





Check Processing Control System  
International MVS/ESA

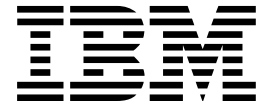


# Propagation of Adjustments Guide

*Release 1*



Check Processing Control System  
International MVS/ESA



# Propagation of Adjustments Guide

*Release 1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

**First Edition (December 1996)**

This edition applies to Release 1 Modification 0 of the IBM Check Processing Control System International MVS/ESA licensed program (Program No. 5799-FKT).

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult your IBM representative to be sure you have the latest edition and any Technical Newsletters.

IBM does not stock publications at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department 72L, MG83/204, 8501 IBM Drive, Charlotte, NC 28262-8563, U.S.A. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Notices</b> . . . . .	vii
Trademarks . . . . .	vii
<b>About This Publication</b> . . . . .	ix
Who Should Read This Publication . . . . .	ix
How Is This Publication Organized? . . . . .	ix
Related Publications . . . . .	x
<b>Chapter 1. Introduction to the Propagation of Adjustments Feature</b> . . . . .	1-1
Introduction to the Propagation of Adjustments Feature . . . . .	1-3
System Considerations . . . . .	1-3
Function Overview . . . . .	1-3
Creating and Maintaining a Repository of Adjustments . . . . .	1-4
Adjustments Repository Update (DKNPRUP) . . . . .	1-5
Adjustment Audit Trail/Reporting (DKNPRAR) . . . . .	1-5
Flow Diagram . . . . .	1-5
Recovery from a CPCS-I System Outage . . . . .	1-6
End User Interactions . . . . .	1-7
<b>Chapter 2. Installation Information for Propagation of Adjustments</b> . . . . .	2-1
Installing the Propagation of Adjustments Feature . . . . .	2-3
Step 1: Install the CPCS-I Libraries . . . . .	2-3
Step 2: Allocate the PRAD Data Sets . . . . .	2-3
Step 3: Compile and Link-Edit the PRAD Source Code . . . . .	2-3
Step 4: Modify DKNBLDL . . . . .	2-3
Step 5: Assemble and Link-Edit DKNBLDL . . . . .	2-4
Step 6: Modify FEATOPTS . . . . .	2-4
Step 7: Assemble and Link-Edit DKNDSAT . . . . .	2-4
Step 8: Modify CPCSPROC . . . . .	2-4
<b>Chapter 3. Customization Information for PRAD</b> . . . . .	3-1
PRAD User-Processing Exits . . . . .	3-3
User Exit Parameters . . . . .	3-3
PRAD Exit 1—Adjustment Repository Update . . . . .	3-4
Activation . . . . .	3-4
Linkage . . . . .	3-4
Restrictions . . . . .	3-5
Example . . . . .	3-5
PRAD Exit 2—Adjustment Repository Get . . . . .	3-6
Activation . . . . .	3-6
Linkage . . . . .	3-6
Restrictions . . . . .	3-7
Example . . . . .	3-7
PRAD Exit 3—Adjustment Propagation . . . . .	3-8
Activation . . . . .	3-8
Linkage . . . . .	3-8
Restrictions . . . . .	3-9
Example . . . . .	3-9
PRAD Exit 4—PRAR Audit Trail . . . . .	3-10
Activation . . . . .	3-10

Linkage . . . . .	3-10
Restrictions . . . . .	3-11
Example . . . . .	3-11
<b>Chapter 4. PRAD—Terminal Operations Information . . . . .</b>	<b>4-1</b>
PRAD—Propagation of Adjustments . . . . .	4-3
Task Initiation . . . . .	4-3
DKNPRAD Main Menu . . . . .	4-3
PRAR—Audit Trail Facility . . . . .	4-4
DKNPRAR Manual Start Main Menu . . . . .	4-5
DKNPRAR Bank Report Screen . . . . .	4-6
DKNPRAR Cycle Report Screen . . . . .	4-7
DKNPRAR Generate Extract File Screen . . . . .	4-8
DKNPRAR Repository Maintenance Screens . . . . .	4-9
DKNPRAR Repository Maintenance Confirmation . . . . .	4-10
PRUP—Repository Update Facility . . . . .	4-11
DKNPRUP Manual Start Screen . . . . .	4-12
PRAD Messages . . . . .	4-12
<b>Chapter 5. PRAD—Programming and Diagnostic Information . . . . .</b>	<b>5-1</b>
Propagation of Adjustments Application Programming Interface . . . . .	5-3
Feature Active Determination . . . . .	5-3
Application Programming Interface Invocation . . . . .	5-3
PRAD Application Programming Interface Input . . . . .	5-6
Request Parameter List (DKNPA001 Structure ID) . . . . .	5-7
Object/Action Pairs and Request Parameter List (RPL) Fields . . . . .	5-8
Primary Return Codes . . . . .	5-11
Extended Return Codes . . . . .	5-11
Propagation of Adjustments Application Programming Interface Keys . . . . .	5-13
PRAD CPCS-I Propagation Flags . . . . .	5-15
PRAD Adjustment Record (DKNPA002 Structure ID) . . . . .	5-15
Security . . . . .	5-16
Creating a Propagation of Adjustments Audit Trail Report . . . . .	5-17
<b>Appendix A. Data Areas for PRAD . . . . .</b>	<b>A-1</b>
Data Areas for Propagation of Adjustments . . . . .	A-1
<b>Appendix B. Messages for PRAD . . . . .</b>	<b>B-1</b>
Messages for Propagation of Adjustments . . . . .	B-1
<b>Glossary . . . . .</b>	<b>GL-1</b>
<b>Bibliography . . . . .</b>	<b>BIB-1</b>
ACF/VTAM Publications . . . . .	BIB-1
Document Processor Support Publications . . . . .	BIB-1
High Performance Transaction System Publications (Version 1) . . . . .	BIB-1
High Performance Transaction System Publications (Version 2) . . . . .	BIB-1
MVS Publications (Version 5) . . . . .	BIB-1
RACF Publications . . . . .	BIB-2
CPCS Enhanced System Manager Publication . . . . .	BIB-2
<b>Index . . . . .</b>	<b>X-1</b>



---

# Figures

1-1.	Propagation of Adjustments (PRAD) Flow Diagram	1-6
3-1.	PRAD Exit 1 Communication Control Blocks	3-4
3-2.	PRAD Exit 1 Routine Return Codes	3-5
3-3.	PRAD Exit 2 Communication Control Blocks	3-6
3-4.	PRAD Exit 2 Routine Return Codes	3-6
3-5.	PRAD Exit 3 Communication Control Blocks	3-8
3-6.	PRAD Exit 3 Routine Return Codes	3-8
3-7.	PRAD Exit 4 Communication Control Blocks	3-10
3-8.	PRAD Exit 4 Routine Return Codes	3-10
4-1.	DKNPRAD Main Menu Screen	4-3
4-2.	Format of the PRAR Command	4-4
4-3.	DKNPRAR Manual Start Main Menu	4-5
4-4.	DKNPRAR Bank Report Screen	4-6
4-5.	DKNPRAR Cycle Report Screen	4-7
4-6.	DKNPRAR Generate Extract File Screen	4-8
4-7.	DKNPRAR Repository Maintenance Only Screen	4-9
4-8.	DKNPRAR Repository Maintenance Confirmation Screen	4-10
4-9.	Format of the PRUP Command	4-11
4-10.	DKNPRUP Manual Start	4-12
5-1.	Data Names and Values for the ESM System Request Block	5-3
5-2.	Assembler Example of Application Programming Interface Invocation	5-4
5-3.	COBOL Example of Application Programming Interface Invocation	5-5
5-4.	PRAD Service Processor Object/Action Definitions	5-6
5-5.	PRAD Application Programming Interface Request Parameter List	5-7
5-6.	PRAD Request Parameter List Action/Required Fields Matrix	5-8
5-7.	PRAD Request Parameter List Action/Required Fields Matrix	5-11
5-8.	PRAD General-Use Programming Interface Extended Return Codes	5-12
5-9.	PRAD Application Programming Interface Request Code to Key Type Matrix	5-14
5-10.	Definition of CPCS-I Flags Field	5-15
5-11.	PRAD Adjustment Record in General-Use Programming Interface Format	5-15
5-12.	Report Layout for the PRAD Audit Trail Report (Sample UK MDS)	5-17



---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectable rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact: IBM Corporation, Department MG39/204, 8501 IBM Drive, Charlotte, NC 28262-8563, U.S.A. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, U.S.A.

---

## Trademarks

The following terms, denoted by an asterisk (\*) elsewhere in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM	COBOL/370
High Performance Transaction System	IBM
ImagePlus	LE/370
MVS/ESA	MVS/SP
RACF	SMP/E
VTAM	

All other trademarks, denoted by a double asterisk (\*\*) in this publication, are the property of their respective owners.



---

## About This Publication

The *Check Processing Control System International MVS/ESA: Propagation of Adjustments Guide* describes this feature's tasks, task initiation formats, installation and customization details, including user exits, and data areas that are part of IBM Check Processing Control System International MVS/ESA (CPCS-I).

You can use this book as the basis for defining the process flows to be used at your institution. Although this guide is not a training manual, it should have all the information you need to operate and program your system for the Propagation of Adjustments feature. For specific information on CPCS-I installation, customization, messages, user exits, and data areas, refer to those specific books in the CPCS-I library (listed under "Related Publications" on page x).

**Note:** Because CPCS-I does not limit the ways that you can combine functions, it is possible to perform combinations of functions that produce illogical results. Your institution should set up its own procedures and manuals for CPCS-I operators and supervisors and restrict the use of CPCS-I functions, based on the results that you want to achieve.

---

## Who Should Read This Publication

The *CPCS-I Propagation of Adjustments Guide* is intended for use by experienced operators of CPCS-I and personnel responsible for establishing and describing the ways that your institution uses this feature with CPCS-I. For more information that describes the system environment for this feature, see *CPCS-I General Information*.

---

## How Is This Publication Organized?

This book discusses in detail the various tasks necessary for using this feature. It follows, by chapter, the tasks as described in the various CPCS-I books for the base product.

This book consists of five chapters and two appendixes.

- "Introduction to the Propagation of Adjustments Feature" gives you a quick introduction to the Propagation of Adjustments feature, system considerations, function, and information regarding creating and maintaining a repository of adjustments.
- "Installing the Propagation of Adjustments Feature" gives you the specific information you need for installing the Propagation of Adjustments (PRAD) feature on your system.
- "PRAD User-Processing Exits" has the specific user exit detail and other customization information necessary for PRAD use.
- "PRAD—Propagation of Adjustments" gives you the terminal operations information for each CPCS-I task. For the task terminal messages and the operator responses that these messages require, see "Messages for Propagation of Adjustments" on page B-1. To start or respond to any CPCS-I

---

\* Trademark of IBM

task from a terminal, enter the response as described under “Task Initiation” or “Response” for each task.

- “Creating and Maintaining a Repository of Adjustments,” gives you programming information for personnel who use the IBM Check Processing Control System (CPCS-I) licensed program. Programmers who have to modify CPCS-I for their installation can use this book to learn the internal operations of the product. This chapter gives functional descriptions and application outputs for the Propagation of Adjustments (PRAD) feature.
- “Data Areas for Propagation of Adjustments” on page A-1 lists the seven PRAD data areas.
- “Messages for Propagation of Adjustments” on page B-1 lists the PRAD messages.

This book also contains a glossary, a bibliography, and an index.

---

## Related Publications

The following publications contain information that relates to Check Processing Control System International MVS/ESA (CPCS-I). For an additional list of relevant publications, see the “Bibliography.”

- *IBM Check Processing Control System International MVS/ESA: General Information*, GC31-2944  
**Short Title:** *CPCS-I General Information*

This publication gives a general introduction to CPCS-I. It describes various features and advantages of CPCS-I and the hardware and software requirements for operating this system. It also discusses CPCS-I support of the IBM 3890 Document Processor and the IBM 3890/XP Series document processors, along with some of the features of these processors.

- *IBM Check Processing Control System International MVS/ESA: Installation Guide*, GC31-2942  
**Short Title:** *CPCS-I Installation Guide*

This guide describes the steps necessary for using the IBM System Modification Program Extended (SMP/E) procedures to install CPCS-I software. It also provides installation procedures for generating CPCS-I modules and creating operational data sets. It provides data for sample problems to test and verify operations after CPCS-I installation.

- *IBM Check Processing Control System International MVS/ESA: Terminal Operations Guide*, SC31-2946  
**Short Title:** *CPCS-I Terminal Operations Guide*

This guide explains how to perform CPCS-I tasks and is written for the CPCS-I operators. Included in this guide are terminal operations for the MICR restart procedures and sample reports.

- *IBM Check Processing Control System International MVS/ESA: Programming Guide*, SC31-2948  
**Short Title:** *CPCS-I Programming Guide*

This guide contains guidelines for CPCS-I programmers. It includes information about application-program processing, problem analysis and documentation procedures.

- *IBM Check Processing Control System International MVS/ESA: Programming Reference*, GC31-3997  
**Short Title:** *CPCS-I Programming Reference*

This publication gives a structured view of the CPCS-I interfaces, specifically application programming, Assembler macros, subroutines, and some control block information.

- *IBM Check Processing Control System International MVS/ESA: Customization Guide*, SC31-2943  
**Short Title:** *CPCS-I Customization Guide*

This guide provides customization information for CPCS-I programmers. It also includes system programming information, and generation and installation procedures.

- *IBM Check Processing Control System International MVS/ESA: Messages and Codes*, SC31-3981  
**Short Title:** *CPCS-I Messages and Codes*

This book describes console and supervisor messages, as well as program return and exit codes.

- *IBM Check Processing Control System International MVS/ESA: Propagation of Adjustments*, SC31-3994  
**Short Title:** *CPCS-I Propagation of Adjustments Guide*

This guide contains the guidelines for the CPCS-I personnel who use the Propagation of Adjustments (PRAD) feature. It includes functional descriptions and information about terminal operations, programming, and application output.

- *IBM Check Processing Control System International MVS/ESA: Master Index*, SC31-3980  
**Short Title:** *CPCS-I Master Index*

This reference combines the index entries for all the publications in the CPCS-I library.





---

## Chapter 1. Introduction to the Propagation of Adjustments Feature

Introduction to the Propagation of Adjustments Feature . . . . .	1-3
System Considerations . . . . .	1-3
Function Overview . . . . .	1-3
Creating and Maintaining a Repository of Adjustments . . . . .	1-4
Adjustments Repository Update (DKNPRUP) . . . . .	1-5
Adjustment Audit Trail/Reporting (DKNPRAR) . . . . .	1-5
Flow Diagram . . . . .	1-5
Recovery from a CPCS-I System Outage . . . . .	1-6
End User Interactions . . . . .	1-7

## Introduction

---

## Introduction to the Propagation of Adjustments Feature

The IBM\* Check Processing Control System International MVS/ESA\* (CPCS-I) Propagation of Adjustments feature maintains an Adjustment Repository accessible to user-written programs. An application programming interface (API) allows users to have single-point access to this repository, which contains item adjustment information and propagation status for each item.

The Propagation of Adjustments (PRAD) feature provides an optional component that creates and maintains this repository of adjustments made by any balancing operation. An adjustment is said to be *propagated* when it has been processed by one or more applications. PRAD also provides an audit trail and reporting facility.

Four user exits and seven data areas support this feature.

---

## System Considerations

The Propagation of Adjustments feature, written in System/370\* Assembler Language and COBOL/370\*, was developed using the following program offerings:

- Multiple Virtual Storage (MVS) Release 3 or higher (MVS/ESA)
- COBOL/370
- Assembler/370

The PRAD feature uses functions of CPCS Enhanced System Manager.

---

## Function Overview

The Propagation of Adjustments feature lets you easily do the following tasks:

- Record all balancing application adjustments
- Record all adjustments successfully listed on dispatched RMIT/KILL lists as *propagated*
- Record all adjustments successfully extracted externally from CPCS-I as *propagated*
- Record all other adjustments as *non-propagated*
- Delete and reapply adjustments if the balancing application rebalanced a CPCS-I entry or subset string
- Report by bank within a CPCS-I cycle, all *propagated/non-propagated* adjustments and extract this information to a dynamically-allocated report file.

---

\* Trademark of IBM

## Creating and Maintaining a Repository of Adjustments

For final control, the Propagation of Adjustments feature delivers, when requested, an output report showing each adjustment.

The chapters that follow describe the Propagation of Adjustments feature in detail and provide you with installation, customization, programming, and operations information.

