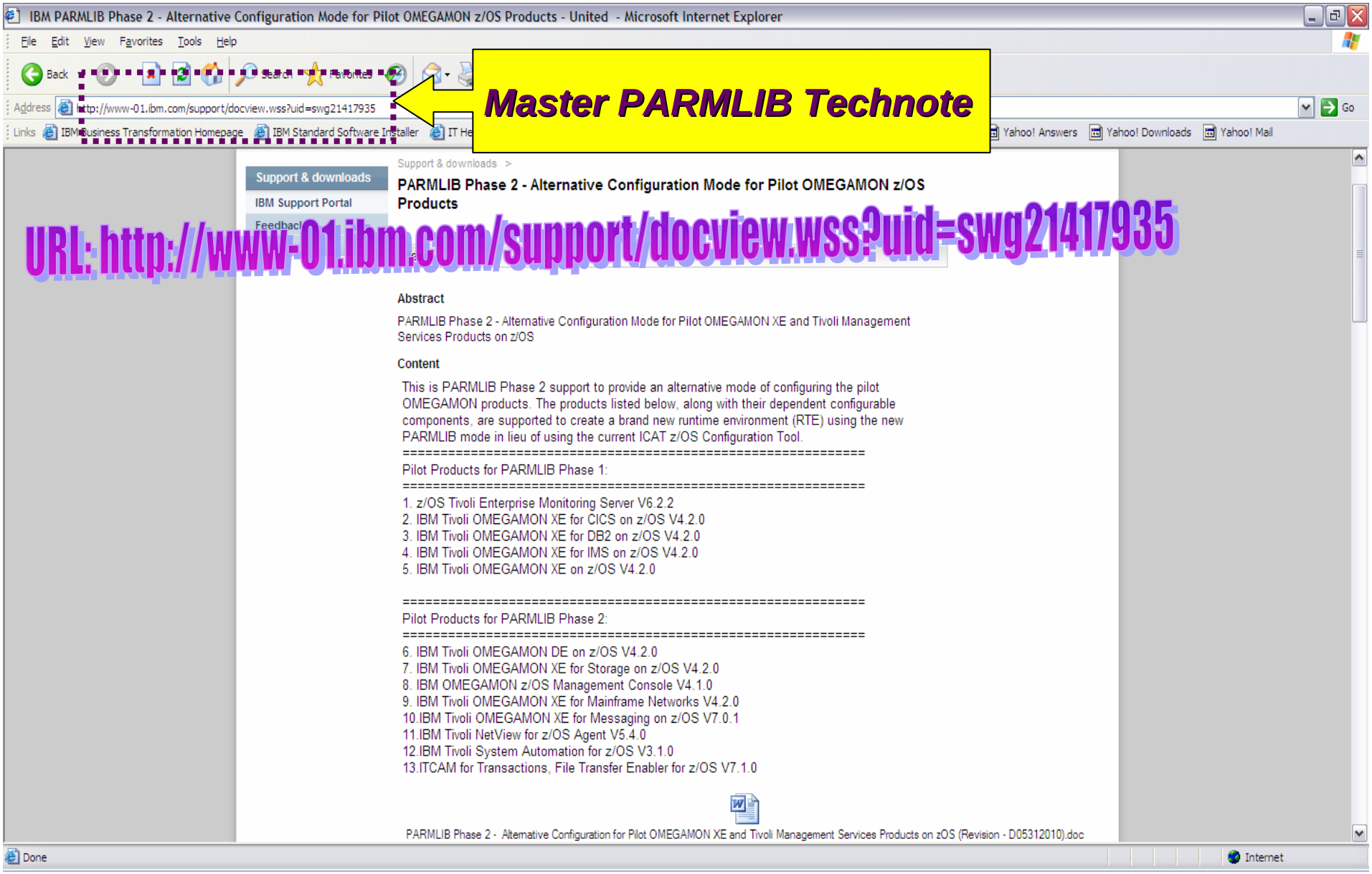
```
CCAPI.ZCAC.CIDSSYSG.WCONFIG($IVPRPT)

*************************************************************************
*   SECTION 2: REQUIRED SEQUENTIAL DATASETS
*************************************************************************
DSNAME                                          STATUS   JOBNAME
----------------------------------------------  -------
.
.
CCAPI.ZCAC.CIDSSYSG.RKM5LPR3                      OK
CCAPI.ZCAC.CIDSSYSG.RKM5LPR2                      OK
CCAPI.ZCAC.CIDSSYSG.RKM5LPR1                      OK
CCAPI.ZCAC.LPAR400J.RKM5PLX3                      OK       KCIJPALO
CCAPI.ZCAC.LPAR400J.RKM5PLX2                      OK       KCIJPALO
CCAPI.ZCAC.LPAR400J.RKM5PLX1                      OK       KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RNASGRP3                      MISSING  KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RNASGRP2                      MISSING  KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RNASGRP1                      MISSING  KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RKNAHIS3                      MISSING  KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RKNAHIS2                      MISSING  KCIJPALO
CCAPI.ZCAC.CIDSSYSG.RKNAHIS1                      MISSING  KCIJPALO
```