---

## Creating and Maintaining a Repository of Adjustments

The Propagation of Adjustments (PRAD) feature provides an optional CPCS-I component that creates and maintains a repository of adjustments made by any Balancing operation. PRAD extracts these adjustments from any M-string passed to the DKNPRUP task upon invocation. This component includes an application programming interface for use by user-written applications. CPCS-I applications (for example: KILL, RMIT, and MCRE) also use the same interface to report the most up-to-date information on the reports and files they generate. The ability of these CPCS-I applications to reflect *all* adjustments in their output depends on operations scheduling. An adjustment is said to be *propagated* when it has been processed by one or more applications. PRAD also provides an audit trail and reporting facility. This facility enables you to determine when or if reports or electronic output have been *dispatched* prior to completing the Balancing process.

The PRAD component allows CPCS-I to perform the following functions:

- Record all adjustments made by a Balancing application.
- Record all adjustments that have been successfully listed on all dispatched RMIT/KILL lists as *propagated*.
- Record all adjustments that have been successfully extracted externally from CPCS-I (related outgoing work from MCRE) as *propagated*.
- Record all adjustments that were not reflected in the dispatched output listings and related extracted data as *non-propagated*.
- Delete and reapply adjustments if the Balancing application decided to re-balance a CPCS-I entry/subset string.
- Report by bank within a CPCS-I cycle, all *propagated/non-propagated* adjustments and extract this information to a dynamically-allocated report file.

In addition, four user exits allow:

- User control of adjustments added to the adjustments data base
- User control of adjustments being read from the adjustment repository
- User control of adjustments being *propagated* to CPCS-I or user-written applications
- User control of adjustment records as the audit trail is being produced.

For specific PRAD user exit information, see “PRAD User-Processing Exits” on page 3-3.

PRAD maintains an adjustment repository that is internal to CPCS-I and is accessible to user-written programs only through the application programming interface (API) (as shown in “Propagation of Adjustments Application Programming Interface” on page 5-3).

All adjustments made to an item are recorded in the repository and reported through the PRAR task. All adjustment information is backed by permanent storage.

### **Adjustments Repository Update (DKNPRUP)**

A CPCS-I M-string is modified and placed into balance through adjustments. For every adjustment that the Balancing application chooses to make, the following information is available within the CPCS-I string:

- User-defined adjustment code
- Fixed adjustment type codes:
  - Insert
  - Change
  - Delete
  - Move

**Note:** PRAD supports all four types.

DKNPRUP extracts adjustments from CPCS-I balanced *nnM*-strings and adds those adjustments to the adjustments repository. This task can be initiated manually through a command (PRUP) and online screens, or automatically through System Manager, based on the workflows defined for *nnM*-Strings.

### **Adjustment Audit Trail/Reporting (DKNPRAR)**

DKNPRAR produces an audit trail report that can be used to research the status of adjustments. This status is relative to the output that is dispatched upon completion of both CPCS-I applications (RMIT, KILL, and MCRE), and those adjustments propagated by user-written applications. The report lists the adjustment detail activity by cycle, bank, and item sequence number, of all the adjusted items within a cycle. Users have the option to create a flat file containing the same information. This file can be used as input to back-end applications that provide a variety of functions to aid the research and resolution of exception items and ensure proper control of the adjustment activity.

This task also performs the repository maintenance function on the adjustments repository. The repository maintenance function deletes adjustments on a cycle basis. IBM recommends that the user invoke this function before end-cycle processing.

### **Flow Diagram**

Figure 1-1 on page 1-6 illustrates the use of the PRAD interface to reflect adjustments made to items that have been distributed to RMIT D-strings prior to being adjusted.

## Creating and Maintaining a Repository of Adjustments

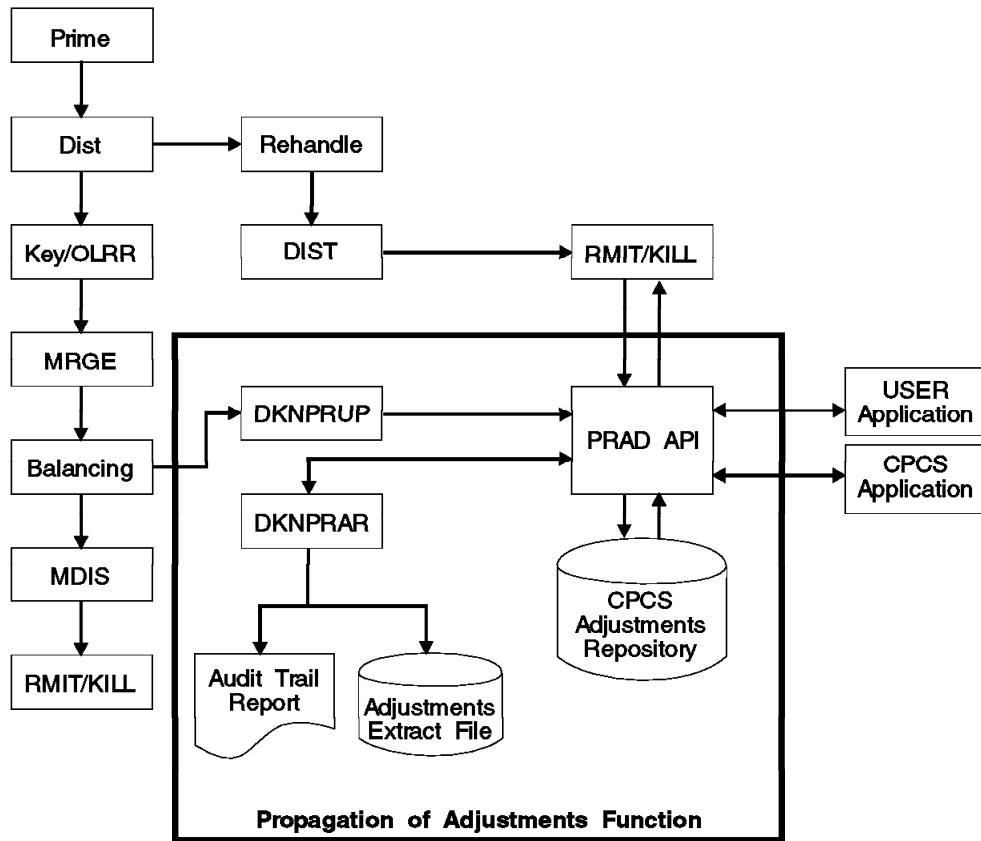


Figure 1-1. Propagation of Adjustments (PRAD) Flow Diagram

By using this operational scenario, you can process rehandle passes of prime D-strings (while the sorter operators and documents are immediately available) prior to the Balancing operation. The adjustments made during Balancing (through an *nnM*-string) are reflected in the remittance listings under one condition. The adjustments must be extracted by the PRAD task prior to the invocation of the RMIT/KILL task that lists one or more of the items contained in that *nnM*-string. At any point in this process, the user may run the PRAR task to determine if and when any adjustments have been propagated.

### Recovery from a CPCS-I System Outage

Because permanent storage backs all adjustment information, recovery from a system outage is automatic and requires no user intervention.

**Application Task Failure:** The recovery for CPCS-I application tasks using the PRAD interface is to simply rerun the task. Since adjustments propagated by an application task are not permanent until the application issues a *commit* request, the adjustments that task propagated prior to abnormally terminating, are useless and are discarded.

**Deleted Strings:** A manual start of the PRUP task recovers the adjustments for deleted *nn M*-strings, once the recovery task, DKNRCVY, runs. These adjustments are placed in the repository as *new* adjustments. When the original string is deleted from the mass data set, the *old* adjustments are soft deleted.

### End User Interactions

End users interact with Propagation of Adjustments services through the PRAD, PRAR, and PRUP commands and their associated screens for manual operation. For specific information concerning the PRAD, PRAR, and PRUP commands, see the specific areas in Chapter 4, “PRAD—Terminal Operations Information” on page 4-1 where each command is documented. In order to automate invocation of the PRAD tasks, users can customize the Enhanced System Manager workflows and task profiles using Enhanced System Manager on-line functions (SMOF). IBM recommends that you invoke the PRUP task automatically with Enhanced System Manager to ensure proper timing of normal workflows. IBM recommends that you invoke the PRAR task manually immediately prior to end-of-cycle processing.

## Creating and Maintaining a Repository of Adjustments



---

## Chapter 2. Installation Information for Propagation of Adjustments

Installing the Propagation of Adjustments Feature . . . . .	2-3
Step 1: Install the CPCS-I Libraries . . . . .	2-3
Step 2: Allocate the PRAD Data Sets . . . . .	2-3
Step 3: Compile and Link-Edit the PRAD Source Code . . . . .	2-3
Step 4: Modify DKNBLDL . . . . .	2-3
Step 5: Assemble and Link-Edit DKNBLDL . . . . .	2-4
Step 6: Modify FEATOPTS . . . . .	2-4
Step 7: Assemble and Link-Edit DKNDSAT . . . . .	2-4
Step 8: Modify CPCSPROC . . . . .	2-4



---

## Installing the Propagation of Adjustments Feature

The installation procedures described in this chapter tell you how to allocate the required PRAD data sets and assemble (or compile) and link-edit the program modules. When you complete these steps, PRAD is active.

The steps for installing PRAD are:

1. Install the CPCS-I libraries.
2. Allocate the PRAD data sets.
3. Compile and link-edit the PRAD source code.
4. Modify DKNBLDL.
5. Assemble and link-edit DKNBLDL.
6. Modify FEATOPTS.
7. Assemble and link-edit DKNDSAT.
8. Modify CPCSPROC.

---

### Step 1: Install the CPCS-I Libraries

In this step, you install the CPCS-I libraries, using the instructions provided in the CPCS-I Program Directory.

---

### Step 2: Allocate the PRAD Data Sets

In this step, you allocate the data sets needed to run PRAD. The JCL is in the member ALLOPRAD of CPCSI.V01R01.SDKNSAM1. This job allocates all the required PRAD data sets, using the following default assumptions:

- All the PRAD data sets are put on Volume nnnnnn. Use a volume label defined for your installation.
- All the PRAD data sets are assigned a high-level qualifier of CPCSI.V01R01.

**Important!**

You can install PRAD with the default qualifier of CPCSI.V01R01, or you can change the qualifier to meet the requirements of your installation. If you change the qualifier, you must review all the JCL members referenced to ensure you are using the correct qualifier before you run the jobs. Also, this job deletes any previous PRAD data sets.

---

### Step 3: Compile and Link-Edit the PRAD Source Code

In this step, you compile and link-edit all PRAD shipped modules. The JCL is in the member CMPLPRAD of CPCSI.V01R01.SDKNSAM1. The load modules are in CPCSI.V01R01.SDKNMOD1.

---

### Step 4: Modify DKNBLDL

In this step, you must uncomment the DKNPRDI entry in the DKNBLDL source file.

---

## **Step 5: Assemble and Link-Edit DKNBLDL**

In this step, you compile and link-edit the module, DKNBLDL. The JCL is in the member CMPLBLDL of CPCSI.V01R01.SDKNSAM1.

---

## **Step 6: Modify FEATOPTS**

In this step, you modify your FEATOPTS by setting the PRADFEAT variable to YES.

---

## **Step 7: Assemble and Link-Edit DKNDSAT**

In this step, you run CMPLDSAT to assemble and link-edit DKNDSAT.

---

## **Step 8: Modify CPCSPROC**

In this step, you modify your CPCSPROC that shows the PRAD data sets commented out. Your modification is to uncomment the PRAD data sets.

---

## Chapter 3. Customization Information for PRAD

PRAD User-Processing Exits . . . . .	3-3
User Exit Parameters . . . . .	3-3
PRAD Exit 1–Adjustment Repository Update . . . . .	3-4
Activation . . . . .	3-4
Linkage . . . . .	3-4
Restrictions . . . . .	3-5
Example . . . . .	3-5
PRAD Exit 2–Adjustment Repository Get . . . . .	3-6
Activation . . . . .	3-6
Linkage . . . . .	3-6
Restrictions . . . . .	3-7
Example . . . . .	3-7
PRAD Exit 3–Adjustment Propagation . . . . .	3-8
Activation . . . . .	3-8
Linkage . . . . .	3-8
Restrictions . . . . .	3-9
Example . . . . .	3-9
PRAD Exit 4–PRAR Audit Trail . . . . .	3-10
Activation . . . . .	3-10
Linkage . . . . .	3-10
Restrictions . . . . .	3-11
Example . . . . .	3-11



---

## PRAD User-Processing Exits

Propagation of Adjustments (PRAD) consists of the following user-exits:

- Adjustment repository update
- Adjustment repository get
- Adjustment propagation
- Audit trail

You can specify the PRAD user exits in the bank control file. See the section on “Bank Control File Record Definitions” in *CPCS-I Customization Guide* for additional information on enabling the PRAD user-exits.

---

## User Exit Parameters

All propagation of adjustments (PRAD) user-exits are passed the following parameter list:

<b>Name</b>	<b>Type, Length</b>	<b>I/O<sup>1</sup></b>	<b>Description</b>
Structure Name	char, 8	I	Constant value of DKNPA003. It is the data structure identifier.
PRAD Action	char, 8	I	PRAD requested service action.
Reserved	byte, 8	N/A	Available for user-exit use
Adjustment Record	fullword, 4	I	Pointer to the adjustment record
Reserved	fullword, 4	N/A	Reserved for expansion

---

<sup>1</sup> Input (I) fields contain values for the Application Programming Interface (API) on entry and must not be modified. Output (O) fields are for values returned by the API and are not initialized on input. Input/Output (I/O) fields contain values for the API on entry. The API also returns values in the fields.

## PRAD Exit 1–Adjustment Repository Update

This exit provides the ability to select the adjustments to be included in the adjustment repository. The exit allows you to:

- Inspect and modify each adjustment record before it is written to the repository.
- Exclude adjustment records from being written to the repository. This provides the ability to establish criteria for the adjustments that are managed by the PRAD component. For example, you can prevent the addition of adjustments for certain account numbers and sort codes to the adjustment repository. The adjustment repository update user-exit has no information on prior or following adjustments of any given item.

The exit routine may be called before writing each record to the adjustment repository.

### Activation

A different PRUP exit routine may be specified for each financial institution. The exit routine gets activated when its name is specified in the bank control data set (see “Bank Control File Record Formats” in the *CPCS-I Customization Guide* for more information).

This user exit is given control within the PRAD application programming interface (API), but runs under the calling module’s task control block (TCB) (for example, RMIT/KILL). This exit routine must be re-entrant.

The PRAD API is a synchronous interface. No support exists for ESTAE processing within the PRAD API. The ESTAE routine for the calling application is invoked during an abnormal termination.

No regenerations, CPCS-I restarts, or CHAPs are required when an exit routine is activated or deactivated or when a new version of an exit routine is installed.

This exit is not recommended for making decisions on dollar amounts, because it has no knowledge of prior or following adjustments of any given item. Those decisions should be made by the adjustment propagation exit routine, which has all adjustment information for a given item.

### Linkage

The standard MVS linkage conventions are followed. Register 1 points to a parameter list that contains the addresses of the following interface control blocks:

*Figure 3-1. PRAD Exit 1 Communication Control Blocks*

<b>Control Block</b>	<b>COBOL Copybook</b>	<b>Assembler Macro/ Copybook</b>
Application Task Control Block	DKNCATCB	DKNAPTCB
Call Parameters	DKNPA03C	DKNPA003

The exit routine may return the following completion codes in the RETURN-CODE special register (COBOL programs) or register 15 (Assembler programs):



*Figure 3-2. PRAD Exit 1 Routine Return Codes*

<b>Code (dec)</b>	<b>Request</b>
+00	Write record to adjustment repository.
+04	Do not write record to adjustment repository.

## Restrictions

PRAD Exit 1 routines:

- Should preferably be written in assembler, but may also be written in COBOL.
- Must be re-entrant.

## Example

Member DKNPDUE1 of CPCSI.V01R01.SDKNSAM2 may be used as an example of coding assembler adjustment repository update exit routines.

## PRAD Exit 2–Adjustment Repository Get

This exit may be used to prevent certain adjustment records from being presented to specific applications on adjustment repository read requests.

The exit routine may be called before reading each record from the adjustment repository.

Adjustment repository get allows you to examine any *get* type of requests for an adjustment record. You can establish criteria for account numbers and the type of work to prevent the transfer of specific adjusted items to specific applications.

### Activation

A different PRAD exit routine may be specified for each financial institution. The exit routine is activated when its name is specified in the bank control data set (see “Bank Control File Record Formats” in the *CPCS-I Customization Guide* for more information).

This user-exit is given control within the PRAD API, but runs under the calling task’s TCB (for example, RMIT/KILL). This exit routine should be re-entrant.

PRAD API is a synchronous interface. There is no support for ESTAE processing within the PRAD API. The ESTAE routine for the calling application is invoked in the event of an abnormal termination.

No regenerations, CPCS-I restarts or CHAPs are required when an exit routine is activated or deactivated or when a new version of an exit routine is installed.

### Linkage

The standard MVS linkage conventions are followed. Register 1 points to a parameter list that contains the addresses of the following interface control blocks:

Figure 3-3. PRAD Exit 2 Communication Control Blocks

Control Block	COBOL Copybook	Assembler Macro/ Copybook
Application Task Control Block	DKNCATCB	DKNAPTCB
Call Parameters	DKNPA03C	DKNPA003

The exit routine may return the following completion codes in the RETURN-CODE special register (COBOL programs) or register 15 (Assembler programs):

Figure 3-4. PRAD Exit 2 Routine Return Codes

Code (dec)	Request
+00	Read record from adjustment repository.
+04	Do not read record from adjustment repository.

## Restrictions

PRAD Exit 2 routines:

- Should preferably be written in Assembler, but may also be written in COBOL.
- Must be re-entrant.

## Example

Member DKNPDUE2 of CPCSI.V01R01.SDKNSAM2 may be used as an example of coding assembler adjustment repository get exit routines.

## PRAD Exit 3—Adjustment Propagation

This exit provides the ability to select the adjustments to be propagated to CPCS-I or user-written applications. The exit allows you to:

- Inspect and modify each adjustment record before it is propagated.
- Exclude adjustment records from being propagated.

The exit routine may be called before propagating each record.

The difference in the function of this exit from that of the repository update exit is that the most up-to-date adjustment information for any given item is available in this exit. IBM recommends that dollar amount criteria logic be applied in this exit.

## Activation

A different PRAD exit routine may be specified for each financial institution. The exit routine is activated when its name is specified in the bank control data set (see “Bank Control File Record Formats” in the *CPCS-I Customization Guide* for more information).

This user-exit is given control within the PRAD API, but runs under the calling task’s TCB (for example, RMIT/KILL). This exit routine should be re-entrant.

PRAD API is a synchronous interface. There is no support for ESTAE processing within the PRAD API. The ESTAE routine for the calling application is invoked in the event of an abnormal termination.

No regenerations, CPCS-I restarts, or CHAPs are required when an exit routine is activated or deactivated or when a new version of an exit routine is installed.

## Linkage

The standard MVS linkage conventions are followed. Register 1 points to a parameter list that contains the addresses of the following interface control blocks:

Figure 3-5. PRAD Exit 3 Communication Control Blocks

Control Block	COBOL Copybook	Assembler Macro/ Copybook
Application Task Control Block	DKNCATCB	DKNAPTCB
Call Parameters	DKNPA03C	DKNPA003

The exit routine may return the following completion codes in the RETURN-CODE special register (COBOL programs) or register 15 (Assembler programs):

Figure 3-6. PRAD Exit 3 Routine Return Codes

Code (dec)	Request
+00	Propagate adjustment record.
+04	Do not propagate adjustment record.

## Restrictions

PRAD Exit 3 routines:

- Should preferably be written in Assembler, but may also be written in COBOL.
- Must be re-entrant.

## Example

Member DKNPDUE3 of CPCSI.V0R01.SDKNSAM2 may be used as an example of coding assembler adjustment propagation exit routines.

---

## PRAD Exit 4–PRAR Audit Trail

This exit provides the ability to control the adjustment repository audit trail and reporting functions. The exit allows you to:

- Inspect and modify each adjustment record before it is processed.
- Exclude adjustment records from being written to the audit trail report.
- Exclude adjustment records from being written to the audit trail extract file.

The exit routine may be called with the following function codes:

- Initialization—This call is made before any adjustment records have been processed.
- Processing—This call is made before each adjustment record is ready to be written to the audit trail report and audit trail extract file.
- Termination—This call is made after all records have been processed.

## Activation

A different PRAR exit routine may be specified for each financial institution. The exit routine is activated when its name is specified in the bank control data set (see “Bank Control File Record Formats” in the *CPCS-I Customization Guide*, for more information). This exit routine is given control under DKNPRAR’s TCB.

No regenerations, CPCS-I restarts, or CHAPs are required when an exit routine is activated or deactivated, or when a new version of an exit routine is installed.

## Linkage

The standard MVS linkage conventions are followed. Register 1 points to a parameter list that contains the addresses of the following interface control blocks:

*Figure 3-7. PRAD Exit 4 Communication Control Blocks*

Control Block	Cobol Copybook	Assembler Macro/ Copybook
Application Task Control Block	DKNCATCB	DKNAPTCB
Call Parameters	DKNPA03C	DKNPA003

The exit routine may return the following codes in the RETURN-CODE special register (COBOL programs) or register 15 (Assembler programs):

*Figure 3-8. PRAD Exit 4 Routine Return Codes*

Code (dec)	Request
+00	Write adjustment record.
+04	Do not write adjustment record.

## Restrictions

PRAD Exit 4 routines:

- Should preferably be written in COBOL, but may also be written in assembler.
- Must be re-entrant.

## Example

Member DKNPDUE4 of CPCS.V01R01.SDKNSAM1 may be used as an example of coding COBOL PRAD audit trail exit routines.





---

## Chapter 4. PRAD—Terminal Operations Information

PRAD—Propagation of Adjustments	4-3
Task Initiation	4-3
DKNPRAD Main Menu	4-3
PRAR—Audit Trail Facility	4-4
DKNPRAR Manual Start Main Menu	4-5
DKNPRAR Bank Report Screen	4-6
DKNPRAR Cycle Report Screen	4-7
DKNPRAR Generate Extract File Screen	4-8
DKNPRAR Repository Maintenance Screens	4-9
DKNPRAR Repository Maintenance Confirmation	4-10
PRUP—Repository Update Facility	4-11
DKNPRUP Manual Start Screen	4-12
PRAD Messages	4-12



## PRAD—Propagation of Adjustments

The Propagation of Adjustments function maintains an adjustment repository that is accessible to user-written programs only through an application programming interface (API). The purpose of this interface is to provide a single access point to the Adjustment Repository. This repository contains item adjustment information and the propagation status of the adjustment information for each item. For more information about how the Propagation of Adjustments component works, see “Creating and Maintaining a Repository of Adjustments” on page 1-4.

### Task Initiation

Start the PRAD main menu by entering the following at the READY prompt: PRAD

### DKNPRAD Main Menu

```

DKNPRAD-10                PROPAGATION OF ADJUSTMENTS          DATE : XX/XX/XX
                           - MAIN MENU -                       TIME : HH:MM:SS
                                                                    ID   : XXXXXXXX

SELECTION: _  1) MANUALLY ADD/DELETE ADJUSTMENTS
               2) PRODUCE AUDIT TRAIL
               3) REPOSITORY MAINTENANCE

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END

```

Figure 4-1. DKNPRAD Main Menu Screen

The option choices are as follows:

- 1 Start up the Manual Screen for Add and Delete adjustments (see Figure 4-10 on page 4-12)
- 2 Start up the Audit Trail Generator for PRAD (see Figure 4-3 on page 4-5)
- 3 Start up the Maintenance Functions of PRAD (see Figure 4-7 on page 4-9)

For all PRAD screens, XXXXX...XXX shows the message line.

## PRAR—Audit Trail Facility

The PRAR command invokes the audit trail facility for Propagation of Adjustments (PRAD). You have options that allow the creation of reports and the extraction of files for a specific cycle or for a specific bank within a cycle.

**Note:** The PRAR options are positional. If you omit an option, you must type a comma as a place-holder.

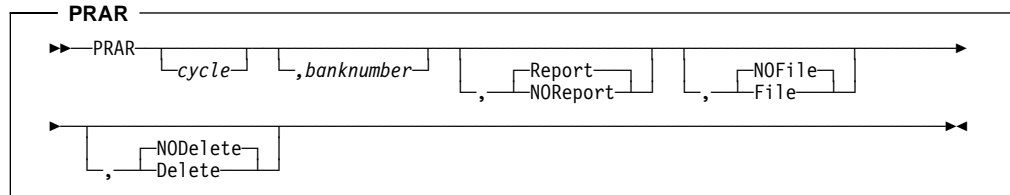


Figure 4-2. Format of the PRAR Command

**where:**

*cycle* Specifies the cycle for which the audit trail is to be run.

*banknumber*

Specifies the bank number for which the audit trail is to be created. This option creates the audit trail for the specified bank only within the specified cycle. This is an optional parameter.

**REPORT|NOREPORT**

Specifies whether to create a hardcopy report.

**REPORT** Indicates that a hardcopy report should be created. This is the default.

**NOREPORT**

Indicates that no hardcopy report should be created.

**FILE|NOFILE**

Specifies whether to create an extract file.

**FILE** Indicates that an extract file should be created.

**NOFILE** Indicates that no extract file should be created. This is the default.

**DELETE|NODELETE**

Indicates whether to perform repository maintenance.

**DELETE** Indicates that all adjustments will be deleted from the PRAD database for the specified cycle.

**Note:** IBM does not recommend the use of both the NOFILE and NOREPORT options, along with the DELETE option, as no audit trail is produced.

**NODELETE**

Indicates that no adjustments will be deleted from the PRAD database. This is the default.

**Note:** Omitting all options causes the display of the manual start screen main menu. The user is prompted for input on the PRAR Main Menu screen. See Figure 4-3 on page 4-5 and the following screens for the formats of the online screens.

## DKNPRAR Manual Start Main Menu

```

DKNPRAR-01          DKNPRAR - PRAD AUDIT TRAIL          DATE : XX/XX/XX
                   - MAIN MENU -                      TIME : HH:MM:SS
                                                         ID  : XXXXXXXX

SELECTION: _  1) REPORT BY BANK
              2) REPORT BY CYCLE
              3) GENERATE EXTRACT FILE
              4) REPOSITORY MAINTENANCE

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END

```

Figure 4-3. DKNPRAR Manual Start Main Menu

This first-level screen allows you to select PRAD audit trail manual start options:

- 1 REPORT BY BANK. Select Option 1 to display another screen that allows you to request an audit trail report and/or an extract file for a bank by entering the bank and cycle number.
- 2 REPORT BY CYCLE. Select Option 2 to display another screen that allows you to request an audit trail report and/or a report file and/or a repository maintenance for a cycle by entering the cycle number.
- 3 GENERATE EXTRACT FILE. Select Option 3 to display another screen that allows you to request an audit trail report file and/or a repository maintenance. This file can be used with user-written applications for customized processing. An audit trail report cannot be created with this option.
- 4 REPOSITORY MAINTENANCE. Select Option 4 to display another screen that allows you to request that all adjustments for a cycle on the PRAD Adjustment Database be deleted by entering the cycle number. No audit trail is produced with this option. Options 2 and 3 allow repository maintenance and produce an audit trail.

**Note:** Repository Maintenance may be restricted to certain users.

### DKNPRAR Bank Report Screen

```
DKNPRAR-10          DKNPRAR - PRAD AUDIT REPORT          DATE : XX/XX/XX
                   - BY BANK -                          TIME : HH:MM:SS
                                                           ID  : XXXXXXXX

                   BANK NUMBER          : 999
                   CYCLE NUMBER         : XX
                   CREATE EXTRACT FILE (Y/N)? : X

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP PF3=END
```

Figure 4-4. DKNPRAR Bank Report Screen

To ask for a PRAD Audit Report by Bank, fill out the screen as follows:

- CYCLE NUMBER**      Enter the cycle number for which you want the report to be generated. This is a required field.
- BANK NUMBER**        Enter the 3-digit bank number or ALL for which you want the report to be generated. This is a required field.
- CREATE EXTRACT FILE**  
Specify "Y" to create an extract file along with the report.  
Specify "N" or leave this field blank if you do not wish to create the extract file.

## DKNPRAR Cycle Report Screen

```

DKNPRAR-20                DKNPRAR - PRAD AUDIT REPORT          DATE : XX/XX/XX
                           - BY CYCLE -                      TIME : HH:MM:SS
                                                                ID  : TCBXXXXX

                           CYCLE NUMBER                      : XX
                           CREATE EXTRACT FILE (Y/N)?       : X
                           REPOSITORY MAINTENANCE (Y/N)?    : X

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END

```

Figure 4-5. DKNPRAR Cycle Report Screen

To obtain a PRAD Audit Report by Cycle, fill out the screen as follows:

**CYCLE NUMBER** Enter the cycle number for the report to be generated. This is a required field.

**CREATE EXTRACT FILE**  
Specify "Y" to create an extract file along with the report. Specify "N" or leave this field blank if you do not wish to create the extract file.

**REPOSITORY MAINTENANCE**  
Specify "Y" to delete all adjustment information on the Adjustment Database that is to be printed on the report. Specify "N" or leave this field blank if you do not wish to perform file maintenance.

**Note:** You may need special authorization to use this option.

### DKNPRAR Generate Extract File Screen

```
DKNPRAR-30          DKNPRAR - PRAD AUDIT EXTRACT FILE          DATE : XX/XX/XX
                   - BY CYCLE -                               TIME : HH:MM:SS
                                                           ID   : XXXXXXXX

                   CYCLE NUMBER          : XX
                   REPOSITORY MAINTENANCE (Y/N)? : X

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END
```

Figure 4-6. DKNPRAR Generate Extract File Screen

To obtain a PRAD Audit Extract File by Cycle, enter the following information:

**CYCLE NUMBER**     Enter the cycle number of the extract file to be generated.  
                         This is a required field.

**REPOSITORY MAINTENANCE**  
Specify "Y" to delete all adjustment information on the Adjustment Database that is to be printed on the report. Specify "N" or leave this field blank if you do not wish to perform file maintenance.

**Note:** You may need special authorization to use this option.



## DKNPRAR Repository Maintenance Screens

```

DKNPRAR-40          DKNPRAR - PRAD REPOSITORY MAINTENANCE          DATE : XX/XX/XX
                    - ONLY -                                       TIME : HH:MM:SS
                                                                ID  : XXXXXXXX

                                CYCLE NUMBER : XX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END
    
```

Figure 4-7. DKNPRAR Repository Maintenance Only Screen

To obtain PRAD Repository Maintenance, enter the following information:

**CYCLE NUMBER** Enter the cycle number for which the repository maintenance is to be performed. All adjustment information for this cycle is deleted from the adjustment database. No audit trail is produced with this option. If you want an audit trail along with Repository Maintenance, use Options 2 or 3 on the main menu.

## DKNPRAR Repository Maintenance Confirmation

Using this option deletes all adjustments in the PRAD Adjustment Database for the specified cycle. Press **ENTER** to continue with the delete process, or press **PF3** to cancel the request and return to the prior screen.

```
DKNPRAR-42          DKNPRAR - PRAD REPOSITORY MAINTENANCE      DATE : XX/XX/XX
                   - CONFIRMATION -                      TIME : HH:MM:SS
CYCLE:  XX                                                  ID  : XXXXXXXX

                   PRESS ENTER TO DELETE ADJUSTMENTS

                               OR

                   PF3 TO CANCEL THE REQUEST

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PF1=HELP  PF3=END
```

Figure 4-8. DKNPRAR Repository Maintenance Confirmation Screen



## DKNPRUP Manual Start Screen

```
DKNPRUP-10      DKNPRUP - PRAD REPOSITORY UPDATE      DATE : XX/XX/XX
                                                         TIME : HH:MM:SS
                                                         ID  : XXXXXXXX

      OPT      STRING NAME      OPT      STRING NAME
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX
      _  XXXX-X-XX-XX-XX-XX-X-XXX  _  XXXX-X-XX-XX-XX-XX-X-XXX

ENTER OPTION D OR A AND THE STRING NAME
PF1=HELP  PF3=END  PF4=SUBMIT
```

Figure 4-10. DKNPRUP Manual Start

The repository update facility of the Propagation of Adjustments component allows you two options:

- D** Mark adjustments previously added to the database for deletion.
- A** Add the adjustments found on the string specified to the database.

Enter the appropriate option in the option field. Previously propagated strings are overlaid with the new adjustment data.

## PRAD Messages

For PRAD messages, see “Messages for Propagation of Adjustments” on page B-1.

---

## Chapter 5. PRAD–Programming and Diagnostic Information

Propagation of Adjustments Application Programming Interface . . . . .	5-3
Feature Active Determination . . . . .	5-3
Application Programming Interface Invocation . . . . .	5-3
PRAD Application Programming Interface Input . . . . .	5-6
Request Parameter List (DKNPA001 Structure ID) . . . . .	5-7
Object/Action Pairs and Request Parameter List (RPL) Fields . . . . .	5-8
Primary Return Codes . . . . .	5-11
Extended Return Codes . . . . .	5-11
Propagation of Adjustments Application Programming Interface Keys . . . . .	5-13
PRAD CPCS-I Propagation Flags . . . . .	5-15
PRAD Adjustment Record (DKNPA002 Structure ID) . . . . .	5-15
Security . . . . .	5-16
Creating a Propagation of Adjustments Audit Trail Report . . . . .	5-17



## Propagation of Adjustments Application Programming Interface

The Propagation of Adjustments Application Programming Interface provides a single access point to the Adjustment Repository. The Adjustment Repository contains item adjustment information and the propagation status of the adjustment information for each item. To achieve this purpose, a set of requests are defined that can be invoked by issuing an Enhanced System Manager (ESM) Service Request Block, using the ESM callable service, DKNSMAPI. This service specifies the appropriate Object/Action pair, along with additional information as parameters. See Figure 5-4 on page 5-6 for details on the Object/Action pairs.