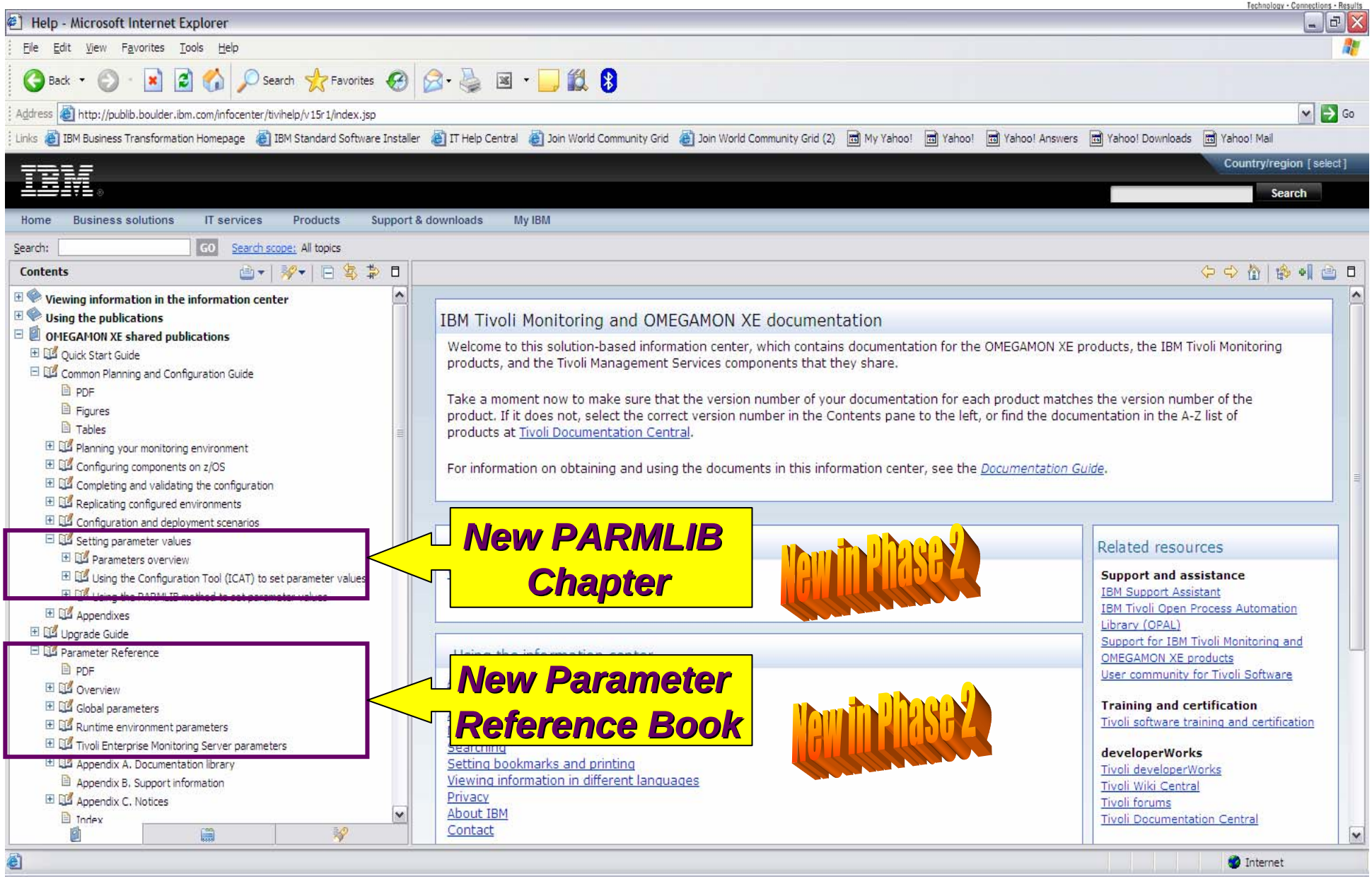
Check the offending job (KCIJPALO in this example) as to why the datasets are missing.

PARMLIB Documentation

PARMLIB Documentation



PARMLIB Documentation (cont'd)



SHARE
Technology • Connections • Results

FOR YOUR TIME!

Questions and/or Feedback

Cecile Day

dayce@us.ibm.com

PARMLIB
*** Live Demonstration ***

Sample PARMLIB Usage Scenarios

- 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

Sample PARMLIB Usage Scenarios (cont'd)

- 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 override the Agent failover ITM default of switching back to the original primary TEMS?
- A: See WCONFIG(KAG$PENV) to add CTIRA_PRIMARY_FALLBACK_INTERVAL=0
- Q: How do I generate a jobcard that is based on the jobname?
- A: See WCONFIG($JOBCARD) and customize using %SYSMEMBER% instead of hardcoded jobname.
- Q: If you refresh CFGIBM IBM Default CONFIG profile via maintenance, how do I sync-up my copy?
- A: See WCONFIG(KCIJPMCF) merge CONFIG profile job