## Feature Active Determination

Call DKNFEATR using CPCSPRAD as the feature name as outlined in the *CPCS-I Programming Guide* (look under DKNFEATR).

## Application Programming Interface Invocation

You can request PRAD Application Programming Interface services by dispatching an Enhanced System Manager Service Request Block to the PRAD Item Handler or Storage Handler by calling the Enhanced System Manager Application Programming Interface module, DKNSMAPI. Invoke the Application Programming Interface from any LE/370\* conforming language in MVS/ESA Primary Address Space Control mode.

The Enhanced System Manager System Request Block must be filled in as shown in Figure 5-1:

Data Name	Value
<b>CSRBEYE</b>	DKNCSRBA
<b>CSRBSS</b>	Low values
<b>CSRBOBJ</b>	PRADSTRG/PRADITEM
<b>CSRBACT</b>	One action from Figure 5-4 on page 5-6.
<b>CSRBAPT</b>	Low values
<b>CSRBOCOP</b>	OCO parameter list address
<b>CSRBOBID</b>	Low values
<b>CSRBREQA</b>	Request Parameter List address (PA001)
<b>CSRBRETA</b>	CSRBRETN address

Figure 5-1. Data Names and Values for the ESM System Request Block

\* Trademark of IBM.

Figure 5-2 shows an Assembler example of an application programming interface invocation:

```

INITONE DS    0H
        MVC   CSRBEYE,EYEC SRB      CSRB EYECATCHER
        MVC   CSRBOBJ,OBJECT        PRAD OBJECT
        LA    R1,PA001              GET PRAD API REQ PARM LIST ADDR
        ST    R1,CSRBREQA           STORE AS SEND DATA AREA ADDR
        LA    R1,CSRBRETN           RETURN DATA ADDRESS
        ST    R1,CSRBRNTNA
        LA    R1,CSRBR              GET CSRB ADDR
        ST    R1,SMAPIPRM           STORE IN SMAPI PARMLIST
        MVC   PA001_APTCB,APTCB@    GET APTCB ADDR
        MVC   CSRBOCOP,OCOPL@      GET OCO PARMLIST ADDR
        MVC   PA001_DATA_ID,PA001EYE DKNPA001 EYECATCHER
INITTWO DS    0H
        MVC   PA001_RECORD_@,=F'0'  INDICATE NO ADJ RECORD RETURNED
        MVI   PA001_KEY_ID+1,PA001_KEY_USTRING
        MVC   PA001_KEY,DATASTRG    MOVE STRING NAME TO KEY FIELD
        MVC   CSRBACT,ACTION        PRAD ACTION
CALLSMAP DS    0H
        LA    R1,SMAPIPRM           GET ADDR OF SMAPI PARMLIST
        L     R15,SMAPIADR          GET API EP ADDR
        BASR  R14,R15              PASS CONTROL
        LTR   R15,R15              TEST FOR ERROR
        BNZ
...
...
...
PA001   DKNPA001  TYPE=DATA        PRAD REQUEST PARM LIST
CSRBR   DKNCSRBA  TYPE=DATA

```

Figure 5-2. Assembler Example of Application Programming Interface Invocation



Figure 5-3 shows a COBOL example of an application programming interface invocation:

```

COPY DKNPA04C.
...
...
...
COPY DKNSMRB.
COPY DKNPA01C REPLACING ==01 :PA001:== BY ==02 PA001==
                        ==:PA001:== BY ==PA001==.
...
...
...
MOVE 'DKNCSRB'           TO CSRB-STRUCTURE-ID
MOVE 'PRADSTRG'          TO CSRB-OBJECT
SET CSRB-REQUEST-AREA   TO ADDRESS OF PA001-PARMLST
SET CSRB-RETURN-DATA    TO ADDRESS OF CSRB-RETURN-AREA
SET PA001-APTCB         TO ADDRESS OF APTCB
SET CSRB-OCO-PLIST      TO ADDRESS OF OCOPARM
MOVE WS-PA001-NAME      TO PA001-NAME
SET PA001-RECORD-ADDR   TO NULLS
MOVE LOW-VALUES         TO CSRB-SUBSS
                        CSRB-OBJECT-ID
                        CSRB-RETURN-OBJID
MOVE ZEROS              TO CSRB-RETURN-CODE1
                        CSRB-RETURN-CODE2
MOVE LOW-VALUES         TO PA001-USER-FLAGS
SET CSRB-RETURN-RSVD1   TO NULLS
SET CSRB-RETURN-RSVD2   TO NULLS
...
...
...
SET PA001-KEY-USTRING   TO TRUE
MOVE WS-PRAD-KEY        TO PA001-KEY
MOVE 'DELETE '         TO CSRB-ACTION
...
...
...
CALL 'DKNSMAPI' USING CSRB
MOVE CSRB-RETURN-CODE1  TO PA004-RETURN-CODE
MOVE CSRB-RETURN-CODE2  TO PA004-REASON-CODE.

```

Figure 5-3. COBOL Example of Application Programming Interface Invocation

## PRAD Application Programming Interface Input

Figure 5-4 presents the valid Objects and Actions supported by the PRAD service processors.

Figure 5-4. PRAD Service Processor Object/Action Definitions. This table presents the valid Objects and Actions supported by the Propagation of Adjustments Service Processors.

Object	Actions
PRADSTRG	<p>Listed below are the valid actions for the PRADSTRG Service Processor.</p> <p><b>DELETE</b> Requests PRAD to logically mark the adjustments on the current string as deleted.</p> <p><b>INQUIRY</b> Requests the PRAD API to determine if adjustments exist for the specified key.</p> <p><b>UPDATE</b> Requests the PRAD API to place adjustment records, from a string, into the Adjustment Repository.</p>
PRADITEM	<p>Listed below are the valid actions for the PRADITEM Service Processor.</p> <p><b>OPEN</b> Requests the PRAD API to establish a connection to the PRAD Adjustment Repository.</p> <p><b>CLOSE</b> Requests the PRAD API to terminate a connection to the PRAD Adjustment Repository.</p> <p><b>COMMIT</b> Requests the PRAD API to commit all adjustments marked as propagated, for this open connection to the Adjustment Repository.</p> <p><b>UPDATE</b> Requests the PRAD API to place an adjustment record into the Adjustment Repository.</p> <p><b>GET</b> Requests the PRAD API to retrieve an adjustment from the Adjustment Repository.</p> <p><b>GETORIG</b> Requests the PRAD API to retrieve the original item associated with the last retrieved adjustment record.</p> <p><b>GETPRIOR</b> Requests the PRAD API to retrieve the previous adjustment associated with the last retrieved adjustment record.</p> <p><b>GETPROP</b> Requests the PRAD API to retrieve an adjustment for a specific Cycle/Item Sequence Number (ISN) from the Adjustment Repository and to mark the item as propagated. This request is the same as specifying a GET followed by a PROPAGAT.</p> <p><b>GETNEXT</b> Requests the PRAD API to retrieve the next sequential adjustment contained in the Adjustment Repository.</p> <p><b>PROPAGAT</b> Requests the PRAD API to mark the current adjustment as propagated in the Adjustment Repository.</p> <p><b>DELETE</b> Requests PRAD to logically mark the current adjustment as deleted.</p> <p><b>REMOVE</b> Requests the PRAD API to physically delete all adjustments by cycle.</p>

## Request Parameter List (DKNPA001 Structure ID)

All PRAD Service Processor requests *must* pass a Request Parameter List (RPL) in the Request Area field of the Enhanced System Manager Service Request Block. The RPL contains information required (for example, buffer addresses and flag values) to successfully complete the service request. Figure 5-5 details the contents of the Request Parameter List (DKNPA001). For more information, refer to “Data Areas for Propagation of Adjustments” on page A-1.

*Figure 5-5 (Page 1 of 2). PRAD Application Programming Interface Request Parameter List. This table presents the layout and characteristics of the data elements required for communication with the PRAD Application Programming Interface.*

Name	Type, Length	I/O <sup>1</sup>	Description
Structure Identification	char, 8	I	Constant value of DKNPA001. It is the data structure identifier.
Structure Length	fullword, 4	I	Constant value of 60. It is the length of this data structure.
Key Id	halfword, 2	I	Identifies the type of key referenced by the <i>Key</i> field.  <b>0</b> Item Sequence Number Adjustment <b>4</b> Item Sequence Number Original <b>8</b> Uncompressed String Name <b>12</b> Cycle  See “Propagation of Adjustments Application Programming Interface Keys” on page 5-13 for the format of different key types.
String Name (compressed format)	byte, 9	I	The CPCS-I String Name that is the source of the adjustment to be updated in the Adjustment Repository.
Bank Id	char, 3	I	The Bank Number of the string with which this adjustment is associated.
Key	byte, 17	I	Provides the Key associated with an action. The format of the Key is interpreted according the value of the <i>Key ID</i> field.
CPCS-I Flags	Byte, 1	I/O	Provides information on base CPCS-I application tasks that have “propagated” an adjustment as an output of their processing. For example, MCRE is assigned a bit in this field to indicate that, during its extraction process, a particular adjustment was recognized and placed on its output file. See Figure 5-10 on page 5-15 for the complete description of this field.

*Figure 5-5 (Page 2 of 2). PRAD Application Programming Interface Request Parameter List. This table presents the layout and characteristics of the data elements required for communication with the PRAD Application Programming Interface.*

Name	Type, Length	I/O <sup>1</sup>	Description
User Flags	Byte, 1	I/O	Provides information for user CPCS-I application tasks that desire to maintain the propagated status of adjustments processed. This field is provided solely for user-written applications and no assumptions are made to any values contained.
Cycle	byte, 2	I	CPCS-I cycle from which the adjustment was extracted and placed into the Adjustment Repository.
Reserved	Byte, 1	N/A	Provides an area for alignment that is reserved for use by IBM.
Adjustment Record Address	fullword, 4	I	Provides the address of the PRAD Adjustment Repository Record (DKNPA002).
Application Task Control Block	fullword, 4	I	Provides the Application Task Control Block.
Reserved	byte, 4	N/A	Provides an area that is reserved for future use by IBM.

## Object/Action Pairs and Request Parameter List (RPL) Fields

Each action requires specific information be placed in the Request Parameter List (RPL). See Figure 5-6 for a listing of fields and their contents required for each PRADITEM action.

*Figure 5-6 (Page 1 of 3). PRAD Request Parameter List Action/Required Fields Matrix. This table presents the RPL fields required for each Object/Action pair for the PRADITEM Service Processor.*

Action	Required RPL Fields	Contents
OPEN	None	Not applicable
CLOSE	None	Not applicable
COMMIT	None	Not applicable
UPDATE	Adjustment Record Address String Name	The address of the variable portion of the DKNPA002 record. The string name, in compressed format, that is the source of the adjustment information to be updated.

<sup>1</sup> Input (I) fields contain values for the Application Programming Interface on entry and must not be modified. Output (O) fields are for values returned by the API and are not initialized on input. Input/Output (I/O) fields contain values for the API on entry, and the API also returns values in them.

Figure 5-6 (Page 2 of 3). PRAD Request Parameter List Action/Required Fields Matrix. This table presents the RPL fields required for each Object/Action pair for the PRADITEM Service Processor.

Action	Required RPL Fields	Contents
	Bank Id	The bank id associated with the string which contains the adjustment to be updated.
	Cycle	The CPCS-I cycle associated with the string which contains the adjustment to be updated.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	Application Task Control Block	The Application Task Control Block
GET	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	Application Task Control Block	The Application Task Control Block
GETORIG	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	Application Task Control Block	The Application Task Control Block
GETPRIOR	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	Application Task Control Block	The Application Task Control Block
GETNEXT	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.

*Figure 5-6 (Page 3 of 3). PRAD Request Parameter List Action/Required Fields Matrix. This table presents the RPL fields required for each Object/Action pair for the PRADITEM Service Processor.*

Action	Required RPL Fields	Contents
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	Application Task Control Block	The Application Task Control Block
GET+PROP	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	CPCS-I Flags	The CPCS-I application propagation flags to be set. See Figure 5-10 on page 5-15 for the flag values.
	User Flags	The user-written application propagation flags to be set.
	Application Task Control Block	The Application Task Control Block
PROPAGAT	Adjustment Record Address	The address of where to place the DKNPA002 record.
	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
	CPCS-I Flags	The CPCS-I application propagation flags to be set. See Figure 5-10 on page 5-15 for the flag values.
	User Flags	The user-written application propagation flags to be set.
	Application Task Control Block	The Application Task Control Block
DELETE	Key Id	Identifies the contents of the <i>Key</i> field. See Figure 5-5 on page 5-7 and Figure 5-9 on page 5-14 for permitted values.
	Key	The retrieval key as specified by the <i>Key Id</i> field.
REMOVE	Key Id	Indicates that the adjustment record key is the CPCS-I cycle.
	Key	The retrieval key as specified by the <i>Key Id</i> field.

Figure 5-7 presents the RPL fields required for each PRADSTRG action.

*Figure 5-7. PRAD Request Parameter List Action/Required Fields Matrix. This table presents the RPL fields required for each Object/Action pair for the PRADSTRG Service Processor.*

Action	Required RPL Fields	Contents
DELETE	Key Id	Indicates that the <i>Key</i> field contains a string name (uncompressed)
	Key (String Name)	An uncompressed string name that is to have all associated adjustments marked as deleted.
INQUIRY	Key Id	Indicates that the <i>Key</i> field contains a string name (uncompressed)
	Key (String Name)	An uncompressed string name that is to have all associated adjustments marked as deleted.
UPDATE	Key Id	Indicates that the <i>Key</i> field contains a string name (uncompressed)
	Key (String Name)	A compressed or uncompressed string name that is to have all associated adjustments marked as deleted.

---

## Primary Return Codes

The PRAD “Primary Return Code” value is contained in the Return Code 1 field of the Enhanced System Manager Service Request Block, Return Data Area. This variable contains information pertaining to the success or failure of the PRAD service request. See “Extended Return Codes” for additional information these fields contain. The following list details the possible primary return codes:

- 0** Request completed successfully
- 4** Warning, request completed successfully.
- 8** Error, request did not complete successfully.
- 12** Severe Error, request did complete successfully.
- 16** Irrecoverable Error, processing is unable to continue.

---

## Extended Return Codes

The PRAD “Extended Return Code” value is contained in the Return Code 2 field of the Enhanced System Manager Service Request Block, Return Data Area. This variable contains additional information pertaining to the success or failure of the PRAD service request.

*Figure 5-8. PRAD General-Use Programming Interface Extended Return Codes. This table presents the base set of extended return codes which can be returned by the Propagation of Adjustments Application Programming Interface.*

Extended Return Code	Primary Return code	Applicable Actions	Description
0	0	All	No information available.
1	8	All but OPEN	No connection to the Adjustment Repository exists.
2	8	OPEN	An existing connection to the Adjustment Repository already exists for this task.
3	8	All	The action specified is not valid for the Object.
4	8	INQUIRY, GET, GETPRIOR, GETNEXT, GET+PROP, PROPAGAT, REMOVE, DELETE	The key type specified by the <i>Key Id</i> field is not valid for the requested function specified by the Request Code field.
5	4	INQUIRY, GET, GETPRIOR, GETNEXT, GET+PROP, PROPAGAT, REMOVE, GETORIG	<p>No adjustment exists to satisfy this request. The meaning of this code is dependent on the type of “GET” service requested.</p> <p><b>Get (GET), Get Original (GETORIG)</b> No adjustment exists for the Item Sequence Number specified by the Key Address field.</p> <p><b>Get Prior (GETPRIOR)</b> No more adjustment exists for the Item Sequence Number specified by the Key Address field in the previous GET request.</p> <p><b>Get Next (GETNEXT)</b> End of adjustment data. No more adjustments exist.</p> <p><b>Get and Propagate (GET+PROP)</b> No adjustment exists for the Item Sequence Number specified by the Key Address field.</p>
6	4	All	The user exit returned a non-zero return code.
7	16	All	A user exit ended abnormally.
8	16	All	PRAD ended abnormally.
9	8	All	The Request Parameter List had an invalid parameter.
10	16	All	Cell pool services issued a bad return code.
11	16	All	System Manager issued a bad return code.



## Propagation of Adjustments Application Programming Interface Keys

Several PRAD Application Programming Interface request codes require an associated key to successfully service the request. “Propagation of Adjustments Application Programming Interface Keys” details the format of each supported key type while Figure 5-9 on page 5-14 displays which key types are supported for each request code. Passing an invalid key type, identified in the *Key Id* field of the Request Parameter List, results in a return of “8” and “4” in the Primary and the Extended Return Code fields, respectively.

The cycle/item sequence number key is *cciiii*.

Where:

- cc* The current cycle, right-justified and padded with EBCDIC (X'40' blank) on the left. For example, a cycle of 8 would be 'b8', or in hexadecimal, X'40f8'.
- iiii* The Item Sequence Number (ISN) in unsigned, packed decimal format. This must be the least significant 5 bytes of the 6-byte DISEQ12 field contained in the mass dataset record definition.

The uncompressed string name key is *eeeepp1p2p3p4tsss*.

Where:

- eeee* 4-byte value, entry number in EBCDIC format
- pp* Byte value, pass number in EBCDIC format
- p1* 2-byte value, first pass pocket history in EBCDIC format
- p2* 2-byte value, second pass pocket history in EBCDIC format
- p3* 2-byte value, third pass pocket history in EBCDIC format
- p4* 2-byte value, fourth pass pocket history in EBCDIC format
- t* Byte value, string type in EBCDIC
- sss* 3-byte value, subset number in EBCDIC format.

## PRAD–Programming Information

*Figure 5-9. PRAD Application Programming Interface Request Code to Key Type Matrix. This table presents a matrix which details the valid key types for each request code supported by the Propagation of Adjustments Application Programming Interface.*

Service Processor	Action	Key Type		
		Item Sequence Number	Cycle	Uncompressed String Name
PRADITEM	OPEN	No	No	No
	CLOSE	No	No	No
	COMMIT	No	No	No
	UPDATE	Yes	No	No
	GET	Yes	Yes	Yes
	GETORIG	No	Yes	Yes
	GET+PROP	Yes	No	No
	GETNEXT	No	Yes	Yes
	GETPRIOR	Yes	Yes	Yes
	PROPAGAT	Yes	No	No
	DELETE	Yes	No	No
	REMOVE	No	Yes	No
	PRADSTRG	UPDATE	No	No
INQUIRY		No	No	Yes
DELETE		No	No	Yes

## PRAD CPCS-I Propagation Flags

The base CPCS-I system application tasks provide a set of flags to indicate the propagation status of adjustments. Only CPCS-I base application tasks may use this set of flags, which must *not* be modified by user-written application tasks. Figure 5-10 details the current bit assignments within this field.

---

<b>Bit 0 (X'80')</b>	Adjustment has been propagated by MCRE
<b>Bit 1 (X'40')</b>	Adjustment has been propagated by RMIT
<b>Bit 2 (X'20')</b>	Adjustment has been propagated by KILL
<b>Bit 3 (X'10')</b>	Reserved for future use by IBM
<b>Bit 4 (X'08')</b>	Reserved for future use by IBM
<b>Bit 5 (X'04')</b>	Reserved for future use by IBM
<b>Bit 6 (X'02')</b>	Reserved for future use by IBM
<b>Bit 7 (X'01')</b>	Update time.

---

*Figure 5-10. Definition of CPCS-I Flags Field. This figure details the values and associations of the bit values used to indicate adjustment propagation status of an item by base CPCS-I application tasks. This flag is reserved for use by CPCS-I.*

## PRAD Adjustment Record (DKNPA002 Structure ID)

*Figure 5-11 (Page 1 of 2). PRAD Adjustment Record in General-Use Programming Interface Format. This table represents the format and characteristics of the adjustment record passed between the PRAD component and the calling applications. This record is also passed to each of the four PRAD user exits, and is the format of the records contained in the audit trail report file.*

Name	Type, Length	I/O <sup>2</sup>	Description
Structure Name	Char, 8	I	Constant value of DKNPA002. It is the data structure identifier.
Reserved	Halfword, 2	I	Reserved for IBM
CPCS-I Task Flags	Byte, 1	I	8 bits, each of which represent the CPCS-I task that may have propagated the adjustment. A value of "1" indicates that the adjustment has been propagated by the corresponding task. More than one bit may be on one time (more than one task may propagate an adjustment). To refer to which bits are assigned to tasks, see Figure 5-10.
User-written Task Flags	Byte, 1	I	Provides information for user CPCS-I application tasks that desire to maintain the propagated status of adjustments processed. This field is provided solely for user-written applications and no assumptions are made to any values contained.
Bank Number	Char, 3	I	Bank Number of associated adjustment, in character format

Figure 5-11 (Page 2 of 2). PRAD Adjustment Record in General-Use Programming Interface Format. This table represents the format and characteristics of the adjustment record passed between the PRAD component and the calling applications. This record is also passed to each of the four PRAD user exits, and is the format of the records contained in the audit trail report file.

Name	Type, Length	I/O <sup>2</sup>	Description
Deleted-Flag	Byte, 1	N/A	Indicates deleted adjustment.
Compressed Mass Dataset Record Offset	Halfword, 2	I/O	Offset within this record at which the compressed Mass Dataset Record starts.
Uncompressed Mass Dataset Record Offset	Halfword, 2	I/O	Offset within this record at which the uncompressed Mass Dataset Record starts.
Reserved	Fullword, 4	N/A	Reserved for IBM.
CPCS-I Time Stamps	Doubleword, 64	I	8 doublewords, each of which correspond to the relative propagation bit of "CPCS-I Task Flags," earlier in this table. These doublewords contain time stamps that indicate when the adjustment was propagated by the CPCS-I task indicated by "CPCS-I Task Flags."
User Time Stamps	Doubleword, 64	I	8 doublewords, each of which corresponds to the relative propagation bit of "User-Written Task Flags," above. These doublewords contain time stamps that indicate when the adjustment was propagated by the user-written task indicated by "User-written Task Flags."
Reserved	Byte, 8	N/A	Reserved for IBM
Compressed Mass Dataset Image	Variable, (derived from XREC)	I	Length is derived from MDXREC.
Uncompressed Mass Dataset Image	Variable, (derived from XREC)	I	Length is derived from MDXREC.

## Security

IBM recommends that the PRAD repository maintenance function be restricted by way of the CPCS-I Security facility. The PRAD repository maintenance option results in deleted adjustment records and should be restricted to privileged users.

<sup>2</sup> Input (I) fields contain values for the Application Programming Interface on entry and must not be modified. Output (O) fields are for values returned by the API and are not initialized on input. Input/Output (I/O) fields contain values for the API on entry, and the API also returns values in them.

## Creating a Propagation of Adjustments Audit Trail Report

The PRAR task uses DKNMDXR to format the print lines of the PRAD Audit Trail report in order to be compatible with variable mass data set sizes (expansions). The following report layout (for the UK) is for informal use only; when you change the MDS record, use the MDX macro. For more information about the MDX macro, see the *CPCS-I Customization Guide*.

DKNPRAR 1 Propagation of Adjustments Audit Trail											2 PAGE nnn1										
Cycle: nn 3																				4 Date: xx/xx/xx	
Bank: ALL 5																				6 Time: hh:mm:ss	
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21							
Item Seq Number	Sort Code	Account Number	Tran Code	Amount	CPU	TYPEI	PKT	Serial Reference	Adj Type	CPCS MRKrrrrr	Tsk USER	Tsk Tsk	Adj Adj	Adj Adj	Adj Adj						
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	01	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	01	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	01	999999999999	aaa	Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	YY.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	04	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	04	999999999999	aaa	YY.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	04	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
Bank 001 22				Total Adjustments: 1234567890 99 + 23											24 Total Propagated Adjustments: 123457890 99 +						
Cycle nn 25				Total Adjustments: 1234567890 99 + 26											27 Total Propagated Adjustments: 1234567890 99 -						
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	06	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	06	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	06	999999999999	aaa	YY.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999									ORIGINAL			
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	.Y.....	99	XX/XX	99:99:99							
9999999999	999999	99999999	99	999999	99	9999	aaaaaaaa	R	999999999999	aaa	YY.....	99	XX/XX	99:99:99							
Bank 333				Total Adjustments: 9999999999 99 -											Total Propagated Adjustments: 9999999999 99 +						
Cycle nn				Total Adjustments: 9999999999 99 +											Total Propagated Adjustments: 9999999999 99 -						
End of Report																					

Figure 5-12. Report Layout for the PRAD Audit Trail Report (Sample UK MDS)

## Creating a PRAD Audit Trail Report

### **Field Definitions**

- 1** Module name of the task that created the report. This is always DKNPRAR.
- 2** Page number
- 3** Cycle represented by this report. This is ALL if the report was run by cycle.
- 4** Date that the report was printed
- 5** Bank number for which the report was run
- 6** Time that the report was printed
- 7** 12-digit Item sequence number
- 8** Sort code
- 9** Account number
- 10** Transaction code
- 11** Amount of the transaction
- 12** CPU Flags—a hexadecimal print of processing-unit flags:  
  
The first 2 digits (byte 1) are the hexadecimal print of the tracking number.  
The last 2 digits (byte 2) are the hexadecimal print of the adjustment type.  
The High Performance Transaction System Balancing package sets the following values for adjustments when a 99-M-string is written back to CPCS-I.  
  
C'F3' = Moved-To  
C'F4' = Change Old (original item)  
C'F6' = Insert  
C'F7' = Change New (adjusted item)  
C'F9' = Force Balance
- 13** Document type:
  - Bundle
  - Tracer
  - Block
  - DCV
  - AST
  - Sub-total voucher
  - Divider
  - Insert
  - Change
  - Move-To
  - Blank (indicates detail record).
- 14** Pocket to which document was selected. Reject pocket is indicated by R/b (b indicates a blank space).
- 15** Serial reference field
- 16** Adjustment type, represented by aaa:
  1. CHG (change)
  2. INS (insert)
  3. MVT (moved-to)
  4. FRC (force balanced)

- 17** CPCS-I tasks that have propagated the adjustment. The header has identifiers for M (MCRE), R (RMIT), K (KILL), and rrrr (reserved for future use by IBM).
- Where:
- Y indicates that the corresponding task has propagated the adjustment. More than one task may propagate an adjustment.
- 18** A "Y" under one of the above identifiers indicates that the corresponding user-written task has propagated the adjustment. More than one task may propagate an adjustment.
- 19** Identifies the order of adjustments applied to this particular item. Each adjustment supercedes the prior adjustments. Adjustments are not cumulative.
- 20** Identifies the date each adjustment was made. Used to validate back to cash letter creation date so that research may determine that adjustments were present at the time the cash letters were generated.
- 21** Identifies the time each adjustment was made. Used to validate back to cash letter creation time so that research may determine that adjustments were present at the time the cash letters were generated.
- 22** Identifies the bank represented by this summary line. The report is sorted by bank number. This summary line is printed each time there is a break in the bank number.
- 23** Total adjustments for the bank identified by #22.
- 24** Total propagated adjustments for the bank identified by #22.
- 25** Repeats the cycle number from the report heading.
- 26** Total adjustments for the cycle identified by #25.
- 27** Total propagated adjustments for the cycle identified by #25.





---

## Appendix A. Data Areas for PRAD

---

### Data Areas for Propagation of Adjustments

The data areas for PRAD are:

<b>Name</b>	<b>Description</b>
DKNPA01C	COBOL version of the Request Parameter List
DKNPA02C	COBOL version of the Adjustment Record
DKNPA03C	COBOL version of the User Exit Parameter List
DKNPA04C	COBOL version of the Return/Reason Codes
DKNPA001	Assembler version of the Request Parameter List and the Return/Reason Codes
DKNPA002	Assembler version of the Adjustment Record
DKNPA003	Assembler version of the User Exit Parameter List.

## Data Areas for PRAD

## Appendix B. Messages for PRAD

### Messages for Propagation of Adjustments

The messages for PRAD are as follows:

---

**PRD 10001** *aaaaaaaa* **FUNC=xxxxxxxx RC=nnnn**  
**RES=zzzz**

**Explanation:** This message is sent by a module, *aaaaaaaa*, when an error occurs while processing a PRAD action, *xxxxxxxx*. The return code, *nnnn*, and reason code, *zzzz*, indicate the type of error that occurred. See "Extended Return Codes" on page 5-11 for return code information.

**Operator Response:** Inform your CPCS-I programmer. Be prepared to give error message numbers and other information pertinent to the abnormal end.

---

**PRD 10002** **PRAD IS NOT ACTIVE.**

**Explanation:** A task has attempted to access the PRAD repository while it is not active.

**Operator Response:** Inform your CPCS-I supervisor. See "Installing the Propagation of Adjustments Feature" on page 2-3 for details on how to activate the PRAD feature.

---

**PRD 10003** *eeee-p-aa-bb-cc-dd-x-sss* **MAY HAVE**  
**ADJUSTMENTS;**  
**USE xxxx REVISED**

**Explanation:** The RMIT or KILL task, *xxxx*, has encountered a D, I, or M-string from an entry which contains adjustments in the PRAD repository.

**Operator Response:** Re-run the task selecting the revised option.

---

**PRD 10004** **PRAD FEATURE IS NOT INSTALLED.**

**Explanation:** A task has attempted to access the PRAD repository, but the feature has not be installed.

**Operator Response:** Inform your CPCS-I supervisor. See "Installing the Propagation of Adjustments Feature" on page 2-3 for details on how to install the PRAD feature.

---

**PRD 30001** **GET HEAP STORAGE FAILED;**  
**FC=xxxx**

**Explanation:** A task using the LE/370 language environment has attempted to obtain HEAP storage and failed. The function code, *xxxx*, indicates the type of error that occurred.

**Operator Response:** Inform your CPCS-I supervisor. See the *AD/Cycle LE/370 Debugging and Run-Time Messages Guide* for an interpretation of the function code.

---

**PRDPR90000900** **Prad Initialization successful.**

**Explanation:** The PRAD repository has been successfully initialized and is ready to process requests.

**Operator Response:** None

---

**PRDPR90030901** **Prad Initialization failed. Feature inactive.**

**Explanation:** The PRAD repository initialization has failed. No PRAD repository requests can be processed.

**Operator Response:** Inform your CPCS-I supervisor. Review PRAD install procedures to ensure that the PRAD feature was installed correctly.

---

**PRDPR90000902** **No xxxxxxxxxx Prad User Exit specified.**

**Explanation:** The PRAD user exit has not been specified in bank control file card 50, where:

Variable	Options
xxxxxxx	UPDATE
	GET
	PROPAGATE
	PRAR

**Operator Response:** None

---

**PRDPR90000904** **Prad xxxxxxxxxx User Exit yyyyyyyy Loaded.**

**Explanation:** The PRAD user exit specified has been successfully loaded and is ready to process requests, where:

Variables	Options
xxxxxxx	UPDATE
	GET
	PROPAGATE
	PRAR
yyyyyyy	DKNPDUE1
	DKNPDUE2
	DKNPDUE3
	DKNPDUE4

Or name specified in bank control file input card 50.

**Operator Response:** None

**PRDPR90030905 Prad xxxxxxxxxx User Exit yyyyyyyy Load Failed.**

**Explanation:** The specified PRAD user exit has not been loaded, where:

Variables	Options
xxxxxxx	UPDATE
	GET
	PROPAGATE
	PRAR
yyyyyyy	DKNPDUE1
	DKNPDUE2
	DKNPDUE3
	DKNPDUE4
	Or name specified in bank control file input card 50.

**Operator Response:** Inform your CPCS-I supervisor. Check run time libraries to ensure that user exit, yyyyyyyy, exists.

**PRDITEM 30906 Prad Database Access Error in xxxxxxxx.**

**Explanation:** The PRAD repository ITEM service processor was unable to allocate Cell Pool storage for module, xxxxxxxx.

**Operator Response:** Inform your CPCS-I supervisor. This is a severe error. Be prepared to give error message numbers and other information pertinent to the abnormal end.

**PRAR 00001 xxxxxxxxxx (yyyyyyy) COMPLETED SUCCESSFULLY**

**Explanation:** This message indicates that either the PRAR Audit Trail or Repository Maintenance subtask has ended successfully, where:

Variables	Options
xxxxxxx	AUDIT TRAIL
	MAINTENANCE
yyyyyyy	DKNPRO22
	DKNPRO23

**Operator Response:** None

**PRAR 00002 REPOSITORY MAINTENANCE REQUEST CANCELLED.**

**Explanation:** The request to do repository maintenance has been cancelled by the user.

**Operator Response:** None

**PRAR 00003 EXTRACT FILE xxxxxxxxxx, DD=yyyyyyy;**

**Explanation:** This message informs the user that the Audit Trail extract file DDNAME has been allocated or unallocated. This message is accompanied by message PRAR 00004, where:

Variables	Options
xxxxxxx	ALLOCATED
	UNALLOCATED
yyyyyyy	DDNAME of extract file

**Operator Response:** None

**PRAR 00004 DSN=xxx...xxx**

**Explanation:** This message informs the user of the extract file's data set name. This message is accompanied by message PRAR 00003, where:

Variables	Options
xxx...xxx	Data set name of extract file

**Operator Response:** None

**PRAR 10001 (xxxxxxx) NOT AUTHORIZED FOR PRAD MAINT.**

**Explanation:** This user ID, xxxxxxxx is not authorized to use the PRAD repository maintenance function.

**Operator Response:** Inform your CPCS-I supervisor for security access if you need to use the PRAD repository maintenance function.

**PRAR 10002 NO ADJUSTMENTS IN REPOSITORY FOR CYCLE xx**

**Explanation:** There are no adjustment records in the PRAD repository for the cycle specified (xx).

**Operator Response:** None. If you were expecting adjustment records to be present for the cycle specified, check the scroll log to make sure the PRUP task has run successfully for the 99-M adjustment string(s).

**PRAR 30001 ERROR IN xxxxxxxxxx (yyyyyyy), RC=nnnn**

**Explanation:** An error has occurred in one of the PRAR subtasks. This message may also be accompanied by other messages, where:

Variables	Options
xxxxxxx	SCREEN-DRIVER
	AUDIT-TRAIL
	MAINTENANCE
yyyyyyy	DKNPRO21
	DKNPRO22
	DKNPRO23
nnnn	One of the return codes as shown in "Primary Return Codes" on page 5-11.

**Operator Response:** Correct problem and rerun PRAR.

---

**PRAR 30002 INVALID SM START PARM #n,  
xxxxxxxxxxxx**

**Explanation:** PRAR was started by System Manager with invalid start parameters specified, where:

Variables	Options
n	1 - 5, the positional number of the start parameter
xxxxxxxxxxxx	<b>CYCLE NBR</b>
	<b>BANK NBR</b>
	<b>REPORT PARM</b>
	<b>FILE PARM</b>
	<b>MAINT PARM</b>

**Operator Response:** Inform your CPCS-I supervisor. Correct the start parameters in the System Manager task profile for PRAR.

---

**PRAR 30003 DD=yyyyyyyyy (xxxxxxxxx) FAILED,  
RC=nnnn**

**Explanation:** The allocation or deallocation of the Audit Trail extract file has failed. This message is accompanied by message PRAR 00004, where:

Variables	Meaning
xxxxxxxxxxxx	ALLOCATE
	UNALLOCATE
yyyyyyyyy	DDNAME of extract file
nnnn	Return code from DKNTDYNA

**Operator Response:** Inform your CPCS-I supervisor. Check DSAT entries for PRAR (PRARDSAT) for possible mismatch of DDNAMEs.

---

**PRAR 30004 MDXR (xxxxxxxxx) REQUEST FAILED  
RC=nnnn**

**Explanation:** The request to the DKNMDXR report formatter has failed, where:

Variables	Meaning
xxxxxxxxxx	OPEN
	CLOSE
	PRINT-ZD
	PRINT-MSG
nnnn	Return code from DKNMDXR

**Operator Response:** Inform your CPCS-I supervisor. See the "Printing Reports with DKNMDXR" section of the *CPCS-I Programming Guide*. for more information concerning the DKNMDXR return codes.

---

**PRAR 30005 USER EXIT (xxxxxxxxx) FAILURE,  
FUNC=yyyy**

**Explanation:** The request to the PRAR user exit failed, where :

Variables	Meaning
xxxxxxxxxx	PRAR user exit name (Card 50 in bank control file)
yyyy	OPEN – user exit initialization PROC – process an adjustment record TERM – user exit termination

**Operator Response:** Inform your CPCS-I supervisor.

---

**PRAR 30006 ERROR ON BCF, NO PRAR USER  
EXIT LOADED.**

**Explanation:** The PRAR user exit specified in bank control file input card 50 has not been loaded. A problem exists in accessing the BCF bank control file.

**Operator Response:** Inform your CPCS-I supervisor. Correct bank control file access problem.

---

**PRUP 00001 PROCESSING COMPLETE**

**Explanation:** PRUP has processed the adjustment 99-M string(s) successfully. The adjustments are now in the PRAD repository.

**Operator Response:** None

---

**PRUP 10003 ERROR(S) DETECTED. PLEASE  
CORRECT. (ERRORS ARE  
HIGHLIGHTED)**

**Explanation:** Invalid data was entered on the PRUP screen.

**Operator Response:** Correct the highlighted data. The string name may be in an invalid format or may not be on the MDS (mass data set). The format is:

Options	Meaning
<b>ACTION</b>	A or D
<b>STRING NAME</b>	eeee-p-aa-bb-cc-dd-x-sss

---

**PRUP 10004 ERROR(S) STILL EXIST ON SCREEN.  
PLEASE CORRECT AND RESUBMIT.**

**Explanation:** Invalid data was entered on the PRUP screen and the submit key (PF4) was pressed. The string name may be in an invalid format or may not be on the MDS (mass data set). The format is:

Options	Meaning
<b>ACTION</b>	A or D
<b>STRING NAME</b>	eeee-p-aa-bb-cc-dd-x-sss

**Operator Response:** Correct the highlighted data and resubmit the request.

---

**PRUP 10005 NOTHING WAS ENTERED. ENTER DATA BEFORE SUBMITTING.**

**Explanation:** No data was entered on the PRUP screen and the submit key (PF4) was pressed. The format is:

<b>Options</b>	<b>Meaning</b>
<b>ACTION</b>	A or D
<b>STRING NAME</b>	eeee-p-aa-bb-cc-dd-x-sss

**Operator Response:** Enter correct data and resubmit the request.

---

**PRUP 10006 INVALID START PARAMETERS ENTERED.**

**Explanation:** Invalid parameters were entered on the PRUP start command. The PRUP input screen can be used to input the action and string name.

**Operator Response:** Enter the correct data and submit the request. The format is:

<b>Options</b>	<b>Meaning</b>
<b>ACTION</b>	A or D
<b>STRING NAME</b>	eeee-p-aa-bb-cc-dd-x-sss

---

**PRUP 10007 INVALID STRING TYPE. STRING TYPE MUST BE M.**

**Explanation:** A string with a string type other than M was entered.

**Operator Response:** Re-enter the string with a string type value of M.

---

**PRUP 30001 xxxxxxxx FAILED RC=nnnn REAS=zzzz**

**Explanation:** The PRUP task failed while processing either the add or delete action (request). The return code *nnnn* and reason code *zzzz* indicate the type of error that occurred. See "Extended Return Codes" on page 5-11 for return code information. This message is accompanied by message PRUP 300020, where:

<b>Variable</b>	<b>Meaning</b>
<i>xxxxxxx</i>	UPDATE – A action DELETE – D action
<i>nnnn</i>	PRAD error return code
<i>zzzz</i>	PRAD error reason code

**Operator Response:** Inform your CPCS-I programmer. Be prepared to give error message numbers and other information pertinent to the abnormal end.

---

**PRUP 30002 FOR STRING eeee-p-aa-bb-cc-dd-x-sss**

**Explanation:** This message contains the string name which PRUP was processing when it failed. This message is accompanied by message PRAP 300010.

**Operator Response:** None

## Glossary

This glossary defines important terms and abbreviations used in this manual. If you do not find the term you are looking for, refer to the Index or to the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

### A

**ABA.** American Bankers Association.

**ABA number.** (1) A numbering system devised by the ABA to provide exact identification of financial institutions. The code structure also identifies the Federal Reserve Bank and branch. (2) The MICR-inscribed field on a US document, containing the financial institution identification number.

**account number field.** An encoded field, on a check or a deposit slip, that indicates the account held by the drawer of the debit or the recipient of the credit.

**adjustment.** A change to a credit or debit document that adjusts the balance status of a deposit group (or transaction group).

**advice.** A letter that is sent to a financial institution or customer from whom checks have been received, advising that errors have been detected in the checks or in the listing that accompanied the checks.

**ALS.** Application Library Services.

**American Bankers Association (ABA).** Among the functions of this group is the specification of banking industry standards for US check-handling documents and procedures.

**amount due field.** This field is on some UK credit documents, typically utility payments, indicating the amount that is due for payment. It might or might not be the same as the actual amount field which will be encoded by the presenting bank when the credit is paid in.

**amount field.** An encoded field on an item that represents the amount of that item.

**Application Library Services (ALS).** See *ImagePlus HPTS Application Library Services*.

**application tasks.** Those application tasks that are delivered as part of the base CPCS-I program product or product feature.

**application program task control block (APTCB).** A CPCS-I area created by the applications task

(DKNATASK) for every active subtask in the system. This area contains operating system control blocks that are related to the subtask; it also contains addresses and constants used by the CPCS-I executive programs.

**APTCB.** Application program task control block.

**assist document (AST).** A document that accompanies incoming work and that supplies information about the work. A remittance/kill list is an example of an assist document.

**AST.** Assist document.

**automatic restart.** The process of restarting (continuing) an interrupted entry without having to find and rebatch any item.

### B

**balanced M-string.** The M-string that has been balanced by a balancing product. The balanced M-string is denoted by the string name *eeee-p1-p2-p3-99-t-sss*.

**balancing.** The act of bringing two sets of related figures into agreement (for example, reconciling accumulated-detail totals and input-control totals).

**bank control file (BCF).** A CPCS-I data set that contains control information for multiple bank processing.

**Bank Giro Credit (BGC).** A UK credit document that may be paid in only through a clearing bank. It may be encoded in MICR or in a mixture of MICR and OCR, but the format of the codeline is broadly similar to a check.

**base CPCS-I application tasks.** See *application tasks*.

**basic direct access method (BDAM).** An access method used to directly retrieve or update particular blocks of a data set on a direct access device.

**batch.** The lowest required level that has monetary control established by a control document. See also *Docket Control Voucher*.

**batch number.** The number that uniquely identifies a specific batch of documents.

**batch slip.** A level of control for balancing items. See also *batch* and *Docket Control Voucher*.

**BCF.** Bank Control File.

## BDAM • concurrent processing

**BDAM.** Basic direct access method.

**BGC.** Bank Giro Credit.

**block.** (1) A prime-pass control level consisting of one or more batches. In CPCS-I, this control level is used to total multiple batches. A block can also represent work from a specific source. (2) A data-processing term used to refer to a series of logical records stored contiguously on external storage devices. (3) To insert control documents in preparation for a prime-pass sorter run. See also *data preparation*.

**block slip.** A level of control for balancing batches. See also *block*.

**branch separators.** A UK term for user control documents used to separate work for different branches in on-us output pockets.

**buffer.** A main storage area used as a data-transfer area for physical records being read or written.

**bundle.** A bundle is a set of documents grouped together for processing and prefixed, for control purposes, by slips (for example, batch).

## C

**capture.** (1) To read the codeline that is inscribed on a document. (2) To make a digitized image of a document. In the HPTS system, full-item images can be captured by the Image Capture System attached to the document processor or by a low-speed scanner attached to a workstation.

**cash letter summary.** In the US, a listing that summarizes kill lists by giving monetary totals and item controls for each kill list. In the UK, this is referred to as a DCV Summary.

**CDM.** Codeline Data Matching.

**CDMP.** Codeline Data Matching Prime.

**CDMR.** Codeline Data Matching Rejects.

**check.** (UK = cheque) A draft drawn on a financial institution and payable on demand on or after the date indicated.

**check number.** See *serial field* or *reference*.

**Check Image Management System Data Base (CIMS Data Base).** A program in ImagePlus HPTS Application Library Services that stores, gets, and manages document images.

**cheque.** UK spelling of "check."

**CIMS.** Check Image Management System. See *Check Image Management System Data Base*.

**clearing house.** An organization, established by financial institutions in the same locality, through which checks and other instruments are exchanged and net balances settled.

**codeline data matching (CDM).** A method by which a computer system controls items on a detail level by comparing the internal data records from a previous pass with data that it reads on the current pass.

**codeline data matching prime (CDMP).** The process of performing codeline data matching during a CPCS-I prime pass. Document codeline data is matched against DEFT data transmitted from another bank or a branch of the processing bank. See also *document-based electronic funds transfer*.

**codeline data matching rejects (CDMR).** The process of performing codeline data matching on CPCS-I prime-pass rejects. Document codeline data is matched against Prime/HSRR codeline data that has been repaired (for example, in OLRR or HPTS key entry).

**codeline data record.** See *data record*.

**cold start.** An initiation of the CPCS-I region that causes the deletion of the previous contents of the mass data set and the control data sets.

**complete task status.** This indicates that this task processed successfully for this UOW. See also *task status*.

**complete UOW status.** This indicates that all tasks in the task list processed successfully or had a bypass status.

**component.** A set of modules that performs a major function within a system; for example, a compiler or a master scheduler.

**component internal data.** All data accessible to any modules within a particular component, but not accessible to any part of the system outside this component.

**concurrent kill.** Producing remittance/kill lists for kill pockets in an entry before the entire entry is processed. The concurrent kill feature is available only with subset processing.

**concurrent processing.** A system where the processing of prime capture work through subsequent processes (such as reject handling, rehandle sorting, or remittance printing) begins before completing capture for the whole entry.



**control block.** A storage area that a computer program uses to hold control information.

**control document.** An encoded document that contains control information, such as the total of the checks that the document controls, the source of the checks, and a code that describes the level of control.

**control slip.** See *control document*.

**control total.** The total value or item count for a group of documents.

**copy library.** A library that contains statements to be modified by the user, accessed by the assembler instruction copy, and inserted into some of the CPCS-I programs.

**correspondent financial institution.** A financial institution that carries a deposit balance for, or engages in an exchange of services with, another financial institution.

**CPCS-I.** Check Processing Control System International MVS/ESA.

**credit.** The opposite of a debit. Common examples are deposit slips and utility payments.

**cross record.** See *XREC*.

**cutoff.** (1) The financial institution's designated point for balancing or releasing work before processing continues. (2) The designated time after which the financial institution cannot accept work for processing.

**cycle.** (1) A group of work or an identification of a group of work processed completely as a single entity. (2) A convenient grouping of work. A cycle normally contains a variable number of entries.

## **D**

**DASD.** Direct access storage device.

**data preparation.** The preparation of documents for processing by a high-speed check-processing system.

**data record.** The electronic representation of the codeline captured from a check, deposit, debit, credit, or control document. The electronic representation can include additional data to help identify the record.

**data space.** An area of virtual storage that a program can ask the system to create. The area's size can range from 4K bytes to 2 gigabytes, according to the program's request. Unlike an address space, a data space contains only data. Program code cannot run in a data space. Unlike data in a Hiperspace, data in a data space is directly addressable.

**DCV.** Docket Control Voucher.

**DCV summary.** A listing that summarizes all of the kill bundles in a DCV summary report by giving monetary and item controls for each remittance list. See also *cash letter summary*.

**DCV summary report.** Report listing the group of items to be delivered to an endpoint. Grouping of the items is usually by kill bundle.

**debit.** A transaction that increases an asset or decreases a liability. In normal check-collection terminology, a check is considered a debit.

**deferred printing.** The method by which data is processed, transferred to a storage device, and later printed (as opposed to printing during the processing of data).

**DEFT.** Document-based Electronic Funds Transfer.

**DEFT input.** Electronically captured data that supports processing of paper documents in a codeline data-matching prime pass.

**deleted UOW status.** This indicates that the string associated with this UOW is deleted. No more processing can be done for this UOW.

**deposit slip.** A document that details a deposit. The total of the deposit is encoded on the deposit slip. A deposit is considered a credit.

**DFD.** Data Flow Diagram.

**direct access storage device (DASD).** A device in which access time is independent of the location of the data.

**distributed string (D-string).** The distribution task reads I-strings that the MICR task created and produces D-strings. Each D-string contains the records that correspond to all of the documents in a given pocket of the document processor.

**divider slip.** A control document that is used to separate kill bundles during machine sorting. It can also be used to support the resynchronization of codeline data matching during subsequent-pass processing.

**Docket Control Voucher (DCV).** A UK document used to prefix a batch of documents for exchange between clearing operations. A DCV is considered a Batch Slip by CPCS-I. See also *batch*.

**document-based electronic funds transfer (DEFT).** The transmission, reception, and processing of codeline data sent or received electronically from another

## document processor • funds availability

location together with the documents. The data is used in codeline data matching and reconciliation to reduce rejects and balance work.

**document processor.** A device that can read encoded characters from documents and sort the documents into multiple pockets.

**document processor station.** A work station consisting of a document processor and a terminal for operator communication.

**drawer.** The person on whose account a check is being drawn.

**D-string.** Distributed string.

## E

**ECDM.** Extended codeline data matching.

**enclosed and not listed.** A condition that exists when an item is in a batch of checks but is not listed on the incoming kill/remittance list or inscriber tape.

**encode.** To imprint a MICR field on a document. The CPCS-I database contains the information that is encoded. Synonymous with *inscribe*.

**encoder.** A machine that encodes or inscribes. Synonymous with *inscriber*.

**endorsement.** (1) The signature of the endorser; (2) the stamp of a financial institution or company.

**endorser.** (1) A person or financial institution, other than the maker, who presents a check for payment. (2) A device that stamps an endorsement.

**endpoint.** The destination of an item (debit or credit).

**enhanced reject processing.** The pockets used in this processing are alternate reject pockets, eligible to receive a reject item and/or an unencoded reject item. These pockets are defined in the J sort pattern definition record with values of J, E, and U respectively.

**entry.** A variable number of documents that are processed as a single group of work. Normally consists of a number of blocks and batches.

**entry number.** The number of the first tracer group within an entry.

**EPC.** Extended process control field.

**ERP.** Enhanced reject processing.

**error description.** The detailed description of an error created, detected, and corrected by the processing financial institution.

**exception printing.** The printing of only the data that requires action external to a computer.

**extended codeline data matching (ECDM).** A feature available on the 389x/XP Series document processors. It allows the matching criteria to be changed on a per-document basis (based on the perfectly read fields or on the number of digit errors in a field) and increases the chance of a successful match.

**extended process control field (EPC).** An optional encoded field that indicates special handling (such as return or truncation).

## F

**fine-sort.** (1) The sorting of items, for example, into account number order for filing. (2) The sorting of items for a single account into serial-number order as a customer service.

**fine sort group (FSG).** A group of documents that have been block-sorted under CPCS-I for fine sorting. Each FSG has a unique CPCS-I endpoint and does not enter fine sorting until all work for that FSG has been processed through all preceding passes.

**flip-flop.** An event that occurs when the volume to which you are writing a file becomes full. The writing continues on a new volume and the full volume is backed up.

**float.** The portion of a financial institution's total deposits, or of a depositor's account, that represents items (for example, checks) in the process of collection.

**flow code.** A 3-digit number (mnemonic) that represents an ordered list of tasks.

**flow control.** The pairing of a CPCS-I string with a task list through the specification of sort type, pass-pocket history, string type, and flow code.

**FSG.** Fine sort group.

**full-page printing.** A method of page formatting in which items are listed in as many columns as can be contained on the page (for example, the first 50 items in column 1, the second 50 in column 2, and so on).

**functional unit of work.** This unit of work corresponds to a CPCS-I string or subset string.

**funds availability.** The portion of the financial institution's total deposits or of a depositor's account that represents items (for example, checks) that have been collected and are now available. This includes cash deposited and checks drawn on the depositor's financial institution.

## G

**generated total.** The total value or item count of checks that are processed by the computer.

## H

**held task status.** This indicates that this task should be the next task to process, but a condition external to CPCS-I must complete first. See also *task status*.

**High Performance Transaction System (HPTS).** See *ImagePlus High Performance Transaction System*.

**high-speed reject re-entry.** The re-entering into the document processor of reconditioned documents that have previously been sorted to the system reject pocket (pocket 1-1).

**Hiperspace.** A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as a buffer. Like a data space, a Hiperspace holds only data, not common areas or system data; code does not execute in a Hiperspace. Unlike data in a data space, data in a Hiperspace is not directly addressable.

**holdover.** (1) Items that were not processed in time to meet their deadline. (2) Items that are held for the next processing cycle.

**HPTS.** High Performance Transaction System. See *ImagePlus High Performance Transaction System*.

**HSRR.** High-speed reject re-entry.

## I

**image.** The captured facsimile (picture) of an item represented in digital form suitable for computer processing and storage, and visual display to an operator.

**ImagePlus High Performance Transaction System (HPTS).** An IBM system that adds image processing capabilities to document processing.

**ImagePlus HPTS Application Library Services.** An IBM licensed program that supplies the HPTS system with services such as communication, data-storage management, recognition facilities, data compression, data reconstruction, and device support. The program consists of Image Host Application services, Image Processor Recognition Services, and Image Workstation Application Services.

**import/export.** The sending of information (export) from one system or application and the acceptance of information (import) by another system or application.

**inclearings/inwork.** A UK term describing checks and credits drawn on your financial institution. Similar to the term "on-us."

**incoming sequence number.** A number that defines the incoming sequence of an item within the input stream. This unique number is associated with the item throughout the whole cycle of computer processing.

**input string (I-string).** A string of documents created by the MICR task. On each document processor run, an I-string is created. The string includes every document read by the document processor, including control documents and rejected documents. Related information, such as the pocket selected, is also stored in each record. The string also includes internally generated control records.

**inscribe.** Synonym for *encode*.

**inscriber.** A machine that encodes and inscribes in a particular format. Synonym for *encoder*.

**interbank settlement sheet.** A UK interbank report, produced by Inwork DCV Reconciliation, summarizing the Inwork DCV totals and the settlement figure.

**Inwork.** A UK term for incoming on-us work from other banks or institutions.

**Inwork DCV Detail Report.** A UK term for a report produced by Inwork DCV Reconciliation for each responding bank listing the DCVs and WDs that are being returned.

**Inwork DCV Recapture File.** A UK term for a file created by Inwork DCV Reconciliation by recapturing the Inwork DCVs and WDs after balancing. This file is matched against the Inwork DCV Summary File to produce the Inwork DCV Reconciliation File.

**Inwork DCV Reconciliation File.** A UK term for a file created by Inwork DCV Reconciliation by matching the Inwork DCV Recapture File against the Inwork DCV Summary File.

**Inwork DCV Reconciliation Report.** A UK term for a report produced by Inwork DCV Reconciliation that lists the free and missing Inwork DCVs detected.

**Inwork DCV Summary File.** A UK term for a file created by DKNIDCS after the completion of Prime Balancing. It contains details of all DCVs and WDs captured in the Inwork cycle and is input to Inwork DCV Reconciliation.

**interface.** A named and shared boundary between two functional units, (for example, component interface, subcomponent interface) defined by functional characteristics, or other characteristics, as appropriate.

## invocation • magnetic ink character recognition (MICR)

**invocation.** Any method of starting a function within a component, subcomponent, or module, such as a direct call with parameters, use of a queue, or event control blocks (ECBs).

**inwork.** Checks and credits that are drawn on the financial institution that is processing them. Also termed "on-us."

**I-string.** Input string.

**item.** A check, deposit slip, or other machine-readable document.

**item-sequence number.** A number that defines the sequence of an item within the input stream. This unique number is associated with the item throughout the entire cycle of computer processing.

## J

**jam.** A condition that exists when items form a blockage anywhere in the transport mechanism of a document processor.

**JGC.** Joint Giro Credit.

**job control language (JCL).** A control language used to identify a job to an operating system and to describe the job's requirements.

**JCL.** Job Control Language.

**JES.** Job entry subsystem.

**job entry subsystem (JES).** A system facility for spooling, job queuing, and managing input and output.

**joggler/jogger.** A device that straightens and aligns items before high-speed sorting, principally to line up the lower edge and right side of a group of documents. This device is an integral component of some document processors.

**Joint Giro Credit (JGC).** A UK credit that may be paid in either through a clearing bank or through a post office. The two JGC types are (1) long joint giro, and (2) short joint giro. The only difference between the two types is that the long version has an Amount Due field and the short JGC does not.

## K

**kill.** To process items to a point where no further distribution is required. See also *remit*.

**kill bundle.** A group of items in a kill pocket, delineated by divider slips, that forms a batch or remittance to another bank. With concurrent kill, this group can span strings. See also *remittance list*.

**kill list.** A document that accompanies a kill bundle, listing detail and controls for the items.

**kill pass.** A pass on which items are distributed to their endpoint pockets.

**kill pocket.** A document-processor pocket assigned to items that are sent and remitted to another bank or destination without further sorting.

## L

**legal tender.** Any money that must, by law, be accepted in payment of debts. A personal check is not legal tender.

**link-edit.** To use a linkage editor to create a loadable computer program.

**listed and not enclosed.** A condition that exists when an item is listed on an incoming remittance/kill list or inscriber tape but is not enclosed in the kill bundle.

**logical unit (LU).** A port through which a user accesses SNA-network functions to communicate with another user on the network.

**low-speed transit.** The manual sorting and processing of checks.

**LU.** Logical unit.

**LU 6.2.** Logical unit 6.2 protocol.

**LU 6.2 protocol.** An SNA service that receives requests from users and from the system services control point. This service provides session management and other services for sessions between two logical units.

## M

**magnetic ink character recognition (MICR).** The reading of magnetically encoded data on the 5/8" clear band that runs along the bottom of a document. The MICR system uses ten specially coded digits and four special symbols.

**Management Information System (MIS).** A DB2 system that maintains data on overall check processing. This is a subcomponent of ImagePlus HPTS Application Library Services (IALS).

**manual restart.** The process of physically finding and rebatching, before resuming an interrupted entry, the items to be recaptured.

**mass data set (MDS).** A file that contains records of all active document strings. This file consists of two direct access data sets: a directory index and a data record set.

**master list.** A list of all items that are read during a computer pass.

**MDS.** Mass data set.

**merged string (M-string).** The M-string, produced by DKNMRGE, represents the merging of images from the prime-pass I-string with corrected reject data. Reports that result from the M-string let you reconcile and balance input to ensure that all items were captured.

**MICR.** Magnetic ink character recognition.

**microfilm number.** The assigned item number that is also captured on microfilm.

**MIS.** Management Information System.

**misread.** A condition that occurs when a document processor interprets a character as a good character other than that which actually appears on the document codeline. Synonymous with *substitution*.

**missort.** An item that is found in a pocket other than the pocket to which it was sorted. This might be the result of a misread.

**M-string.** Merged string.

**Multiple Virtual Storage (MVS).** An operating system that consists of MVS/System Product (MVS/SP)\*, MVS/ESA\*, and the MVS Data Facility Product operating on a System/370 processor.

## O

**OCR.** Optical character recognition.

**OLMS.** Online manual split.

**OLRR.** Online reject re-entry.

**online fine sort.** A computer-controlled sorting of documents (for example, checks) by either or both the account number and the serial number sequence for filing. This process commonly uses codeline data match techniques.

**online manual split (OLMS).** The process that sorts reject data from the MDS to produce remittance/kill lists and branch reports in the same sequence as manually sorted rejects.

**online reject re-entry (OLRR).** Manual entry or correction of MICR data through a display terminal.

**on-us.** Documents belonging to a bank that are sent to its clearing center from other banks or financial institutions. See also *inwork*.

**Optical character recognition (OCR).** Character recognition that uses optical means to identify graphic characters.

**optional field 1.** An optional, encoded field used by some US financial institutions for check truncation. It can also be used for other internal purposes.

**out-clearing.** A UK term meaning the sorting of documents to external destinations. The US term is *transit*. See also *outwork*.

**outgoing sequence number.** A sequence number or unique identification assigned to each item, identifying the kill bundle in which the item left the financial institution.

**outwork.** Documents that when processed leave the bank for collection from other institutions. See also *out-clearing*.

**Outwork DCV Detail Report.** A UK term for a report produced by Outwork DCV Reconciliation for each responding bank. It is essentially a listing of the Outwork DCV Reconciliation File.

**Outwork DCV File.** A UK term for a file produced by Remittance (Kill) processing. It is essentially an electronic version of the Outwork DCV Report and is used to power encode DCVs.

**Outwork DCV Interbank Settlement Sheet.** A UK term for a report produced by Outwork DCV Reconciliation for each responding bank, summarizing the agreed DCV totals and the figure for settlement.

**Outwork DCV Recapture File.** A UK term for a file created by Outwork DCV Reconciliation by recapturing the DCVs returned by other banks. This file is then

---

\* Trademark of IBM

## Outwork DCV Reconciliation File • reject string (R-string)

matched against the Outwork DCV Summary File created on the previous day.

**Outwork DCV Reconciliation File.** A UK term for a file created by Outwork DCV Reconciliation by matching the Outwork DCV Recapture File against the Outwork DCV Summary File.

**Outwork DCV Reconciliation Report.** A UK term for a report produced by Outwork DCV Reconciliation for each responding bank listing the missing and free DCVs detected.

**Outwork DCV Report.** A UK term for a report produced by Remittance (Kill) processing. It is similar to a CPCS-I cash letter and summarizes a number of kill bundles. It is not sent with the documents but is used to manually encode DCVs.

**Outwork DCV Summary File.** A UK term for a file produced by Remittance (Kill) processing. It contains a record for every Remittance (Kill) bundle processed and is grouped by endpoint within a cycle. It is used as input to Outwork DCV Reconciliation when the DCVs are returned by the responding bank on the following day.

## P

**pass.** A single reading and sorting of a group of checks and control documents on a document processor.

**pass-to-pass control.** A process that maintains the total amount and item control of a group of documents on subsequent passes, when control has been established on the previous pass.

**path.** The path of a functional unit of work is the ordered list of tasks processed for the associated CPCS-I string. See also *flow code* and *flow control*.

**pending status queue.** A first-in-first-out System Manager queue through which CPCS-I applications interface to the System Manager, in sequence, to perform UOW creations, deletions, inquiries, and updates.

**piggyback item.** An item that was missing from its assigned pocket in a sorter and sorted “free” to an unidentified pocket, as when one document attaches itself to or overlaps another during processing.

**pocket 1-1.** See *system reject pocket*.

**PRAD.** Propagation of Adjustments.

**presenting bank.** A UK term for the bank sending documents and DCVs and requesting funds for the DCVs.

**prime pass.** The first pass of an entry on a document processor.

**printing after the fact.** See *deferred printing*.

**process control field.** Used in the US by the payor bank to know which process applies to each item. In the UK this field is called *transaction code* and is used to identify document types.

**proof.** Receives checks that come from tellers, mail and night depository, and internal departments of the financial institution. Proof balances transactions and inscribes or encodes the monetary amount in MICR.

**proof of deposit.** The act of totalling items at the deposit level and ensuring that the total of the credits equals the total of the debits.

**propagation of adjustments.** The process of ensuring that adjustments made in Balancing and elsewhere are carried forward to kill/remittance and other system output processes.

## R

**RACF.** Resource Access Control Facility.

**RBA.** Relative block address.

**reconcile.** To find and correct the cause of a difference between two sets of totals.

**reconciliation.** See *balancing*.

**reconditioning.** The process of straightening folded items, inverting upside-down items, flipping reversed items, and removing any residual staples or rubber bands.

**reference.** A UK term for a field encoded on credit documents, corresponding to the 6-digit Serial field on debits. The Reference field may be up to 18 digits in length and (if printed in OCR) may contain alphanumeric characters.

**rehandle pocket.** A document processor pocket that receives items for multiple endpoints. Items directed to rehandle pockets are processed again on a later pass.

**reject.** A document that cannot be read in its entirety by a document processor or that fails certain editing checks. This document is normally directed to a special pocket called a reject pocket.

**reject string (R-string).** Strings that are created by the online reject re-entry task. Each R-string represents checks that have been re-entered online. R-strings are input to the DKNMRGE task.

**relationship.** Shows the parent/child hierarchy of units of work.

**relative block address (RBA).** In CPCS-I, the calculated location of a specific record.

**remit.** A UK term; to send items to another financial institution.

**remittance file.** A UK term for an MVS data set that is created by Remittance (Kill) processing. It is essentially an electronic version of the remittance list and may be used to support DEFT input processing at the receiving institution.

**remittance list.** A UK term for a CPCS-I Kill List that is produced to support negotiation and settlement of a batch of documents prefixed by a DCV. It is used for conventional interchange between clearing operations.

**repass.** See *rehandle pocket*.

**rerun.** A group of items that are sorted into a pocket on one pass and later brought into a document processor for more sorting.

**Resource Access Control Facility (RACF).** An MVS security subsystem that determines the validity of each operator's ID password and that controls operator access to application tasks and transactions.

**responding bank.** A UK term for the bank making payment on documents/DCVs received from the presenting bank.

**restart.** An initiation of the CPCS-I system after a system failure. A restart is generally used to start the system (after an abnormal end of a task) to cause the executive routines to re-establish the system to the status that existed before the failures.

**restart buffer.** An area where records are stored in an IBM 389x/XP Series document processor during online operations until they are sent to the host. The buffer is accessed during automatic restart.

**resynch document.** A control document used in DEFT processing to match DEFT data to the documents currently being processed on Prime and also used to separate and identify kill bundles on output.

**return item.** A check that is not honored by the maker's financial institution and that is returned to the depositor's financial institution.

**routing/transit number field.** An encoded check field that represents the financial institution on which the check is drawn. In the UK, this is referred to as the *Sort Code*.

**R-string.** Reject string.

## S

**SCI.** Stacker Control Instruction.

**scroll.** The ability to use the DKNSCRL application to page through or look at the scroll data set. This data set includes supervisor terminal messages and DKNATASK log messages.

**SDE.** String directory entry.

**separator.** See *divider slip*.

**sequence number.** A number, assigned to a document, that uniquely identifies its position in a group of incoming or outgoing work.

**serial field.** A UK term for the 6-digit field, (equivalent to the check number in the US), which is normally the serial number of a check. On credits, the same field is called a Reference and may be up to 18 digits in length.

**settlement.** The act of bringing sets of related figures from two financial institutions into agreement. Adjustments are made to offset the differences.

**simulated sorter.** A CPCS-I facility that allows a user to run MICR, using an input file without a physical sorter.

**slip.** A slip is a control document used to prefix bundles for control purposes.

**SMOF.** System Manager Online Functions.

**SNA.** Systems Network Architecture.

**sort code.** A UK term for the field (equivalent to the routing transit field in the US) which identifies the bank and branch to which a debit or credit item belongs. It is in the format *BB-bbbb*, where *BB* identifies the bank, and *bbbb* identifies the branch within that bank. It may be printed in MICR (on checks and some credits) or in OCR (on some credits). If printed in MICR, the two parts of the field are separated by a dash (SS4).

**sorter station (also document-processor station).** A work station consisting of a document processor and a terminal for operator communications. Synonym for document-processor station.

**sort pattern.** A table used by the sort routine to determine the pocket to which a check is to be directed.

**sort-pattern definition file.** A collection of records that contains control information that MICR in CPCS-I uses to set up and control document sorting; it also contains data about endpoints.

## sort routine • tracer group

**sort routine.** A time-dependent routine that does all processing required to direct a document to a specific document processor pocket.

**sort program.** A routine that performs all processing required to select a document to a pocket.

**spool data set.** A data set used to store printed output lines. Each spool (Simultaneous Peripheral Operations On-Line) data set is written by a CPCS-I application task and is read by the CPCS-I output writer as it is being printed.

**SSB.** String status block.

**SSM.** String segment map.

**Stacker Control Instruction (SCI).** SCI is the name of a language used to write programs to control the sorting of documents on a 389x document processor.

**statistics.** The processing of unit-of-work (UOW) data through a statistical program such as the ImagePlus Application Library Services (MIS) system. This term can also refer to the processing of unit-of-work data through a user-written statistical program.

**string.** The data records representing a group of items, for example, an I-string, a D-string, or an M-string. See related definitions for details.

**string segment map (SSM).** One of three types of segment maps in CPCS-I. Each string in the system is associated with a string segment map. Each bit in a map represents a segment of direct access storage.

**string status block (SSB).** This CPCS-I control block is maintained by the MDS programs for every open string.

**STV.** Subtotal voucher.

**subcomponent.** Functional subset of a component where subsetting is appropriate based on data use, logic flow, or other factors relating to modules.

**subcomponent internal data.** All data accessible to any modules within this particular subcomponent, but not accessible to any part of the system outside this subcomponent.

**subsequent pass.** A pass on which previously sorted items are resorted for further distribution.

**subset.** A defined portion of an entry, indicated by one or more tracer groups.

**subset processing.** Processing a portion of an entry beyond the document-entry step before the whole entry is run through the document processor.

**subset string.** A predefined group of data records that represents a portion of the physical items in an entry. A subset string can contain multiple tracer groups.

**substitution.** See *misread*.

**subtotal voucher (STV).** An optional UK document that can be inserted into a batch of documents to mark the point at which a cumulative subtotal is printed on the accompanying remittance list.

**supervisor.** (1) An MVS term used to refer to the system nucleus in internal storage. (2) A person responsible for operation of a financial institution area.

**supervisory terminal.** A special terminal or operating mode used in CPCS-I.

**System Manager.** A subsystem of CPCS-I that directs and controls the operations.

**System Manager Online Functions (SMOF).** A set of application-level tasks that monitor and modify the queues and databases of System Manager.

**system reject pocket.** The first physical pocket on the document processor. It is used by CPCS-I to hold machine and user-selected rejects.

**System Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration of, networks.

## T

**tab key.** A keyboard function key. The tab key causes the cursor to position to the next colon on the screen or to the top of the screen.

**task.** A CPCS-I application or function. User-written tasks must be in the CPCS-I BLDL list.

**task list.** The ordered list of tasks to be performed for a unit of work. It is determined by selecting the flow code for a given flow control record.

**task status.** A representation of what will happen, what is happening, or what happened during processing of this unit of work. Can be pending, ready, or complete. See related definitions for details.

**total system.** A system in which the computer is used for all phases of an operation.

**tracer.** A check-processing document used to provide pass-to-pass control.

**tracer group.** A grouping of documents between sets of tracers for control purposes. If subset processing is



in operation, this tracer group normally becomes a unit of work that can be processed independently of other units of work within that entry.

**tracer ID.** The tracer group and slip numbers corresponding to a tracer slip.

**transaction code.** A UK term for the 2-digit field that identifies debit, credit and control document types (similar to the Process Control Field in the US). A blank transaction code is a valid identifier for a check.

**transit.** The sorting of checks to external destinations. See also *out-clearing* and *outwork*.

## U

**unit of work (UOW).** A logical entity that the System Manager uses to track a piece of work through CPCS-I. It can be informational or functional. See also *functional unit of work*.

**UOW.** Unit of work.

**UOW status.** This status represents the state of a unit of work and its associated string. Can be pending, ready, or complete.

## V

**Virtual Storage Access Method (VSAM).** An access method for indexed or sequential processing of fixed or variable-length records on direct access storage devices.

**Virtual Telecommunications Access Method (VTAM).** A set of programs that control the communication between terminals and application programs.

**VSAM.** See *Virtual Storage Access Method*.

**VTAM.** See *Virtual Telecommunications Access Method*.

## W

**warm start.** An initiation of the CPCS-I system, causing the contents of the MDS and the control data sets to be retained. A warm start is generally used for restarting CPCS-I after a normal ending.

**WD (wrongly delivered).** A UK term for items (debits or credits, not DCVs) that have been dispatched to the wrong bank. They are returned rather than redirected.

**XREC.** The dynamic control block that maps the string data at various points in the system. It cross-records or

maps the string as it is in the data base, or as it is in the data space.

**work.** Any document or group of documents that CPCS-I processes.

**work flow.** An ordered list of tasks for a specific CPCS-I string. Each CPCS-I string must have a work flow.

## Z

**zero-balancing.** The procedure that ensures that generated totals for a group of items plus any documented errors minus the control total equals zero.

## Numerics

**3890/XP Document Processor.** A document processor in the 3890/XP Series of document processors that can read and sort documents at a rate of up to 2400 documents per minute.

**3890/XP Series document processors.** A series of high-speed document processors that can read and sort up to 1000, 1700, or 2400 documents per minute. These document processors include the IBM 3890/XP Document Processor, the IBM 3891/XP Document Processor, and the IBM 3892/XP Document Processor.

**3891/XP Document Processor.** A document processor in the 3890/XP Series of document processors that can read and sort documents at a rate of up to 1700 documents per minute.

**3892/XP Document Processor.** A document processor in the 3890/XP Series of document processors that can read and sort documents at a rate of up to 1000 documents per minute.

**3892/XP Power Encoder Feature.** An optional device that can be attached to the 3892/XP Document Processor to encode the MICR codeline field on a document.

**99 M-string.** See *balanced M-string*.



---

## Bibliography

The publications in this bibliography contain information related to CPCS-I.

---

### ACF/VTAM Publications

The following publications are related to the ACF/VTAM Version 3 Release 4 product:

*IBM ACF/VTAM Programming*, SC31-6436

*IBM Planning and Reference for NetView, NCP, and VTAM*, SC31-6124

*IBM VTAM Programming for LU 6.2*, SC31-6437.

---

### Document Processor Support Publications

The following publications are related to document processor support:

*IBM 3890/XP Series Document Processor General Information*, GA34-2012

*IBM 3890/XP Series Programming Guide*, GC31-2662

*IBM 3890/XP Series SPXServ Reference*, GC31-2704

*IBM 3890/XP MVS Support and 3890/XP VSE Support Program Reference*, SC31-2654

---

### High Performance Transaction System Publications (Version 1)

The following publications are related to the IBM ImagePlus High Performance Transaction System (Version 1):

*IBM ImagePlus High Performance Transaction System General Information Manual*, GC31-2706

*IBM ImagePlus High Performance Transaction System Application Library Services Programming Reference*, SC31-2794

*IBM ImagePlus High Performance Transaction System Installation Guide*, SC31-3943

---

### High Performance Transaction System Publications (Version 2)

The following publications are related to the IBM ImagePlus High Performance Transaction System (Version 2):

*IBM ImagePlus High Performance Transaction System Planning Guide*, GC31-4005

*IBM ImagePlus High Performance Transaction System Installation Guide*, GC31-4006

*IBM ImagePlus High Performance Transaction System Application Library Services Operations Guide*, SC31-4010

*IBM ImagePlus High Performance Transaction System Application Library Services Programming Guide*, SC31-4011

*IBM ImagePlus High Performance Transaction System Application Library Services Programming Reference*, SC31-4012

---

### MVS Publications (Version 5)

The following publications are related to MVS:

*IBM MVS/ESA Programming: Extended Addressability*, GC28-1468

*IBM MVS/ESA Installation Exits*, SC28-1459

*IBM MVS/ESA Initialization and Tuning Reference*, SC28-1452

*IBM MVS/ESA JCL User's Guide*, GC28-1473

*IBM MVS/ESA System Messages, Volume 1 (ABA-ASA)*, GC28-1480

*IBM MVS/ESA System Messages, Volume 2 (ASB-EWX)*, GC28-1481

*IBM MVS/ESA System Messages, Volume 3 (GDE-IEB)*, GC28-1482

*IBM MVS/ESA System Codes*, GC28-1486

*IBM MVS/DFP Version 3 Release 3: Utilities*, SC26-4559

To help you find other MVS library references for various release levels, check:

*MVS/ESA Library Guide for MVS/ESA System Product Version 4*, GC28-1601

*MVS/ESA System Product Library Guide with JES2*, GC28-1423

---

## **RACF Publications**

The following publications are related to RACF:

*IBM Resource Access Control Facility Command Language Reference*, SC28-0773

*IBM Resource Access Control Facility Security Administrator's Guide*, SC28-1340

*IBM System Programming Library: Resource Access Control Facility*, SC28-1343.

*IBM Resource Access Control Facility Master Index*, GC28-1035

---

## **CPCS Enhanced System Manager Publication**

The following publication is related to Enhanced System Manager:

*IBM Check Processing Control System: Enhanced System Manager User's Guide*, SC31-4002

---

# Index

## A

- adjustment propagation 3-8
- adjustment repository 4-3
  - item adjustment information 4-3
  - propagation status 4-3
- adjustment repository get 3-6
- Adjustments Repository Update (DKNPRUP)
  - information available from string 1-5
  - manual start 1-5
  - PRUP command 1-5
- allocating PRAD data sets 2-3
- API (application programming interface) 1-3, 3-4
- API – application programming interface 1-4
- application programming interface (API) 1-3, 1-4, 3-4
  - actions 5-6
  - alternate 5-13
  - input 5-6
  - key types 5-13
  - objects 5-6
  - prime 5-13
- application task failure 1-6
- assembling
  - DKNBLDL 2-4
  - DKNDSAT 2-4
- audit trail 3-10
- audit trail facility 4-4
- audit trail report
  - See PRAR 5-17

## C

- codes, primary return 5-11
- compiling PRAD source code 2-3
- CPCSPROC, modifying 2-4
- creating a repository of adjustments 1-4
- creation of reports 4-4
- customization 3-3

## D

- data areas A-1
- definition for propagation 1-4
- DKNAPTCB 3-4, 3-6, 3-8, 3-10
- DKNBLDL, modifications to (for PRAD) 2-3
- DKNCATCB 3-4, 3-6, 3-8, 3-10
- DKNDSAT, assembling and link-editing 2-4
- DKNPA003 3-4, 3-6, 3-8, 3-10
- DKNPA03C 3-8
- DKNPDUE1 3-4
- DKNPDUE2 3-6
- DKNPDUE4 3-10

- DKNPRAD module
  - main menu screen 4-3
  - options, start up
    - Audit Trail Generator 4-3
    - Maintenance functions 4-3
    - Manual Screen for Add and Delete adjustments 4-3
  - task initiation 4-3
- DKNPRAR module
  - creating an audit trail report 5-17
  - creation of reports 4-4
  - format of the PRAR command 4-4
  - screens 4-5
    - bank report 4-6
    - cycle report 4-7
    - generate extract file 4-8
    - manual start main menu 4-5
    - repository maintenance confirmation 4-10
    - repository maintenance only 4-9
- DKNPRUP module
  - command format 4-11
  - screens 4-12
    - manual start 4-12
- DKNPRUP task 1-4
- DKNPU01 3-4
- DKNPU02 3-6
- DKNPU03 3-8
- DKNRCVY (recovery task) 1-6

## E

- exits (PRAD) 3-4
- extended return codes, PRAD 5-11

## F

- failure of application task 1-6
- FEATOPTS, modifying 2-4
- fields, Request Parameter List 5-8
- flags, propagation 5-15
  - adjustment record 5-15
- flow diagram 1-5
- function list 1-4
- function overview 1-3

## I

- installation 2-3
- installing PRAD
  - allocate data sets 2-3
  - assembling and link-editing 2-4
  - assembling DKNBLDL 2-4
  - compiling source code 2-3

- installing PRAD (*continued*)
  - link-editing DKNBLDL 2-4
  - link-editing source code 2-3
- installing the CPCS-I libraries 2-3
- introduction to propagation of adjustments 1-3

## K

- key types
  - API 5-13
  - valid 5-14

## L

- link-editing
  - DKNBLDL 2-4
  - DKNDSAT 2-4
  - PRAD source code 2-3

## M

- maintaining a repository of adjustments 1-4
- messages B-1
- modifying
  - CPCSPROC 2-4
  - DKNBLDL 2-3
  - FEATOPTS 2-4

## O

- object/action pairs 5-8
- operations scheduling 1-4
- overview, function 1-3

## P

- parameter list (DKNPA003) 3-4
- parameter list, request
  - DKNPA001 5-7
  - DKNPA01C 5-7
- PR130 3-6
- PRAD component functions 1-4
- PRAD Exit 1 3-4
- PRAD Exit 2 3-6
- PRAD Exit 3 3-8
- PRAD Exit 4 3-10
- PRAD exits
  - PRAR audit trail (PRAD exit 4) 3-10
  - repository update exit (PRAD Exit 1) 3-4
- PRAR command, options for
  - DELETE|NODELETE 4-4
  - FILE|NOFILE 4-4
  - REPORT|NOREPORT 4-4
- primary return codes 5-11
- programming information for PRAD
  - See PRAD component functions 1-4

- propagation
  - definition 1-4
  - flags 5-15
  - status 4-3
- PRUP command, options for
  - ADD 4-11
  - DELETE 4-11

## R

- recovery from system outage 1-6
- recovery task (DKNRCVY) 1-6
- repository of adjustments 4-3
- repository update facility 4-11
- request parameter list
  - DKNPA001 5-7
  - DKNPA01C 5-7
- return codes, primary 5-11
- routines, user exit 3-4

## S

- sample user exit routines 3-4
- security with PRAD 5-16
- structure identification
  - DKNPA001 5-7
  - DKNPA002 5-8
  - DKNPA003 3-4, 5-8
- system considerations 1-3
- system outage, recovery 1-6

## T

- task control block (TCB) 3-4
- TCB (task control block) 3-4

## U

- user exits 1-4
  - adjustment propagation 3-8
  - adjustment repository get 3-6
  - adjustment repository update 3-4
  - audit trail 3-10

## V

- valid key types 5-14



---

# Communicating Your Comments to IBM

Check Processing Control System  
International MVS/ESA  
Propagation of Adjustments  
Guide  
Release 1  
Publication No. SC31-3994-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments electronically, use this network ID:  
Internet: [www.ibm.com/Products/CPCS](http://www.ibm.com/Products/CPCS)

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.



---

# Readers' Comments — We'd Like to Hear from You

Check Processing Control System  
International MVS/ESA  
Propagation of Adjustments  
Guide  
Release 1  
Publication No. SC31-3994-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

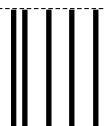
\_\_\_\_\_  
Phone No.



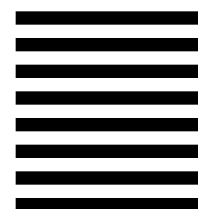
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Payment Solutions  
Department 58G, MG96/204  
8501 IBM Drive  
Charlotte NC 28262-8563



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5799-FKT



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC31-3994-00

